Genesys Voice Platform 8.0

User's Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit 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

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

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 Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support 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 7 Licensing Guide***.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 80gvp_us_06-2008_v8.0.001.00

Table of Contents

List of Procedures

Preface

Welcome to the *Genesys Voice Platform 8.0 User's Guide*. This document provides detailed information about the configuring, provisioning, and monitoring of Genesys Voice Platform (GVP) and its components.

This document is valid only for the 8.0 release.

Note: For releases of this document created for other releases of this product, please 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 provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information: It contains the following sections:

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 8.0 is fully integrated with the Genesys Management Framework. GVP uses the 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, but the initial release of GVP 8.0 is available for single-tenant configurations only.

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.

Chapter Summaries

In addition to this preface, this Deployment Guide contains the following chapters and appendixes:

- Part 1: "Overview" on page 19
  - Chapter 1, "Introduction," on page 21, provides a high-level description of the GVP architecture and the Genesys Administrator.
  - Chapter 2, "How GVP Works," on page 25, describes in detail how the GVP components perform their functions.
- Part 2: "Provisioning GVP" on page 75
  - Chapter 3, "Configuration and Provisioning Overview," on page 77, provides a general description of the steps to configure GVP components, as well as an overall summary of the tasks to configure GVP.
  - Chapter 4, "Configuring Common Features," on page 85, describes how to configure features that are configured across components.
  - Chapter 5, "Configuring the Resource Manager," on page 123, describes how to configure the Resource Manager, including configuring logical resource groups and High Availability.
  - Chapter 6, "Provisioning IVR Profiles," on page 139, describes how to configure the IVR Profiles and provision the VoiceXML and CCXML applications for the deployment.
  - Chapter 7, "Configuring the Media Control Platform," on page 157, describes how to configure the Media Control Platform, including enabling Automatic Speech Recognition (ASR) and Text-to-Speech (TTS).
  - Chapter 8, "Configuring the Call Control Platform," on page 183, describes how to configure the Call Control Platform and provision the device profiles.
  - Chapter 9, "Configuring the Fetching Module and Squid Proxy," on page 197, describes how to configure the Fetching Module and Squid Caching Proxy.

  - Chapter 10, “Configuring the Reporting Server,” on page 203, describes how to configure the Reporting Server for important reporting behavior, such as granularity of reports and data retention policies.
- Part 3: “Monitoring GVP” on page 217
  - Chapter 11, “Reporting Overview,” on page 219, describes how to run reports and filter report results.
  - Chapter 12, “Real-Time Reports,” on page 227, describes the content of the real-time reports that are available in GVP.
  - Chapter 13, “Historical Reports,” on page 233, describes the content of the historical reports that are available in GVP.
  - Chapter 14, “Voice Application Reports,” on page 243, describes the Voice Application Reporter (VAR) reports that are available in GVP.
- Part 4: “Appendixes” on page 253
  - Appendix A, “Module and Specifier IDs,” on page 255, provides module, specifier, and metrics ID information that is required for configuring certain reporting and logging settings.
  - Appendix B, “SIP Response Codes,” on page 279 describes the SIP response codes that are generated by GVP components.
  - Appendix C, “Media Control Platform Reference Information,” on page 289 provides detailed reference information that relates to Media Control Platform functioning.
  - Appendix D, “Default Device Profiles,” on page 304 lists the properties of the device profiles that are provisioned by default.
  - Appendix E, “Specifications and Standards,” on page 307 lists the W3C Voice Browser Working Group and Internet Engineering Task Force (IETF) standards that GVP supports.
  - Appendix F, “Caching Reference Information,” on page 315 describes the caching algorithms that the Fetching Module and Squid caching proxy use, as well as the entries in the Squid access logs.

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:

```
72fr_ref_09-2005_v7.2.000.00
```

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents' titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys 7 Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, $x + 1 = 7$ where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for 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; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.
- The following table presents the complete set of error messages T-Server® distributes 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.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. 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.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Use of Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Note: In some cases (for example, VoiceXML and CCXML tags), angle brackets are required syntax elements and do not indicate placeholders. Notes in the text indicate where this is the case.

Related Resources

Consult the following additional resources as necessary:

- *Genesys Voice Platform 8.0 Deployment Guide,* which provides information to install and configure GVP.
- *Genesys Voice Platform 8.0 VoiceXML 2.1 Help,* which provides information about developing VoiceXML applications. It presents VoiceXML concepts and provides examples that focus on the GVP implementation of VoiceXML.
- *Genesys Voice Platform 8.0 CCXML Reference Manual,* which provides information about developing CCXML applications for GVP.

- *Genesys Voice Platform 8.0 Troubleshooting Guide,* which provides information about SNMP MIBs and traps for GVP, as well as troubleshooting methodology.
- *Genesys Voice Platform 8.0 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 8.0 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.
- *Voice Platform Solution 8.0 Integration Guide,* which provides information about integrating GVP 8.0 and SIP Server 7.6.
- *Composer Voice 8.0 Deployment Guide,* which provides installation and configuration instructions for Composer Voice.
- *Composer Voice 8.0 Help,* which provides online information about using Composer Voice, a GUI for the development of applications based on VoiceXML and CCXML.
- *W3C Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation 19 June 2007,* which is the W3C VoiceXML specification that GVP 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.
- *Framework 8.0 Deployment Guide,* which provides information to configure, install, start, and stop Framework components.
- *Framework 8.0 Configuration Options Reference Manual,* which provides descriptions of configuration options for Framework components.
- *Framework 8.0 Genesys Administrator Help,* which provides instructions for configuring and provisioning contact center objects using Genesys Administrator.
- *Framework 7.6 SIP Server Deployment Guide,* which provides information to configure and install SIP Server.
- *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.

- 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 Technical Support website in the following documents:

- *Genesys Supported Operating Systems and Databases*
- *Genesys Supported Media Interfaces*

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.

Making Comments on This Document

If you especially like or dislike anything about this document, please 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 information in this document only and to the way in which the information is presented. Speak to 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.

Part

1 Overview

This part of the manual provides general information about Genesys Voice Platform (GVP) and the Genesys Administrator.

This information appears in the following chapters:

Chapter

1 Introduction

Genesys Voice Platform (GVP) is a software suite that integrates a combination of 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 the Genesys Administrator, the GUI to configure and manage GVP.

This chapter contains the following sections:

About GVP

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

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. The Resource Manager is responsible for:
  - Allocating and monitoring resources.
  - Managing sessions.
  - Selecting services.
  - Enforcing policies.

The Resource Manager also functions as a SIP Registrar, and monitors the health of GVP resources in the deployment.

For information about how the Resource Manager performs its major functions, see "How the Resource Manager Works" on page 27.

- **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. Functions include:
  - Initiating, answering, transferring, and disconnecting calls.
  - Playing audio and Text-to-Speech (TTS) prompts.
  - Handling Automatic Speech Recognition (ASR) and DTMF inputs.
  - Providing conference services.

  For information about how the Media Control Platform performs its functions, see "How the Media Control Platform Works" on page 37.
- **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. Functions include:
  - Accepting, rejecting, and redirecting calls, including handling call setup information to enable intelligent routing and selective answering. Call-handling capabilities include supervised transfer, whispering, and call hold.
  - Creating outbound calls through third-party gateways.
  - Using Media Control Platform services to initiate VoiceXML dialogs, start conferences, and perform implicit transcoding.
  - Providing multi-party conference support with moderator and floor control capabilities.
  - Providing 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 how the Call Control Platform performs its functions, see "How the Call Control Platform Works" on page 50.

  For information about creating CCXML applications to utilize Call Control Platform capabilities, see the *Genesys Voice Platform 8.0 CCXML Reference Manual.*
- **Fetching Module, with Third Party Squid**—Fetches content (such as audio files, VoiceXML pages, or CCXML pages) for the Media Control Platform and Call Control Platform sessions. Each Media Control Platform and Call Control Platform component in the deployment requires a Fetching Module. Third Party Squid is a caching proxy that the Fetching Module uses to improve response times by caching and reusing frequently requested resources.

For information about how the Fetching Module and Squid perform their functions, see "How the Fetching Module Works" on page 55.

- **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 about how the Reporting Server performs its functions, see "Logging and Reporting" on page 60.

For more general information about the GVP components, see the chapter about GVP architecture in the *Genesys Voice Platform 8.0 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 to control the interaction between the external user and the GVP software.

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

GVP MIBs

The VP 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 "SNMP Monitoring" on page 70. For detailed information about the MIBs, see the *Genesys Voice Platform 8.0 Troubleshooting Guide.*

Genesys Administrator

The Genesys Administrator is a GUI that provides a web-based interface to the Genesys Configuration and Management Layers.

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

Figure 1 shows a typical Genesys Administrator page.

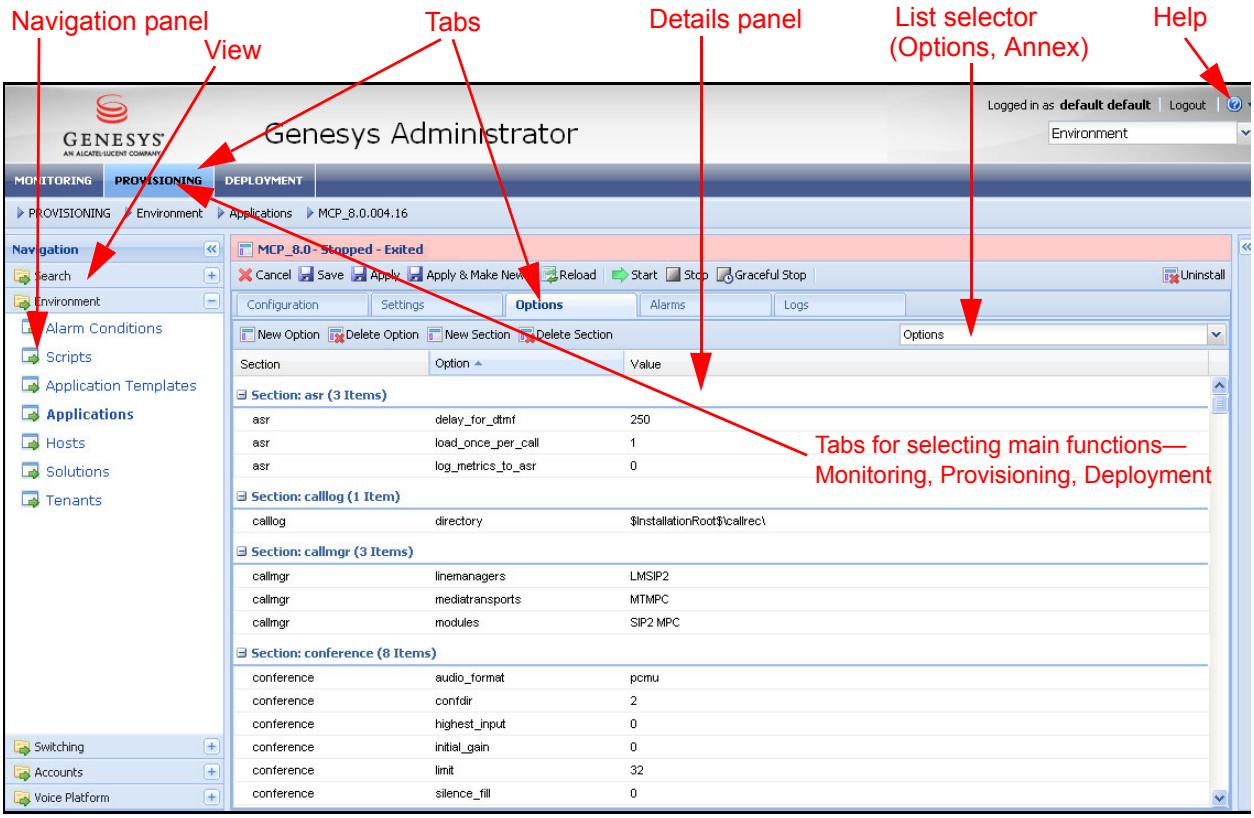


Figure 1: Genesys Administrator

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

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

More Information

- For information about installing the Genesys Administrator, see the *Framework 8.0 Deployment Guide.*
- For general information about using the Genesys Administrator, see the online *Framework 8.0 Genesys Administrator Help.*
- For information about using the Genesys Administrator to configure and provision GVP `Application` objects and IVR Profiles, see Viewing or modifying GVP configuration parameters, page 78 and Configuring logical resource groups, page 127.
- For information about using the Genesys Administrator to monitor GVP and view reports, see Part 3 of this manual, starting on page 217.

Chapter

2 How GVP Works

This chapter describes how the Genesys Voice Platform (GVP) components operate in a GVP deployment. It also provides general information about the identifiers and Session Initiation Protocol (SIP) messages that are used in a GVP deployment.

This chapter contains the following sections:

Together with SIP Server 7.6 and Management Framework 8.0, GVP 8.0 constitutes the Voice Platform Solution (VPS) 8.0, which integrates voice self-service, agent-assisted service, and application management functions into a single, IP-based contact center solution.

For more information about how GVP functions within a VPS deployment, see the *Voice Platform Solution 8.0 Integration Guide.*

Sample Call Flow

GVP can be deployed and provisioned in a number of ways, to provide a range of services for inbound calls, outbound calls, transfers, and conferences.

Figure 2 illustrates how GVP handles a typical inbound call.

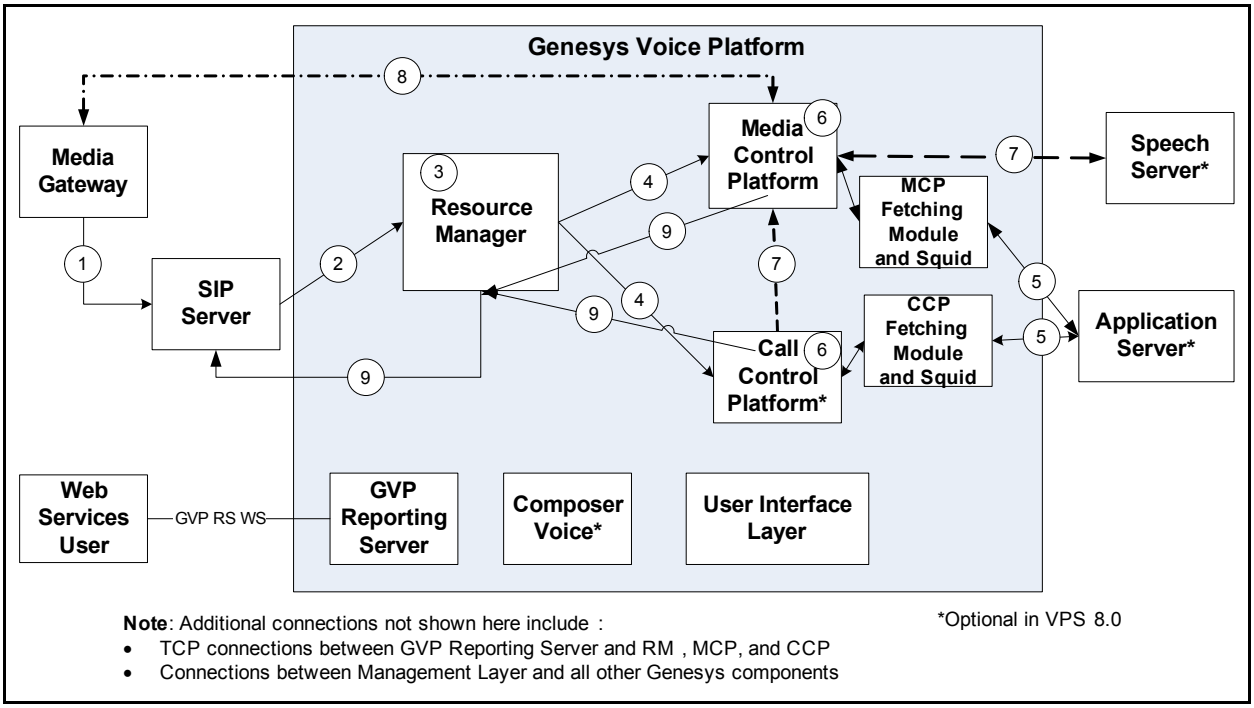


Figure 2: Typical Inbound Call Flow

1. A call comes in to the SIP Server from an external source through a third-party media gateway.
2. The SIP Server passes the call to the GVP Resource Manager (SIP `INVITE`).
3. The Resource Manager determines what to do with the call. If the Resource Manager accepts the call, it matches the call to an IVR Profile, and selects a resource.

   For more information about how the Resource Manager selects services and resources and enforces policies, see "How the Resource Manager Works" on page 27.
4. The Resource Manager sends the call to a Media Control Platform or Call Control Platform resource (SIP `INVITE`). When it forwards requests to resources, the Resource Manager inserts additional SIP headers or parameters, as required by the service prerequisites, service parameters, and policies that have been configured for the IVR Profile. For more information, see "Service Request Modification" on page 32.
5. The Fetching Module for that Media Control Platform or Call Control Platform resource fetches the required VoiceXML or CCXML page from the application server (file, HTTP, or HTTPS request).
6. The Next Generation Interpreter (NGI) on the Media Control Platform or CCXML Interpreter (CCXMLI) on the Call Control Platform interprets the page and runs the application (VoiceXML or CCXML).

7. Depending on the application, the Media Control Platform or Call Control Platform will request and use additional services:
   - For Automatic Speech Recognition (ASR) or Text-to-Speech (TTS) services, the Media Control Platform communicates with the third-party speech application server using Media Resource Control Protocol (MRCPv1 or MRCPv2).
   - If the Call Control Platform requires conferencing or audio play/record services, it obtains them from a Media Control Platform resource.

   The Media Control Platform or Call Control Platform send all requests for services from other GVP components through the Resource Manager (SIP or NETANN).
8. The Real-time Transport Protocol (RTP) media path is established between the Media Control Platform and the SIP end-user, in this example the originating caller through the media gateway.
9. The Resource Manager ends the call when one of the parties (the SIP end-user, the Media Control Platform, or the Call Control Platform) disconnects or when the call is transferred out of GVP (SIP `BYE` or `REFER`).

How the Resource Manager Works

This section describes in more detail what the Resource Manager does with service requests, such as the call described in "Sample Call Flow" on page 25.

The Resource Manager performs the following functions:

- Session Management
- Service Selection (see page 29)
- Policy Enforcement (see page 32)
- Service Request Modification (see page 32)
- Resource Management (see page 32)
- High Availability and Scalability (see page 37)

For information about how the Resource Manager reports it activities to the Reporting Server, see "Logging and Reporting" on page 60.

Session Management

A *session* is a set of related services that are used to deliver an end-user experience.

There is a global logical session that encompasses the Resource Manager interactions with SIP Server. The global session is managed by SIP Server. Within the global session, the Resource Manager manages logical call sessions for specific GVP services, and individual call legs within a call session.

The Resource Manager manages GVP call sessions as follows:

1. The Resource Manager creates a new call session when it receives a new SIP `INVITE` request for a GVP service.
2. The Resource Manager generates a GVP Session ID, and inserts this information in the `X-Genesys-GVP-Session-ID` SIP extension header. For more information about session IDs, see "Session Identifiers" on page 72.
3. The Resource Manager maps the session to an IVR Profile (a voice or call control application) and identifies the type of service for each component session (call leg). For more information, see "Service Selection" on page 29.
4. The Resource Manager adds the following parameters to the `X-Genesys-GVP-Session-ID` header:
   - `gvp.rm.datanode`—To identify itself as the Resource Manager for the session. In GVP 8.0, the value of this parameter is not configurable.
   - `gvp.rm.tenant-id`—To identify the voice or call control application under which the session executes. The value of this parameter is the name of the IVR Profile that was assigned when the profile was configured. For more information about IVR Profile IDs, see "Application Identifiers" on page 72.
5. The Resource Manager inserts the `X-Genesys-GVP-Session-ID` header when it forwards new SIP `INVITE` requests. The Resource Manager also adds the `X-Genesys-GVP-Session-ID` header when it forwards responses, if the header does not already exist in the response.
6. The Resource Manager inserts the `X-Genesys-RM-Application-dbid` header when it forwards the first SIP `INVITE` request to a GVP resource, to identify the IVR Profile under which the session executes. The value of this parameter is the Database Identifier assigned by Configuration Server (DBID) for the `IVR Profile` object.

   The GVP components need this information to log the IVR Profile DBID in the call detail records (CDRs) that they send to the Reporting Server.
7. When the Resource Manager receives a SIP `INVITE` request for a new call leg within an existing session (as identified by the `X-Genesys-GVP-Session-ID` header), the Resource Manager consults the policies of the IVR Profile and Tenant related to the existing session, to determine if the call leg can be created. For more information, see "Policy Enforcement" on page 32.

   If the Resource Manager cannot identify an existing session from the `X-Genesys-GVP-Session-ID` header (for example, because the session has timed out), the Resource Manager accepts and processes the incoming request without checking policies.
8. The Resource Manager maintains the session in accordance with configurable session inactivity and session expiry timers.

   The Resource Manager associates a session inactivity timer with each call leg, and monitors SIP traffic for the session to determine when a SIP

session is stale. If the Resource Manager receives no SIP messages for the call leg within the inactivity interval, the Resource Manager internally cleans up the call leg data (application, tenant, and resource usage) as if a `BYE` was received.

You can set session inactivity timers for each IVR Profile, for each tenant, for each resource, and for the Resource Manager. For more information about the session timer configuration parameters, see "Configuring Session Timers and Timeouts" on page 117.

Session-Expires Header

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 timer, except under the following conditions:

- If the incoming request contains a `Session-Expires` header with a value greater than the configured value of the applicable session timer, and if the `Min-SE` header is also present, the Resource Manager reduces the value of the `Session-Expires` header to the greater of the `Min-SE` and configured session timer values.
- If the incoming request contains a `Session-Expires` header with a value greater than the configured value of the applicable session timer, but the `Min-SE` header is not present, the Resource Manager reduces the value of the `Session-Expires` header to the configured session timer value.
- If the incoming request contains a `Session-Expires` header with a value less than the minimum session expiry value configured for the Resource Manager (see `proxy. sip.min_se` on page 119), the Resource Manager rejects the request.
- If the incoming request contains a `Session-Expires` header with a valid value less than the configured value of the applicable session timer, the Resource Manager uses the value of the `Session-Expires` header.

The Resource Manager restarts the session inactivity timer each time it receives a SIP request or `200 OK` response.

Service Selection

When the Resource Manager receives a request for a new SIP session, it maps the call to an IVR Profile, and then selects a service for the request. If the SIP request arrives in the context of an existing Resource Manager session for which a VoiceXML or CCXML application is already executing, the Resource Manager does not perform another mapping.

Mapping the Call to an IVR Profile

The Resource Manager maps the SIP request to an IVR Profile as follows:

1. If the SIP `Request-URI` includes a `gvp-tenant-id` parameter, the Resource Manager looks for an IVR Profile with a name that matches the value of the `gvp-tenant-id` parameter.
   - If the Resource Manager finds an IVR Profile that matches, it routes the SIP session to that application. When it does so, it removes the SIP `Request-URI` parameter from the outgoing request.
   - If the Resource Manager does not find an IVR Profile that matches, it executes the default VoiceXML or CCXML application that has been configured for the `Environment` tenant (see default-application, in the `gvp.general` section, on page 154).
2. If the SIP `Request-URI` does not include a `gvp-tenant-id` parameter, but the VPS has been configured so that SIP Server provides DNIS information in the SIP header, the Resource Manager uses the DNIS that it extracts from the SIP message to map to an IVR Profile from a preconfigured DNIS resource list. When the Resource Manager routes the call to the application, it attaches the `trunkport` parameter to the SIP `Request-URI`, with the DNIS as the value.

   For information about configuring the mapping between DNIS ranges and IVR Profiles for the `Environment` tenant, see "Mapping IVR Profiles to Dialed Numbers" on page 151.

   A Resource Manager configuration option (`sip-header-for-dnis`, in the `rm` section) enables you to specify the header in which the Resource Manager will look for the DNIS. Ensure that the value you specify is consistent with the headers that you expect the Media Gateway to use. Valid values are:
   - The user part of the SIP `Request-URI`.
   - The user part of the URI in the `To` header.
   - The user part of the URI in the `History-Info` header, with index = `1`.

   For more information, see the description of the `rm.sip-header-for-dnis` configuration option on page 126.

   Note: GVP 8.0 supports numeric DNIS only. All other characters (for example, `*`, `#`, `a`, `b`) are stripped from the incoming request.

3. If the Resource Manager cannot map the SIP request to an IVR Profile, the Resource Manager executes the new SIP session in the context of the default VoiceXML or CCXML application that has been configured for the `Environment` tenant (see default-application, in the `gvp.general` section, on page 154).
4. If the Resource Manager cannot map the SIP request to an IVR Profile and no default application has been configured for the `Environment` tenant, the incoming SIP request fails with a `404 Not Found` response.

Selecting the Service

After the IVR Profile for the Resource Manager session has been determined, the Resource Manager identifies the required service and the service prerequisites for each call leg.

The Resource Manager performs service selection as follows:

1. If the user part of the SIP `Request-URI` includes parameters that specify the required service, the Resource Manager handles the SIP request as a request for the specified service. The Resource Manager appends service parameters to the `Request-URI` if they are not already included.

   Table 1 describes the parameters that the Resource Manager looks for in SIP messages, to specify the required service.

Table 1: Service Specified in SIP Request

SIP Request-URI	Service	Comment
User part is `dialog`, and contains a parameter with name `voicexml`	voicexml	Service prerequisites included.
User part starts with `dialog.vxml`	voicexml	Service prerequisites included.
User part is `ccxml`, and contains a parameter with name `ccxml`	ccxml	Service prerequisites included.
User part starts with `conf=`	conference	Service prerequisites included. The remainder of the user part is the Conference ID for this request.
Contains a parameter `user=phone`	gateway	Request originates from a resource that supports `voicexml` or `ccxml` service.
User part is composed of characters `0`–`9` and `-` (dash)	gateway	Request originates from a resource that supports `voicexml` or `ccxml` service.

2. If the SIP `Request-URI` does not include parameters to identify the required service, and the request originates from a resource that supports `voicexml` or `ccxml` service, the Resource Manager handles the incoming SIP request as a request for `external-sip` service.
3. If the SIP `Request-URI` does not include parameters to identify the required service, and the request does not originate from a resource that supports `voicexml` or `ccxml` service, the Resource Manager uses the service type and

service prerequisites information that has been configured for the IVR Profile (see the configuration options described in "gvp.general Section" on page 141 and "gvp.service-prerequisite Section" on page 150).

4. If the Resource Manager cannot map the request to a service in accordance with the above rules, the Resource Manager rejects the request with a `404 Not Found` SIP response.

Policy Enforcement

The Resource Manager tracks sessions and IVR Profile and service usage to enforce policies that are imposed on a per-application basis. The Resource Manager also consults dialing and service allowability rules to enforce policies that are imposed on a per-application and per-tenant basis.

For the IVR Profile and Tenant configuration parameters that specify the policies you can configure for GVP, see "gvp.policy Section" on page 143 and "gvp.policy.dialing-rules" on page 148. Configuration options enable you to customize the SIP responses that are sent when the Resource Manager rejects a call because of policy criteria, as well as whether certain policy violations will trigger an alarm.

Service Request Modification

Before the Resource Manager forwards the request to a resource that can handle the service, it adds, deletes, or otherwise modifies SIP parameters to capture user-defined data and to translate policy and other configuration information into SIP parameters, which can be extracted by the VoiceXML or CCXML applications. You can configure GVP so that different service parameters are used for each service of each application.

For details about the service parameters and service prerequisites that you can configure for IVR Profiles and the `Environment` tenant, as well as the SIP parameters in which the Resource Manager captures this information, see "gvp.service-parameters Section" on page 149 and "gvp.service-prerequisite Section" on page 150.

External SIP Service

External SIP service, which is one of the service types that GVP supports, enables the Resource Manager to route an outbound call through an external, RFC3261-compliant SIP proxy. When the Resource Manager forwards a request for external SIP service, it adds a `Route` header to the SIP message with a `Route URI` that includes the `lr` parameter.

Resource Management

All requests for GVP services go through the Resource Manager, which identifies a SIP resource capable of serving the request and forwards the request to it. The Resource Manager monitors GVP resources to maintain up-to-date status of the resources used in the GVP deployment. The Resource

Manager manages GVP resources to provide load-balancing and, if applicable, High Availability for each resource type.

The following subsections provide more details about how the Resource Manager performs its resource management functions.

Logical Resource Groups

The Resource Manager receives resource information from Genesys Management Framework. Logical resource objects in Management Framework represent groupings of resources that share common properties, such as service type (for example, `voicexml`), capabilities (for example, support for a specific VoiceXML grammar), and method of load-balancing (for example, round robin).

Resources are grouped by the following service types:

- `voicexml`
- `ccxml`
- `conference`
- `gateway`
- `external`

For detailed information about all the resource group properties, see Table 20 on page 131.

Management Framework gives the Resource Manager a list of logical resource objects and a list of physical resources. Each physical resource belongs to a logical resource.

Monitoring Status

The Resource Manager acts as a SIP Registrar (RFC3261) for resources about which it receives information from Management Framework. The Resource Manager maintains information on the registration and usage status of each resource. If the logical resource group has been configured for monitoring (see monitor-method on page 132), the Resource Manager also monitors resource health. Health monitoring performed by the Resource Manager is separate from Simple Network Management Protocol (SNMP) management (see "SNMP Monitoring" on page 70).

Configuration options enable you to control the monitoring behavior of the Resource Manager (see the `registrar` and `monitor` configuration sections for the Resource Manager `Application` object).

Selecting a Resource

Service Capabilities

After the Resource Manager has mapped a new SIP request to a service, it allocates the request to a logical resource group that can provide the service,

with the specific capabilities that are required by the VoiceXML or CCXML application.

The Resource Manager uses two sources to identify what the capability requirements are:

- The IVR Profile service policies (see "gvp.policy Section" on page 143).
- Information parsed from SIP `Request-URI` parameters that have the prefix `gvp.rm.resource-req`. The Resource Manager does not forward these `Request-URI` parameters.

Load-Balancing

After the Resource Manager has selected a logical resource group for the service request, it allocates the request to a physical resource. Except for `conference` services (see "Selecting a Resource for Conference Services" on page 34), the Resource Manager selects the physical resource based on the load-balancing scheme for the group. Load-balancing options are:

- Round robin—From a circular list, the Resource Manager chooses the next resource whose usage has not exceeded configured limits.
- Least used—The Resource Manager chooses the resource with the lowest usage and whose usage has not exceeded configured limits. *Usage* is calculated as specified by the `port-usage-type` parameter (see port-usage-type on page 133).

Notes: The Resource Manager load-balances within a logical resource group. It does not load-balance between resource groups.

The MRCP Client on the Media Control Platform, which provides Speech Resource Management (SRM), load-balances the selection of the third-party speech engines, on a round-robin basis.

Load-Balancing for Gateway Service

For `gateway` service, the Resource Manager selects a resource based on a configurable policy option that lets you specify whether the call must be routed to the gateway resource already associated with the session, or whether the usual load-balancing scheme will be used (see the IVR Profile `gvp.policy.use-same-gateway` configuration option, on page 148).

No Resource Selected

If the Resource Manager is not able to select a resource to meet the request, the Resource Manager responds to the SIP request with a configurable error message (see "Customizing SIP Responses" on page 116 and Table 64 on page 279).

Selecting a Resource for Conference Services

Conference services have the special requirement that the Resource Manager must route requests for the same conference ID to the same conference resource, even if the requests come from different Resource Manager sessions.

1. If the SIP `Request-URI` includes `confmaxsize` and `confreserve` parameters, and the specified `confmaxsize` is less than the `confreserve`, the Resource Manager rejects the conference request.

2. For the first request that the Resource Manager receives for a conference (in other words, the Resource Manager is not already handling requests with the requested conference ID), it identifies eligible conference resources by matching `confmaxsize` and `confreserve` requirements specified in SIP `Request-URI` parameters with conference maximums that have been configured for the IVR Profile and the resource group, in combination with its knowledge of the current status and usage of `conference` resources. For more information about the IVR Profile and resource group parameters that are considered, see "Enabling Conference Services" on page 102.
3. The Resource Manager selects a resource for the conference by load-balancing across the eligible resources, in accordance with the load-balancing scheme for the logical group (see "Load-Balancing" on page 34).

   The Resource Manager adds the `confmaxsize` and `confreserve` parameters to the outgoing `Request-URI` when it forwards the request.
4. When the conference session is successfully established, the Resource Manager increments the current usage of the resource by the expected size of the conference (as specified in the `confreserve` SIP `Request-URI` parameter—default value is `1` if the parameter was not defined).

   As new call legs join or leave the conference, the Resource Manager keeps track of the current conference size.

   Note: Once the conference session is established, the maximum number of participants in the conference is the smallest of the conference size maximums (`confmaxsize` parameters) specified in the SIP request, the IVR Profile, and the resource group.

   As a result, the Resource Manager may internally modify the `confmaxsize` parameter in the outgoing SIP `Request-URI`, and this might cause the Resource Manager to reject the conference request if the `confmaxsize` parameter in the outgoing SIP `Request-URI` becomes smaller than the `confreserve` parameter (see Step 1 on page 34)
5. When the Resource Manager receives subsequent requests with the same conference ID, the Resource Manager forwards the request to the same `conference` resource, provided the maximum conference size is not exceeded.

   If conference size maximums have not been defined in the SIP request, IVR Profile, or resource group, the Resource Manager forwards the request to the conference resource, and leaves it to the conference resource to reject the request if necessary.

If adding the new call leg means that the maximum size of the conference or the usage limit configured for the resource would be exceeded, the Resource Manager rejects the request.

6. If the Resource Manager is not able to select a resource to meet the request, the Resource Manager responds to the SIP request with a configurable error message (see "Customizing SIP Responses" on page 116 and Table 64 on page 279).

Failed Requests

The behavior of the Resource Manager in response to failed requests depends on the type of service and the failure response code received by the Resource Manager.

- For `voicexml`, `ccxml`, and `conference` service requests for which it receives a `4xx` or `5xx` response code, the Resource Manager tries to select another resource, in accordance with the load-balancing scheme for the group, until all resources in the group have been tried.
- For a `gateway` service request for which it receives a `4xx` or `5xx` response code or for which the request times out, the Resource Manager forwards the failure response to the User Agent Client (UAC).
- For any `INVITE` requests to create a new SIP dialog for which it receives a `6xx` response, the Resource Manager immediately forwards the response to the UAC, without trying to select another resource.
- If the Resource Manager has received no successful `2xx` responses and has received at least one final response from one of the resources, the Resource Manager chooses one of the received responses to forward to the UAC, in the following order of selection:

Response Received	Response Returned to the UAC
`6xx`	`6xx`
`401, 407, 415, 420, 484`	`401`
Any other `4xx` response	The first `4xx` response received
Any `5xx` other than `503`	The first `5xx` response received
`500` (Server Internal Error)	`500` (Server Internal Error)

- If the Resource Manager has sent at least one request to a resource and has not received any final responses from any resource, the Resource Manager sends a `408` (Request Timeout) response to the UAC.

For more information about SIP response codes that are generated by GVP components, see "SIP Response Codes" on page 279.

High Availability and Scalability

GVP 8.0 supports GVP High Availability (HA) in two senses:

- Resource components—Because resources are provisioned in logical resource groups, the Resource Manager provides HA for GVP components in the normal way that it manages, monitors, and load-balances the resource groups. As long as more than one instance of a particular GVP `Application` has been provisioned in a resource group, the service that the GVP process provides will still be available to other VPS components if one of the provisioned instances is not available.

Resource Manager HA

- Resource Manager—Windows Network Load Balancing (NLB) provides HA for the Resource Manager itself. You can configure two Resource Managers to run as a warm active-standby pair with a common virtual IP.

  Incoming IP traffic is load-balanced across NLB clusters. A Cluster Manager on each Resource Manager host monitors the Network Interface Cards (NICs) in the cluster, to determine when network errors occur. If any of the monitored NICs encounter an error, the Cluster Manager considers the network down. The load-balancing of incoming IP traffic is adjusted accordingly.

  For information about configuring the Resource Manager for HA, see "Enabling High Availability" on page 134.

Scalability

NLB clustering also provides scalability, because adding Resource Manager hosts to a cluster increases the management capabilities and computing power of the Resource Manager function in the GVP deployment.

Note: GVP 8.0 does not support more than two Resource Managers in a cluster.

How the Media Control Platform Works

This section provides information about the following topics, to explain how the Media Control Platform performs its role in a GVP deployment:

About the Media Control Platform

The Media Control Platform is composed of:

- A core executable that consists of a number of application modules, such as the Call Manager API (CMAPI) and the SIP Line Manager. (For a list of the Media Control Platform application modules, see "Media Control Platform" on page 255.)
- The Media Server, which is a group of libraries (`libCMMPC.dll`, `libDSP*.dll`, and third-party transcoder Dynamic Link Libraries [DLLs]) running in-process in the Media Control Platform, for media processing and Real Time Transport Protocol (RTP) streaming.
- The Next-Generation VoiceXML Interpreter (NGI), which is a DLL (`libAPPVXML3.dll`) running in-process in the Media Control Platform.
- The MRCP Client, which is a group of libraries (`libmrcpclient.dll`, `libmrcpv1client.dll`, `libmrcpv2client.dll`) running in-process in the Media Control Platform, to handle MRCPv1 or MRCPv2 communication with ASR and TTS speech engines.

The Fetching Module, which is a separate GVP component, co-resides on the Media Control Platform host. The NGI and the Fetching Module share memory.

For more information about the Media Control Platform architecture and interfaces, see the chapter about GVP architecture in the *Genesys Voice Platform 8.0 Deployment Guide*.

Operational Overview

The Media Control Platform receives requests for call and media services from the Resource Manager in the form of SIP `INVITE` messages. The platform can conference, transfer, or redirect calls using other kinds of SIP messages (see "Transfers" on page 45). The platform can also initiate outbound calls by sending SIP `INVITE` requests through the Resource Manager or directly to the destination.

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

Incoming Calls

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

1. The Media Control Platform, acting as a SIP User Agent Server (UAS), receives a SIP `INVITE` from the Resource Manager. Because the Resource Manager has modified the SIP request to insert service-prerequisites for the IVR Profile, the SIP `Request-URI` includes a `voicexml` parameter that specifies the URL of the initial page of the required VoiceXML application.

   Alternatively, the Media Control Platform may be configured to accept calls in which the originator specifies the initial VoiceXML URL in the `Request-URI` of the SIP `INVITE` (see `sip.vxmlinvite` on page 176). In these cases, provided the syntax and format of the `Request-URI` is correct, the normal Resource Manager method of mapping calls to IVR Profiles will be bypassed.

   - The Media Control Platform recognizes the following `Request-URI` formats:
     - `sip:dialog.vxml.<URL>@host.com`, where the URL portion must be properly encoded (draft-rosenberg-sip-vxml format)
     - `sip:<user>@host.com;voicexml=<URL>` (NETANN dialog service format)
     - `sip:conf=<ID>@host.com` (NETANN format, for calls to join a specified conference without going through a VoiceXML application)
   - The Media Control Platform supports the following service parameters in the `Request-URI`:
     - `voicexml`—Value must conform to the URI syntax defined in RFC 3986.
     - `maxage`—Value must be all digits.
     - `maxstale`—Value must be all digits.
     - `method`—Value must be either `get` or `post`.
     - `postbody`
     - `timeout`
     - `gvp.alternatevoicexml`—Specifies an alternative VoiceXML page if the NGI fails to fetch the primary page.
     - `gvp.config.<parameter name>`—Sets the values of certain platform configuration options for the duration of the media session. This mechanism enables an IVR Profile to override certain Media Control Platform configuration parameters, for the session that is being executed in the context of that VoiceXML application. For the configuration parameters whose values can be set dynamically, see "Dynamic Media Control Platform Parameters" on page 294.

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

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

  For more information about these policy parameters, see Table 23 on page 143.
- If media services are required, the Resource Manager includes the SIP User Agent (UA) Session Description Protocol (SDP) offer in the SIP `INVITE`.

  For more information about how the Media Control Platform negotiates media services, see Step 6 on page 41.

2. For valid `INVITE` requests, the platform immediately responds to the Resource Manager with a `100 TRYING` message. In addition, configurable options enable you to specify whether the platform will also send intermediate provisional responses while the call is being set up.

   Provisional responses can include custom SIP headers, which must have the prefix, *X-*.

   Note: The Media Control Platform does not support sending early media after a provisional response has been sent.

   For responses that the Media Control Platform sends if an error occurs during call setup, see "SIP Response Codes" on page 279.
3. The Media Control Platform passes all the generic SIP `Request-URI` parameters to the NGI.
4. The NGI sends an HTTP/HTTPS or file retrieval request to the Fetching Module to fetch the initial page. The request includes the `timeout`, `maxage`, and `maxstale` values, if present, to determine if the fetch can be satisfied from the cache store.

   For more information about how caching is used to improve Media Control Platform performance, see "Caching" on page 56.
5. The NGI compiles and interprets the initial page, and all subsequent pages, for the Media Control Platform to run the application. The VoiceXML application is ready to proceed when the VoiceXML document is fetched, parsed, and compiled.

   The NGI supports the following encodings for VoiceXML pages and external ECMAScripts:
   - UTF-8

- UTF-16
- ISO-8859-x
- Far-East encoding for Japanese, Chinese and Korean

The NGI retrieves the encoding information for a document from the `encoding` attribute of the XML header or the `charset` attribute of the `<script>` tag.

6. At the same time that it passes SIP `INVITE` information to the NGI (Step 3), the Media Control Platform passes the SDP to the Media Server, to negotiate media capabilities.

   For information about the capability negotiation and the codecs that the Media Control Platform supports, see "Codec Negotiation" on page 43. For information about the file formats that are supported for audio and video play and record for various codecs, see "Audio and Video File Formats" on page 289.

7. The Media Control Platform sends a `200 OK` response to the initial `INVITE` request when the VoiceXML application is ready to execute. The response includes the Media Server SDP answer, if applicable.

   If the initial `INVITE` and the `ACK` that is returned do not contain the required SDP information, a *media-les*s dialog is established.

   In general, the VoiceXML application starts to execute when the `200 OK` response is acknowledged (the platform receives an `ACK`). However, it is the VoiceXML application itself that decides whether or not it is ready to start execution. In particular, if a media-less dialog has been established, the VoiceXML application will not start executing until the platform receives a re-`INVITE` that includes the SDP information for the caller.

8. Once the VoiceXML application starts executing, it controls the session. The NGI is responsible for driving the Media Control Platform to appropriately execute the VoiceXML application. The NGI performs DTMF recognition, and issues commands to the platform to execute call and media operations.
   - The platform sends and receives SIP `INFO` messages for the following application events:
     - To accept DTMF digits—The SIP `INFO` content type is `application/dtmf-relay`, and content format is `Signal = <digit>`.

       A configurable option (`sip.sipinfodtmf`) enables you to specify whether the application can also use SIP `INFO` messages to send DTMF.

       The DTMF event is generated when the VoiceXML application tries to play an audio file that is named `dtmf_<digit>.vox` or `dtmf_<digit>.wav`.
     - To make a request—The application can specify the content type and content body of the SIP `INFO` message.

- To send or receive data—The application can send data in custom SIP headers. The platform can send information that it receives in SIP `INFO` headers to the NGI, and this information is provided to the application in shadow variables.

- The application uses dialogs to initiate transfers and conferences as required. For more information about how the Media Control Platform performs transfers, see "Transfers" on page 45.

  The NGI supports use of the `userdata` attribute on the `<transfer>` tag, to abstract computer-telephony integration (CTI) data. The NGI exposes CTI userdata to the application in a session variable, `session.com.genesyslab.userdata`.
- The platform provides media services through the Media Server, for operations such as playing prompts and recording audio and video. For more information, see "Media Services".
- For ASR or TTS, the Media Control Platform controls speech resources through the MRCP Client. For more information, see "Speech Services" on page 44.

9. The VoiceXML application can invoke other VoiceXML applications. The NGI is responsible for issuing commands to the Fetching Module to fetch VoiceXML pages and other applications.
10. When a caller disconnects (a `BYE` is received), the platform notifies the Voice XML application (through the `connection.disconnect.hangup` event). If the `BYE` includes a `Reason` header, the value of the `Reason` header is passed verbatim to the application.

    If the application disconnects, the platform generates a `BYE` request.
11. For each VoiceXML session, the Media Control Platform generates call detail records (CDRs), which it sends to the Reporting Server. For more information, see "CDR Reporting" on page 66.
12. For each VoiceXML session, the Media Control Platform sends logs and metrics (VoiceXML application event logs) to the log sinks, from where it sends them to the Reporting Server.

    For more information about metrics, see "Metrics" on page 64. For descriptions of the Media Call Control Platform metrics, see *Genesys Voice Platform 8.0 Metrics Reference.*

Media Services

The Media Server provides the following services:

- Prompt playback
- Utterance recording
- DTMF collection
- ASR streaming (streaming TTS audio to the SIP call, and streaming audio data to an ASR server to perform speech recognition)

- Audio and video encoding and transcoding

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

Selected Features

The Media Server provides advanced features for audio and video services. Features include:

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

Codec Negotiation

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

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

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

The Media Control Platform can support the following codecs:

- pcmu
- pcma
- g726
- g729
- gsm
- amr
- h263
- h263-1998
- tfci
- telephone-event

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

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

For information about the supported audio and video file formats, see "Audio and Video File Formats" on page 289.

Speech Services

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

- For MRCPv1, the MRCP Client uses RTSP to establish MRCPv1 control sessions.
- For MRCPv2, the MRCP Client uses SIP and SDP to create the client/server dialog and set up the media channels to the server. It also uses SIP and SDP, over TCP or TLS, to establish MRCPv2 control sessions between the client and the server for each media processing resource required for that dialog.

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

Grammars

The Media Control Platform generates the following grammars:

- **Hotkey grammars**—Grammars that are used to match the UNIVERSALS properties for the hotwords *Help, Cancel,* and *Exit.* There are separate grammar files for each supported speech engine. The hotkey grammars are installed in the <MCP Installation Path>\grammar\<engine name>\hotkey\ directory.

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

- **Builtin grammars**—The set of builtin grammars provided in the VoiceXML specification. The Media Control Platform provides these because some engines do not support VoiceXML builtin grammars internally on the engine side. The builtin grammars are installed in the `<MCP Installation Path>\grammar\<engine name>` directory.
- **Inline and Implied grammars**—Menu and option grammars that are generated dynamically by the Media Control Platform. These grammars are temporarily stored in the `<MCP Installation Path>\tmp` directory.

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

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

The Microsoft Internet Information Server (IIS) on the Media Control Platform host serves the grammars to the off-board ASR server. Ensure that you configure the IIS application server to serve the required grammars.

Transfers

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

Transfer Types

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

- Blind—The application is detached from the incoming call (and the outbound call if one is involved) as soon as the transfer is successfully initiated. This means that the application is unable to detect the result of the transfer request.
- Consultation (also referred to as *supervised*)—The application is detached from the incoming call when the transfer process finishes successfully. If the transfer process fails, the application retains a relationship with the call. This means that the application is able to report transfer failures.
- Bridge—The application does not detach from the incoming call, unless the incoming call disconnects. Control of the call always returns to the application when the transfer ends, regardless of the transfer result.

Whisper Transfer

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

Whisper transfer enables the platform to continue performing media operations with the callee, and transfer the call out later. For the VoiceXML application developer, whisper transfer enables an application to be written to consult with the callee first, to determine whether the callee will accept the transferred call.

If the callee accepts the call, the transfer proceeds. If the callee rejects the call, the callee is disconnected, and the VoiceXML application can return control to the original caller.

Transfer Methods

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

- **HKF**—Hookflash transfer, using DTMF digits (RFC 2833).
  a. The Media Control Platform sends DTMF digits on the media channel. The platform leaves it to the media gateway or switch to perform the transfer on the network.
  b. Configurable options enable you to specify whether the call will be disconnected by the platform or by the remote end. Otherwise, the call is disconnected after a configured timeout.

  This is a one-leg transfer (in other words, it occupies only one channel on the platform).
- **REFER**—Transfer is based on a SIP `REFER` message (RFC 3515).
  a. The platform sends a `REFER` request to the caller, with the callee (as specified in the VoiceXML application) in the `Refer-To:` header.
  b. The transfer fails if a non-`2xx` final response is received for the `REFER`.

  This is a one-leg transfer (in other words, it occupies only one channel on the platform).
- **BRIDGE**—The Media Control Platform bridges the media path.
  a. The platform sends an `INVITE` request to the callee, and a dialog is established between the callee and the platform.
  b. The transfer fails if a non-`2xx` final response is received for the `INVITE` request.

  This is a two-leg transfer (in other words, it occupies two channels on the platform). The platform stays in the signaling path and is responsible for bridging the two call legs.
- **REFERJOIN**—Consultative REFER transfer (RFC 3891). Also referred to as *`REFER` with replaces* transfer.
  a. The platform sends an `INVITE` request to the callee, and a dialog is established between the callee and the platform.
  b. The platform also sends a `REFER` request to the caller, with the callee's information in the `Replaces` header.
  c. The platform considers the transfer to be successful if it receives a `BYE` from the caller after a `2xx` response for the `REFER`.
  d. The transfer fails if a non-`2xx` final response is received for the `INVITE` request or for the `REFER` request.

  This is a two-leg, or join-style, transfer (in other words, it occupies two channels on the platform).

- `MEDIAREDIRECT`—Media redirection transfer. The Media Control Platform uses SIP to handle call control between the caller and the callee, and the RTP media channel is connected directly between the caller and callee.
  a. The platform sends an `INVITE` request to the callee without SDP.
  b. If the transfer is proceeding, the callee responds with a `200 OK` that includes an SDP offer.
  c. The platform forwards the SDP offer in a re-`INVITE` request to the caller.
  d. The caller responds with a `200 OK` that includes the SDP answer.
  e. The platform forwards the SDP answer to the callee in an `ACK` response.
  f. The transfer fails if a non-`2xx` final response is received for the initial `INVITE` request.

  This is a two-leg transfer (in other words, it occupies two channels on the platform).

The NGI controls the transfer method that is selected, based on the `method` attribute that is specified in the VoiceXML application. If the method is not specified, the default method for the applicable transfer type is used. The default methods are configurable (see `sip.defaultblindxfer`, `sip.defaultconsultxfer`, and `sip.defaultbridgexfer`, on page 167). In addition, configurable parameters enable you to specify whether the Media Control Platform will fall back to the `BRIDGE` or `MEDIAREDIRECT` method if one of the other methods fails.

Because of the actual mechanisms involved, the SIP transfer methods do not all support all the transfer types. Table 2 summarizes SIP transfer method support for the different types of transfer.

Table 2: SIP Transfer Methods and Supported VoiceXML Transfer Types

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

Table 2: SIP Transfer Methods and Supported VoiceXML Transfer Types (Continued)

SIP Transfer Method	Supported Transfer Type	Notes
REFER	• Blind • Consultation	• The default platform method for `type=blind`. • Transfer connect timeout is not supported. • The platform can be configured to send an `INVITE` hold to the caller. • For `type=consultation`, the platform also supports the `NOTIFY` method for notification of the transfer result. If the transfer fails, the platform puts the original caller off hold, and the VoiceXML application proceeds with the caller. • The platform can be configured to send a `BYE` request to the caller, or to wait for a `BYE` from the caller. • For transfer requests from the Call Control Platform, the Media Control Platform sends a `REFER` request to the Call Control Platform, which throws a `dialog.transfer` event to the CCXML application. The `type` attribute for the event is always set to `blind`, whether the request from the VoiceXML application was for `blind` transfer or `consultation` (supervised) transfer.
BRIDGE	• Blind • Consultation • Bridge Whisper transfer supported	• The default platform method for `type=bridge`. • Non-whisper transfers support the `connectwhen=immediate` attribute in the VoiceXML application. If this value is specified, a one-way media path from the callee to the caller is established before the call is connected. • If specified in the VoiceXML application, the platform can continue to support media operations (such as handling DTMF grammars, ASR, transactional recording, and playing transfer audio) during a bridge transfer. • For transfer requests involving the Call Control Platform, if the VoiceXML application uses a `<send>` tag to notify the Call Control Platform about a bridge transfer request, the Media Control Platform sends a SIP `INFO` message to the Call Control Platform.

Table 2: SIP Transfer Methods and Supported VoiceXML Transfer Types (Continued)

SIP Transfer Method	Supported Transfer Type	Notes
REFERJOIN	• Blind • Consultation Whisper transfer supported	• The default platform method for `type=consultation`. • The platform can be configured to send an `INVITE` hold to the caller. • If the transfer fails, the platform puts the original caller off hold, and the VoiceXML application proceeds with the caller. • The platform can be configured to send a `BYE` request to the caller and then the callee, or to wait for a `BYE` from the caller. • For whisper transfer, if the callee rejects the transfer request, the platform sends a `BYE` to the callee to disconnect the call. • Non-whisper transfers support the `connectwhen=immediate` attribute in the VoiceXML application. If this value is specified, the media path is established between the caller and the callee as soon as the media session is ready.
MEDIAREDIRECT	• Blind • Consultation • Bridge Whisper transfer supported	• For whisper transfer, a media channel is established between the callee and the Media Control Platform for the consultative part of the transfer, if necessary. If the callee rejects the request, the platform sends a BYE request to the callee to disconnect the call. The media path and interaction between the platform and the caller then resumes. • If the caller disconnects during the transfer (platform receives a `BYE`), the platform sends a `BYE` to the callee to disconnect the call. • If the callee disconnects during the transfer, the platform updates the caller's media path back to the platform, using a new re-`INVITE` if necessary.

Implications of Transfer Method–Transfer Type Combinations

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

Conferencing

Conferencing is, in essence, a special case of bridge-type transfer.

Calls join a conference directly, by specifying the conference bridge identifier (conference ID) in the SIP `Request-URI` in NETANN format:

`sip.conf=<conf ID>@host.com`

The Conference application module handles calls and manages call interactions with the conference bridge for NETANN conferencing. Platform-level configuration options (in the `conference` configuration section) support standard NETANN conference requirements, such as configurable participant roles (talk-only, listen-only, full duplex), audio gain parameters, and the video output algorithm (first, loudest, or no video).

Debugging VoiceXML Applications

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

If the real-time debugger is enabled (see `vxmli.debug.enabled` on page 178), information about calls is passed between the NGI and the debugger client in SIP `INVITE` and `18x` messages.

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

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

How the Call Control Platform Works

This section provides information about the following topics, to explain how the Call Control Platform performs its role in a GVP deployment:

About the Call Control Platform

The Call Control Platform is composed of:

- A core executable (`ccpccxml.exe`), of which one of the main processes is the HTTP event I/O processor (`ioproc`).

- The CCXML Interpreter (CCXMLI), which is a static library (`ccxmli.lib`) running in-process in the Call Control Platform.

The Fetching Module, which is a separate GVP component, co-resides on the Call Control Platform host. The Call Control Platform and the Fetching Module share memory.

Operational Overview

The Call Control Platform receives requests for call control or conference services for incoming connections from the Resource Manager in the form of SIP `INVITE` messages. The platform can conference, transfer, or redirect calls using other kinds of SIP messages (see "Transfers" on page 45). The platform can also initiate outbound calls by sending SIP `INVITE` requests through the Resource Manager or directly to the destination.

For more detailed information, see the *Genesys Voice Platform 8.0 CCXML Reference Manual*.

Incoming Connections

The Call Control Platform handles service requests for incoming connections as follows:

1. The Call Control Platform receives a SIP `INVITE` from the Resource Manager.
2. The Call Control Platform assigns a device profile to the connection. For more information about device profiles, see "Device Profiles" on page 53.
3. The Call Control Platform can accept, reject, or redirect the connection. Configuration options allow you to customize some of the SIP responses that the Call Control Platform sends to the Resource Manager for the respective events (see "Customizing SIP Responses" on page 116 and Table 64 on page 279).
4. For connections that it accepts, the Call Control Platform sends an HTTP/HTTPS or file retrieval request to the Fetching Module to fetch the initial page.
   - Because the Resource Manager has modified the SIP request to insert service-prerequisites for the IVR Profile, the SIP `Request-URI` includes a `ccxml` parameter that specifies the URL of the initial page of the required CCXML application.
   - If the `Request-URI` from the Resource Manager does not include an initial page URI, or if the Call Control Platform is being used in a deployment without the Resource Manager, the Call Control Platform uses the default that has been configured for the platform (`ccpccxml.default_uri`).

- Call Control Platform HTTP requests comply with Hypertext Transfer Protocol (HTTP) 1.1. For secure HTTP (HTTPS), the Call Control Platform supports HTTP over Secure Socket Layer (SSL) 3.0 and HTTP over Transport Layer Security (TLS).
- The HTTP/HTTPS fetch method (`get` or `post`) depends on whether the `method` parameter was specified in the SIP `Request-URI`. The default is `get`.
- Other parameters that are supported in the HTTP/HTTPS fetch are:
  - `namelist`—A list of ECMAScript variables whose values are submitted as part of the request.
  - `enctype`—The encoding type to be used for `namelist` data, if the `method` is `post`. The only supported value is `application/x-www-form-urlencoded`.

  For both `get` and `post` methods, `namelist` variables must be encoded in the URI query string in the `form url-encoded` format (as described in the HTML 4.01 specification). If the `get` method is used, the `namelist` variables are appended after a `?` (question mark) character.

5. The CCXMLI compiles and interprets the initial page, and all subsequent pages, for the Call Control Platform to run the application.
   - A CCXML page may transition to another page as it executes, but only one page is executed at a time.
   - The CCXML session may create and interact with other entities:
     - Connections (other SIP sessions)
     - Dialogs (VoiceXML sessions)
     - Conferences
     - Other CCXML sessions
   - To improve performance, the Call Control Platform enables the root page of the initial page to be cached. Caching is not relevant for the initial page itself, or for other pages, because CCXML application pages are session-specific. For more information about how the Fetching Module caches pages, see “Caching” on page 56.
   - The Call Control Platform supports receiving DTMF events in SIP `INFO` messages, and propagates the events and data to the CCXML application and other connections.
6. The Call Control Platform uses the Media Control Platform to provide bridging, conference, and transcoding services. It may also perform implicit conferencing and transcoding if the endpoints of a connection do not have the required bridging capabilities or support required codecs. The Call Control Platform obtains these services by sending SIP requests through the Resource Manager.

   Dialog-initiated transfers between CCXML sessions, or between CCXML and VoiceXML sessions, are application driven, with the SIP messaging going through the Resource Manager in SIP `INFO` messages. For more information about transfers, see “Transfers” on page 45.

7. For each CCXML session, the Call Control Platform generates call detail records (CDRs), which it sends to the Reporting Server. For information about the CDR attributes, see "CDR Reporting" on page 66.
8. For each CCXML session, the Call Control Platform sends logs and metrics (CCXML application event logs) to the log sinks, from where it sends them to the Reporting Server.

   For more information about metrics, see "Metrics" on page 64. For descriptions of the Call Control Platform metrics, see *Genesys Voice Platform 8.0 Metrics Reference.*

Note: The Call Control Platform does not support operational reporting (OR).

9. If configured to do so (see `ccxmli.debug.data.*` and `ccxmli.platform.save.*` parameters), the Call Control Platform captures fetch data for CCXML and ECMAScript files, to aid in debugging CCXML applications.

Outgoing Connections

The Call Control Platform can start a new CCXML session if it receives a session creation request directly from a web server, to place an outbound call. Alternatively, the Call Control Platform may place an outbound call within the context of an existing session.

The CCXML application uses the `<createcall>` tag to create the connection, and specifies the destination of the call (in the `dest` attribute) as a SIP URI. The value of the `dest` attribute is used in the `To:` header of the SIP `INVITE` that the Call Control Platform sends to the Resource Manager to place the call. The Resource Manager, in turn, forwards the request to another SIP Proxy that may have been configured in a `routeset` for the Call Control Platform, or to the SIP Server.

Note: You can override the default outbound proxy configured in the route set by specifying an `outboundproxy` hint in the CCXML `<createcall>` tag.

Device Profiles

The Call Control Platform interacts with a variety of SIP devices, all of which have different characteristics and features. The concept of a *device profile* enables the Call Control Platform to interact with a wide range of devices, even though they might differ in the way that they support SIP.

The *device profile* defines a number of properties that describe the SIP and SDP capabilities of a class of devices. The Call Control Platform uses a device profile when it performs call control operations (for example, `<join>` and

<accept>). The Call Control Platform assigns a device profile to any SIP device with which it interacts. The properties of the device profile then govern how the Call Control Platform interacts with the SIP device.

Device Profile Configuration File

The properties of the various device profiles are defined in a text file in the Call Control Platform `config` directory (`<Call Control Platform Installation Directory>\config\ccpccxml_provision.dat`). For more information about the properties that are defined for CCXML device profiles, see Table 33 on page 189.

Assigning Device Profiles

The Call Control Platform assigns device profiles as follows:

- Incoming connections—The Call Control Platform tries to match the SIP header from the incoming SIP `INVITE` with the value of the `SIP Header Name` property that is defined in the device profile configuration file, by order of precedence that is also specified in the configuration file. (By default, the SIP header that the platform looks for is `User-Agent`.) If it cannot match the SIP header, the Call Control Platform uses the `Default Inbound` profile that has been provisioned (see "Default Device Profiles").
- Outbound connections, dialogs, and conferences—The Call Control Platform matches CCXML hints with the value of the `Device Profile Name` property that is defined in the device profile configuration file. If it cannot match the hint, the Call Control Platform uses the `Default Outbound`, `Default Dialog`, or `Default Conference` profile that has been provisioned (see "Default Device Profiles").

Default Device Profiles

By default, VP Call Control Platform 8.0 is provisioned with the following device profiles for SIP devices, by order of precedence:

- Cisco Gateway
- Audiocodes Gateway
- Convedia Media Server
- X-Lite
- Brooktrout Snowshore
- GVP MCP
- Audiocodes MP 104
- Default Inbound
- Default Outbound
- Default Conference
- Default Dialog

For the property values that have been defined for the preprovisioned device profiles, see Appendix D on page 304.

If your deployment uses SIP devices that are not adequately represented by the default device profiles, you must provision additional device profiles or else modify an existing device profile. For more information, see "Configuring Device Profiles" on page 188.

How the Fetching Module Works

This section provides information about the following topics, to explain how the Fetching Module and the Squid Caching Proxy perform their role in a GVP deployment:

About the Fetching Module

The Fetching Module is an executable (`pwproxy.exe`) that is responsible for fetching VoiceXML and CCXML file and HTTP/HTTPS resources.

The Fetching Module uses shared memory to communicate with the NGI (on the Media Control Platform host) or CCXMLI (on the Call Control Platform host). Therefore, even though they are separate processes, the Fetching Module can pass fetch results and other information back to the interpreters very efficiently.

Note: The Fetching Module must be started before the Media Control Platform and the Call Control Platform are started. If the Fetching Module stops for any reason, the Media Control Platform and the Call Control Platform must also be stopped, then restarted after the Fetching Module has restarted.

Fetching Module Caching

To improve performance, the Fetching Module performs caching as follows:

- The Fetching Module itself performs some limited in-memory caching, which is not HTTP/1.1 compliant.
- If the Fetching Module determines that it cannot serve the request from its in-memory cache, it goes to the Squid Caching Proxy to try to fetch the content. The Squid Caching Proxy performs HTTP/1.1-compliant caching.
- If Squid determines that it cannot serve the content from its cache, it goes to the Web Server to try to fetch the content.

For more information, see “Caching”.

Caching

Audio and video recordings are common in VoiceXML documents, and they can be very large. Their content is also mostly static. Therefore, using cached content significantly improves performance.

Unlike visual browsers, there are no end-user controls in the VoiceXML interpreter context to enable stale content to be updated or refreshed. Rather, the VoiceXML document itself enforces cache refresh, through appropriate use of the `maxage` and `maxstale` attributes. However, these attributes interact with other proxy settings and HTTP cache-control mechanisms at various levels, as described in the following subsections.

Non-HTTP/1.1-Compliant Caching

The Fetching Module caches documents in-memory, in accordance with configurable maximum age and URL substring parameters (see the `iproxy.*` parameters described in "Important Fetching Module Configuration Options" on page 198). For the algorithm that the Fetching Module uses to determine whether it will use a cached version in its own memory, see "Fetching Module Caching Algorithm" on page 315.

Because this level of caching is non-compliant with HTTP/1.1, use it carefully. If you require strict HTTP/1.1 compliance in your deployment, set the `iproxy.*` parameters so that this in-memory caching is turned off.

HTTP/1.1-Compliant Caching

The Fetching Module uses a caching proxy (Third-Party Squid) for HTTP/1.1-compliant caching. The caching proxy generates HTTP/1.0 requests, but supports HTTP/1.1 caching functionality.

The caching policies of the VoiceXML interpreter context adhere to the cache correctness rules of HTTP/1.1. In particular, the `Expires` and `Cache-Control` headers are honored.

Caching Policies

- The application server maintainer/content provider can provide guidelines for content expiry using the `Cache-Control` and `Expires` HTTP response headers.
- If these headers are not present, Squid will use heuristics to generate expiry times.
- The application developer can deterministically control the caching behavior of application resources, by using the `maxage` and `maxstale` attributes for each URI-related VoiceXML tag. This behavior includes forcing a validation of the current cache contents (using `maxage`), and accepting expired cache contents (using `maxstale`).

- The platform maintainer can control cache resource usage by the way in which Squid is configured.

Caching Behavior

The primary impact of the caching policies is that the client has control over what it will accept from the cache, even if the server has specified an `Expires` header or `maxage/maxstale` attributes, or if the caching proxy has generated an expiry time itself.

- Documents from the web server will be delivered with zero, one, or both of the response headers.
- If an `Expires` header is present, it is used to set the expiry time of the object in the cache.
- If the `Expires` header is not present, Squid applies a heuristic to set an expiry time.
- If a `Cache-Control` header is in the response, it will be used to control expiry times, and will override an `Expiry` time if also provided.

For the algorithm that Squid uses to determine whether it will fetch a fresh version, see "Squid Caching Algorithm" on page 316.

Note: It is an optimization to perform a *get if modified* (the request includes an `If-Modified-Since` [IMS] header) on a document still present in the cache when the policy requires a fetch from the server. Squid does perform this optimization.

Maxage and maxstale

VoiceXML enables the application developer to control caching policy for each use of each resource.

The application developer can specify `maxage` and `maxstale` attributes for each resource-related element. These attributes provide fine-grained control over when documents are returned from the cache, and when they are fetched from the origin server. For example:

- Setting `maxage` to a non-zero value means that Squid might be forced to get a fresh copy of a resource that may not yet have expired in the cache. Setting `maxage` to zero means that Squid will unconditionally be forced to get a fresh copy.
- Using `maxstale` enables the application developer to state that an expired copy of a resource that is not too stale (according to the rules of HTTP/1.1) may be used. This can improve performance by eliminating a fetch that would otherwise be required to get a fresh copy. This is especially useful for application developers who might not have direct server-side control of the expiration dates of large static files.

Notes: Like other caching proxies that support `maxage` and `maxstale`, Squid does not delete items from the cache after their expiry time, unless other cache requirements (such as memory or disk usage limits) dictate such action. The reason for this is that the client may specify that an expired resource is acceptable.

Some resources may be addressed by URIs that name protocols other than HTTP and that do not support the `maxage` and `maxstale` attributes. If the protocol does not support the notion of resource age, the interpreter context computes the age of a resource from the time it was received. If the protocol does not support the notion of resource staleness, the interpreter context considers the resource to have expired immediately upon receipt.

The `maxage` and `maxstale` attributes interact with server-provided expiry times to produce a variety of caching behaviors. Table 3 describes some sample behaviors.

Table 3: Using maxage and maxstale Attributes

Desired Behavior	maxage	maxstale	Notes
Client control over expiry	<desired_expiry>	0	• Caching based on `Expires` header. • Refetch based on `maxage` and `maxstale`. • Uses `IMS`.
Expired document acceptable	<large_value>	<desired_maxstale>	• Caching based on `Expires` header. • Refetch after Expiry time plus `maxstale`. • Uses `IMS`.

Maxage, Maxstale, and the Initial Page

For the initial page request, the GVP Session ID is submitted as part of the URL. Because this ID is unique, the requested URL appears unique, so the `maxage` and `maxstale` parameters have no meaning for that page. However, they do have meaning for the initial root page.

Configuration parameters in the Media Control Platform set the values of the `maxage` and `maxstale` parameters for the initial root page (see `vxmli.initial_request_maxage` and `vxmli.initial_request_maxstale`). For both parameters, the default value is `-1` (undefined).

Determining Expiry Time

Web servers may or may not return an `Expires` response header to the client.

- If the Web Server does return an `Expires` response header, this expiry time is used in the cache refresh algorithm.
- If, instead, the Web Server provides expiry information as part of a `Cache-Control` header (using `maxage/maxstale`), this information will be used to control cache expiry.

Expiration Model

Squid uses a Refresh-Rate model, rather than a time-based expiration model. Objects are not purged from the cache when they expire. Instead of assigning a "time-to-live" when the object enters the cache, Squid checks freshness requirements when objects are requested.

- If an object is "fresh", Squid gives it directly to the client.
- If an object is "stale", Squid makes an `If-Modified-Since` request for it to the Web Server.

For the algorithm that Squid uses to determine freshness or staleness, see "Squid Expiry Time Algorithm" on page 316.

Squid Configuration File

The Squid configuration file (`C:\squid\etc\squid.conf`) controls configuration of the caching proxy. The configuration file is a text file that contains pairs of keywords and values (with no equals [=] sign). For example:

```
http_port 3128
```

defines port `3128` as the TCP port that the caching proxy will use for receiving requests.

In general, the default Squid configuration file should be suitable for most installations. However, you might need to modify the Squid configuration file for the following reasons:

- 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.

For more information about modifying the Squid configuration file, see "Configuring the Squid Caching Proxy" on page 200.

For details about all Squid configuration items, see the *Squid Configuration Guide* (`http://squid.visolve.com/squid24s1/contents.htm`).

Changes to the Squid configuration file are not reflected in the running configuration immediately.

Squid Log Files

The caching proxy logs can provide useful information to assist you in identifying performance issues or resolving VoiceXML or CCXML application problems.

Access Logs

The Squid `access.log` file is in the following location:

```
C:\squid\var\logs\
```

The access log contains one entry for each HTTP (client) request and each Inter-Cache Protocol (ICP) Query. HTTP requests are logged when the client socket is closed. The native `access.log` has ten fields. A single dash (-) indicates unavailable data.

For details about the fields in the Squid `access.log`, see "Squid Access Logs" on page 317.

Managing the Cache Manually

Table 4 summarizes the commands to force refreshes of cached objects, purge an object from the cache, or clear the entire cache. Issue these commands in the `cmd` console window on the Media Control Platform or Call Control Platform host whose cache you want to manage.

Table 4: Manual Cache Management Commands

Objective	Command
Refresh an object in the cache.	`C:\squid\bin\squidclient -r <uri>` where `<uri>` is the full URI of the object you want to refresh.
Purge an object in the cache.	`C:\squid\bin\squidclient -m PURGE <uri>` where `<uri>` is the full URI of the object you want to purge.

Clear the entire cache.	`C:\squid\bin\sbin\squid -k shutdown -n SquidNT`
	`echo '' > C:\squid\var\cache\swap.state`
	`net start SquidNT`

Logging and Reporting

EMS Reporting refers to the GVP logging and reporting feature, which provides the following services:

- Accumulates key measurements and data describing the calls being processed by the deployment.

- Delivers infrastructure for reliably delivering data to a relational backend.
- Provides services for near real-time reporting on operational aspects of the deployment.
- Provides historical reporting on VoiceXML and CCXML application usage.

EMS Reporting leverages the Management Framework role-based security system to offer consistent, permissions-based access to data.

This section provides information about the following EMS Reporting topics:

EMS Reporting Architecture

EMS Reporting uses a client/server architecture. Figure 3 illustrates the EMS Reporting architecture.

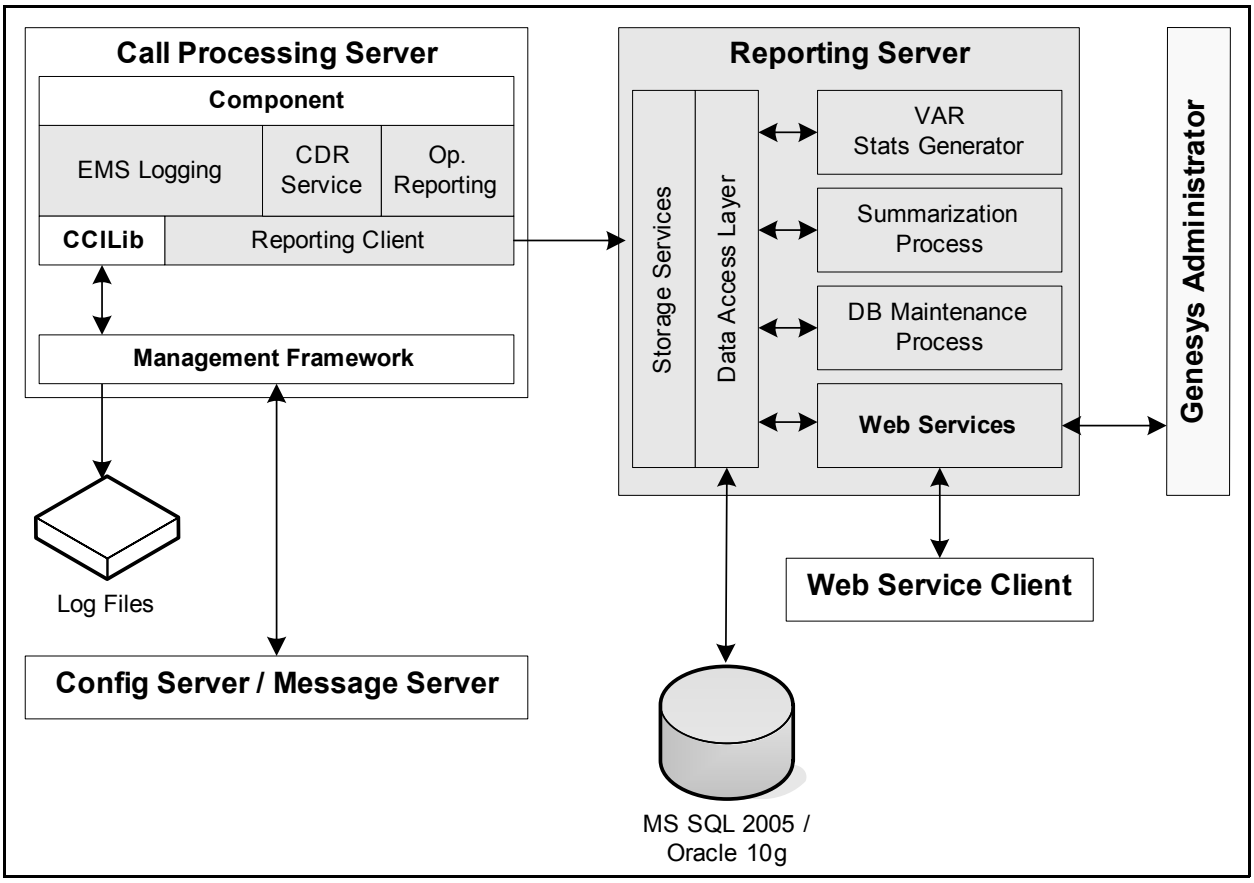


Figure 3: EMS Reporting Architecture

EMS Logging

- An EMS Logging interface on each component, or GVP `Application`, enables the component to log events relating to component activity. For more information, see "EMS Logging" on page 63.
- Additional interfaces on each call-processing component provide the following services to accumulate data:

CDR Reporting

- Call Detail Record (CDR) Service, which enables the component to submit and update CDRs to the Reporting Server. For more information about CDRs, see "CDR Reporting" on page 66.

OR Service

- Operational Reporting (OR) Service, which accumulates call arrival, call length, and call peak statistics. (Applicable for the Resource Manager and Media Control Platform only.) For more information about OR reporting, see "OR Service" on page 68.

The call-processing components are the Resource Manager, Media Control Platform, and Call Control Platform `Applications`.

VAR

- A `<log>` tag interface supports the Voice Application Reporter (VAR) reporting product, which is delivered by the Reporting Server. The `<log>` tag interface enables users to demarcate their VoiceXML applications into logical transactions, and assign success or failure to individual transactions or to a call as a whole. The `<log>` interface also provides a means of attaching application-specific data (such as call notes and custom name-value pairs) to calls.

  The Reporting Server accumulates summary statistics based on the processing of appropriately formatted VAR `<log>` tags. The summary statistics that it derives are accessible through web services and the Genesys Administrator.

  For more information, see "VAR Metrics" on page 65 and Table 73 on page 301.

Reporting Client

- A Reporting Client on each call-processing component is responsible for reliably delivering the accumulated data to the Reporting Server. In GVP 8.0, the data is transported over TCP. For more information, see "Reporting Client" on page 68.

Reporting Server

- The Reporting Server receives and persists data submitted from the Reporting Clients on the call-processing servers. There can be only one Reporting Server in a GVP deployment. For more information about the Reporting Server, see "Reporting Server" on page 68.

EMS Logging

The EMS Logging API enables GVP components to raise logging events at two levels:

- At the level of the component, or GVP `Application`—these are referred to as *logs*.
- At the level of the VoiceXML or CCXML application—these are referred to as *metrics*.

Logs

Logs include three important elements by which they can be filtered: Severity, Module ID, and Specifier.

Severity

GVP components, like most other Genesys components, can raise events at the following levels of severity (in descending order of severity):

0 = Critical

1 = Error

2 = Warning

3 = Note

4 = Info

5 = Debug

Module IDs Each GVP component is composed of one or more Application Modules, each of which is assigned a Module ID. The component logically organizes the logs it emits by Module ID.

Specifiers A specifier is a number that uniquely identifies a given event that is logged by a given module.

For a list of Module IDs and specifiers that are used in GVP 8.0, see Appendix A, "Module and Specifier IDs," on page 255.

Additional Log Data EMS Logging associates a UTC timestamp, to millisecond precision, with each logging event when it is raised.

Logs for call-processing component events that are associated with a call session include the GVP Component ID and the GVP Session ID, as well as the UTC timestamp. Metrics can therefore be mapped to CDRs, which provide further information, such as call start time, call end time, the Application ID (DBID of the IVR Profile), and local and remote SIP URIs.

For more information about GVP IDs, see "GVP Identifiers and SIP Headers" on page 71.

Log Delivery Log events are delivered to one or more log sinks for the component (see "Log Sinks"), and then sent on to Management Framework or the Reporting Server.

By default, log files are located in the following folders:

```
C:\Program Files\GCTI\gvp\<IP name>\<Your component application name>\logs\
```

Metrics

Metrics describe application-level events, and have no severity.

Each metric has a unique type identifier (for example, `start_session`) and is associated with a specific VoiceXML or CCXML session ID. The body of the metric is defined by the component. For example, the body of a metric can be a text string that consists of a number of pipe-delimited parameters (such as ANI|DNIS|SIP `Request-URI`), encoded in UTF-8.

Metrics Examples The following are examples of the kinds of metrics that are logged. For full details about the metrics that are available in GVP 8.0, see the *Genesys Voice Platform 8.0 Metrics Reference.*

- The Media Control Platform logs an `INCALL_BEGIN` metric when an inbound call is accepted.
- The NGI logs a `PROMPT` metric when it starts to play back a prompt queue.
- The time to fetch a VoiceXML page is measured and logged.

VAR Metrics

VAR metrics are events that are generated by the Media Control Platform when it encounters VAR-specific `<log>` tags in the VoiceXML applications. The VAR-specific `<log>` tags have the prefix `com.genesyslab.var.`

For the metrics that the Media Control Platform generates when the NGI executes a VAR-specific `<log>` tag, see Table 73 on page 301.

For more information about using `<log>` elements in VoiceXML applications, see the *Genesys Voice Platform 8.0 VoiceXML 2.1 Help.*

Metrics Delivery

Metrics are delivered to the log sinks for the component (see "Log Sinks"). *Upstream metrics,* also referred to as *call events,* are metrics that are configured to be sent to the Data Collection Sink (`DATAC`) and then sent on to the Reporting Server for storage and reporting purposes. The Media Control Platform and the Call Control Platform are the only sources of upstream metrics.

Log Sinks

Every component that uses logging has configurable access to one or more log sinks, which receive a real-time stream of logs or metrics, as defined by configurable filters (see `ems.logconfig.<Sink Name>` on page 106 and `ems.metricsconfig.<Sink Name>` on page 107).

The log sinks are DLLs that are loaded dynamically at runtime. For the default log sinks that are attached to each component, see `ems.log_sinks` on page 105.

The log sinks enable EMS Reporting to implement upstream reporting, integrate with Management Framework, and accumulate summary statistics that are used by the Reporting Server. *Upstream reporting* refers to the ability of components to send a configured subset of metrics to the Reporting Service for storage and reporting purposes.

The following log sinks are available in GVP:

- `MFSINK`—The Management Framework Adaptation Sink. `MFSINK` connects the GVP and Management Framework logging systems, through CCILib, for file- and network-based logging. Configurable parameters in the `log` configuration section for each GVP component determine what Management Framework does with the logs—for example, writing them to file or delivering them to Message Server.
- `DATAC`—The Data Collection Sink. `DATAC` derives resource-specific summary statistics, and delivers summary statistics and metrics to the Reporting Server, where they can be queried through the Call Events reporting service. (Not applicable for the Resource Manager or Fetching Module.)

  A configurable option on the Media Control Platform and Call Control Platform (`ems.dc.default.metricsfilter`) enables you to specify which of the metrics delivered to `DATAC` will be forwarded to the Reporting Server.

- TRAPSINK—The SNMP integration sink. For GVP components that have been configured to raise traps, TRAPSINK forwards log messages to the Management Data Agent library, which the GVP process uses to implement the applicable MIBs.

Depending on the configured filters, a particular log or metric may be directed to more than one log sink for the component, or to none. If a given log event does not match the configured event types for that component's log sinks, the log event is silently discarded.

The log sinks themselves may generate log events, and can therefore have one or more log sinks attached to them. For example, in GVP 8.0, DATAC has an MFSINK, which enables DATAC logs to be delivered to Message Server or to file.

Calculated Statistics

The data in DATAC includes statistics that are calculated from metrics data.

CDR Reporting

CDRs are records that describe key attributes of a call session that is being, or has been, processed by the deployment.

The CDR Service on the Resource Manager, Media Control Platform, and Call Control Platform enables the component to submit and update CDRs to the Reporting Server in near real-time.

The Reporting Server correlates the CDRs based on the GVP Session-ID (see "Session Identifiers" on page 72).

The intervals at which the Reporting Client submits CDRs to the Reporting Server depends on configuration (see ems.rc.cdr.batch_size on page 108).

CDR Attributes

The CDRs share a common set of attributes that, at a minimum, the component must include. In addition, the Media Control Platform and Call Control Platform include certain attributes that are specific to the component type.

Common CDR Attributes

All components include the following attributes in the CDRs that they submit:

- Session start and end time.
- IDs for the VoiceXML or CCXML application; Media Control Platform or Call Control Platform session; Resource Manager session; and overall Genesys session (UUID).
- Call type—Available types are:
  - Inbound (1)—for Resource Manager and Media Control Platform
  - Outbound (2)—for Resource Manager and Media Control Platform

  - `Bridged (3)`—for Media Control Platform
  - `Unknown (4)`—for Resource Manager
  - `New Call (5)`—for Call Control Platform
  - `Create-CCXML (6)`—for Call Control Platform
  - `External (7)`—for Call Control Platform
- Local-URI—The URI that identifies the local service that was delivered.
- Remote-URI—The URI of the party with whom the dialog was conducted. The platform obtains this information from the `From` header on an inbound call or the `Request-URI` on an outbound call.

Media Control Platform–Specific Attributes

The Media Control Platform includes the following additional attribute in CDRs:

- For bridged calls, the parent Component ID (in other words, the Media Control Platform ID of the call session that originated the bridged session).

Call Control Platform–Specific CDR Attributes

The Call Control Platform includes the following additional attributes in CDRs:

- How the session was started:
  - `EXTERNAL`—The session was created through the HTTP session creation I/O processor.
  - `CREATECCXML <parent-ccxml-session-id>`—The session was created by a `<createccxml>` tag from a parent session.
  - `NEWCALL <call-params>`—The session was created because of an inbound call, where `<call-params>` records relevant parameters (for example, the UUID on the connection).
- The reason the CCXML session ended:
  - `EXIT`—The `<exit>` tag was executed.
  - `KILL`—The session was terminated by a `ccxml.kill.unconditional` event or an `unhandled ccxml.kill` event.
  - `DOCINIT`—The session ended because an error was encountered during document initialization.
  - `ERROR`—The session was ended by an unhandled error event.
  - `SYSERR`—The session aborted because of an internal error.
- An ID indicating the source of the session:
  - For calls started by a connection, the connection ID of the initiating call.
  - For externally created calls, the `eventsource` URI.
  - For forked sessions, the Component ID of the parent session.

OR Service

The OR interface enables the Resource Manager and Media Control Platform components to accumulate statistics about call arrivals and call peaks, and submit them to the Reporting Server, through the Reporting Client.

- Call arrivals—Counts are derived from the CDRs as they are submitted or updated.
- Call peaks—Statistics are derived from counts of the maximum number of concurrent calls that are observed within a given five-minute time period.
  - The Resource Manager submits peaks for the deployment as a whole and for each VoiceXML application processed.
  - The Media Control Platform submits peaks for itself alone.

The Reporting Client submits OR data to the Reporting Server at a configurable interval (see the `ems.ors.reportinginterval` parameter). The default is every minute.

Reporting Client

The Reporting Client on each component provides reliable delivery of `DATAC` logs and metrics, CDRs, and OR statistics to the Reporting Server.

The Reporting Client persistently queues data when the Reporting Server is unavailable, and uses exponential back-off to attempt to reconnect to the Reporting Server. Data that has been submitted to the Reporting Client will eventually be sent to the Reporting Server, even in the face of extended Reporting Server outages.

Note: Data in memory is lost if the call-processing component shuts down unexpectedly. Data is persisted to disk only if the Reporting Client is unable to successfully deliver the data to an available Reporting Server.

The Reporting Client can be configured to send CDRs and metrics in batches, to improve performance. However, this can result in slight delays in data delivery. For more information, see the description of the `ems.rc.cdr.batch_size` parameter on page 108.

Reporting Server

As shown in Figure 3 on page 62, the services that the Reporting Server provides include the following:

- Storage services—Logs and metrics that are delivered by the Reporting Client on the components are stored in the GVP reporting database, where they can be queried by Reporting Web Services.

- Reporting Web Services—HTTP web services return XML that conforms to well-defined schemas. XML-based reports display on the `Monitoring` tab in the Genesys Administrator. For more information, see "Reporting Web Services" on page 69.
- VAR Stats Generator—The Reporting Server computes VAR statistics based on the VAR-specific metrics it receives (see "VAR Metrics" on page 65).
- Summarization process—Every hour (on the half-hour), the Reporting Server rolls five-minute statistics into higher-level hourly, daily, weekly, and monthly summaries. The process summarizes VAR and OR data only.

  For performance reasons, the process does not start summarizing for a period until that period has ended. For example, a monthly summary for January will not be created until the start of February.

  The Reporting Server can derive summaries on the fly. (For example, you can request a monthly report for January before January has completed.) However, this puts more load on the database than when the regular summarization process derives summaries from precomputed data.
- Database Maintenance Process (DBMP)—The DBMP purges old data in accordance with data retention policies. By default, the process runs once daily, at a configurable time. The data retention policies are also configurable. For more information, see "Configuring Database Retention Policies" on page 206.

Reporting Web Services

Reporting Web Services can be deployed over HTTP or HTTPS. By default, Reporting Web Services are deployed at the following URL:

`http://<Reporting Server host name>:8080/ems-rs`

The reporting services return results (reports) as XML documents that conform to available RelaxNG schemas. Therefore, the EMS Reporting data is available to third-party reporting products as well as through the `Monitoring` tab in the Genesys Administrator GUI. For detailed information about the XML schemas for GVP Reporting web services, contact Genesys Technical Support.

GVP can leverage Genesys Management Framework access control mechanisms to control access to reporting data. If this feature is enabled, users must provide credentials (login ID and password) with reporting service requests, and the credentials are validated with Management Framework to determine if the user is authorized to access the reporting data. For information about configuring access control for GVP Reporting, see "Controlling Access to Reporting Services" on page 210.

Report Categories

Reporting services are grouped into the following categories:

- Real-time
- Historical

- VAR

For more information about the GVP reports that are available in these categories, see the Monitoring part of this guide, starting on page 217.

SNMP Monitoring

GVP 8.0 supports Simple Network Management Protocol (SNMP) monitoring for the Resource Manager, Media Control Platform, Call Control Platform, and Fetching Module components. Using SNMP in a GVP deployment is optional.

To use SNMP in your GVP deployment, you must first install the Genesys SNMP Master Agent on each GVP host, as described in the *Genesys Voice Platform 8.0 Deployment Guide.*

Each GVP `Application` acts as an AgentX subagent, and connects to the Genesys SNMP Master Agent on the host. Multiple subagents can register to the same master agent.

The Master Agent handles all queries from the Management Data GUI and any Network Management Station (NMS), and sends AgentX queries to the respective subagent (in other words, the GVP processes).

Traps, which are generated from logs, flow from the subagent to the Master Agent, and then to trap destinations as configured on the Master Agent.

The traps are defined in the GVP Management Information Bases (MIBs), which are available in their own installation package (IP). For more information about the GVP MIBs, see the *Genesys Voice Platform 8.0 Troubleshooting Guide.*

Secure Communications

GVP 8.0 supports the following protocols for secure communications:

- Secure SIP (SIPS)—SIP over the Transport Layer Security (TLS) protocol, for call control and resource management messaging between the Resource Manager and the Media Control Platform and Call Control Platform resources.

  GVP supports TLSv1.
- Secure HTTP (HTTPS)—HTTP over Secure Socket Layer (SSL), for fetch communications among the NGI/CCXMLI, Fetching Module, and Squid caching proxy, and the Fetching Module and web application server. The Reporting Server also supports HTTPS, for receiving and responding to authenticated report requests from the web server.

  GVP supports SSLv2, SSLv3, and SSLv23.

- Secure RTP (SRTP)—A profile of RTP that provides encryption and authentication of audio and video data in RTP streams between the Media Control Platform and the Media Gateway.

  SRTP encryption keys and options are exchanged in SIP `INVITE` and response messages, preferably using SIPS.

The GVP components ship with a generic private key and SSL certificate, and default SIP transports for TLS are configured in the component `Application` objects. It is therefore possible to implement basic security out of the box. However, for more stringent security, Genesys recommends that you obtain your own SSL keys and certificates.

For more information about obtaining SSL keys and certificates, and configuring the GVP components to use SIPS, HTTPS, and SRTP in the GVP deployment, see "Enabling Secure Communications" on page 93.

Considerations and Usage Notes

Before you implement widespread use of HTTPS in your GVP deployment, consider the following:

- Complete use of SSL will have an impact on platform performance and capacity. Lags in fetch times and high CPU usage are normal when SSL is used, because the web server must encrypt every byte of data, and then the platform must decrypt the received data. In addition, an SSL handshake takes place between the web server and the platform before data transmission starts.
- Data fetched with HTTPS will not be cached.
- Before you use HTTPS to reference grammars, ensure that your ASR engine supports it.
- Be aware that, if a VoiceXML page was fetched with `https` and resources within the page (such as audio files, grammars, and scripts) are referenced with a relative URI, the full URI for the resource will also use `https`. If you want to use HTTP to fetch a resource from a page that was fetched with HTTPS, ensure that the VoiceXML page explicitly references the resource as an `http` URI.

GVP Identifiers and SIP Headers

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

- Session Identifiers
- Application Identifiers (see page 72)

Session Identifiers

There are three types of session identifiers that are used to track, co-ordinate, and report on GVP sessions. Table 5 describes the session identifiers and the SIP extension headers in which the ID information is captured.

Table 5: GVP Session Identifiers and SIP Headers

Session ID	Description	SIP Header
Genesys CallUUID	The Universally Unique Identifier (UUID) that is generated by T-Server or SIP Server 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) 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.	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.
GVP Component ID	The ID that is generated by the GVP component 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).	`gvp.ccp.sessionid`—For the Call Control Platform only, when it sends a request to initiate an outbound session and the Resource Manager session has not been assigned yet. This allows 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 6 describes the identifiers that GVP uses for both kinds of applications, and the SIP extension headers in which the ID information is captured.

Table 6: GVP Application Identifiers and SIP Headers

Application ID	Description	SIP Header
Application DBID	The DBID assigned to the GVP component `Application` by the Configuration Layer. This ID is used internally by the Reporting Server to generate reports. You can view the ID in the Genesys Administrator on the `Deployment > Deploy > Deploy GVP > <GVP Solution Deployment Scenario> > 3. Summary` tab.	N/A
IVR Profile name	The user-defined name that was assigned to the IVR Profile when it was created. **Note:** Spaces are not allowed in the IVR Profile name. A GVP component can control the voice or call control application for a new call leg session, by extracting the value of the `gvp.rm.tenant-id` parameter from the `X-Genesys-GVP-Session-ID` header of an existing session, and using this value for the `gvp-tenant-id` parameter in a new SIP request.	• `gvp-tenant-id` parameter in the SIP `Request-URI`—The Resource Manager uses this parameter, if present, to identify the voice or call control application for a new session. For more information, see "Mapping the Call to an IVR Profile" on page 30. • `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.
IVR Profile DBID	The DBID assigned to the IVR Profile by the Configuration Layer.	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.

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) may change. 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

2 Provisioning GVP

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

This information appears in the following chapters:

- Chapter 3, “Configuration and Provisioning Overview,” on page 77
- Chapter 4, “Configuring Common Features,” on page 85
- Chapter 5, “Configuring the Resource Manager,” on page 123
- Chapter 6, “Provisioning IVR Profiles,” on page 139
- Chapter 7, “Configuring the Media Control Platform,” on page 157
- Chapter 8, “Configuring the Call Control Platform,” on page 183
- Chapter 9, “Configuring the Fetching Module and Squid Proxy,” on page 197
- Chapter 10, “Configuring the Reporting Server,” on page 203

Chapter

3 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 8.0 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 the Genesys Administrator. For information about creating and deploying the GVP `Applications`, see the *Genesys Voice Platform 8.0 Deployment Guide*.

To make the voice and call control applications available for GVP to use, you must provision the IVR Profiles in the Genesys Administrator. To trigger the launching of a particular VoiceXML or CCXML application when an incoming call is received, map the IVR Profile to a DNIS range. For more information about provisioning IVR Profiles and, if required, mapping them to DNs, see Chapter 6 on page 139.

Configuring GVP Processes in the Genesys Administrator

The following procedure describes how to configure GVP `Application` and `IVR Profile` objects in the Genesys Administrator. For more information about using the Genesys Administrator, see the online *Framework 8.0 Genesys Administrator Help.*

Procedure: Viewing or modifying GVP configuration parameters

Purpose: To describe the general method to use the 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 8.0 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 the Genesys Administrator. To access the Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`

Start of procedure

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

   Figure 4 and Figure 5 on page 80 show the `Settings` tab.

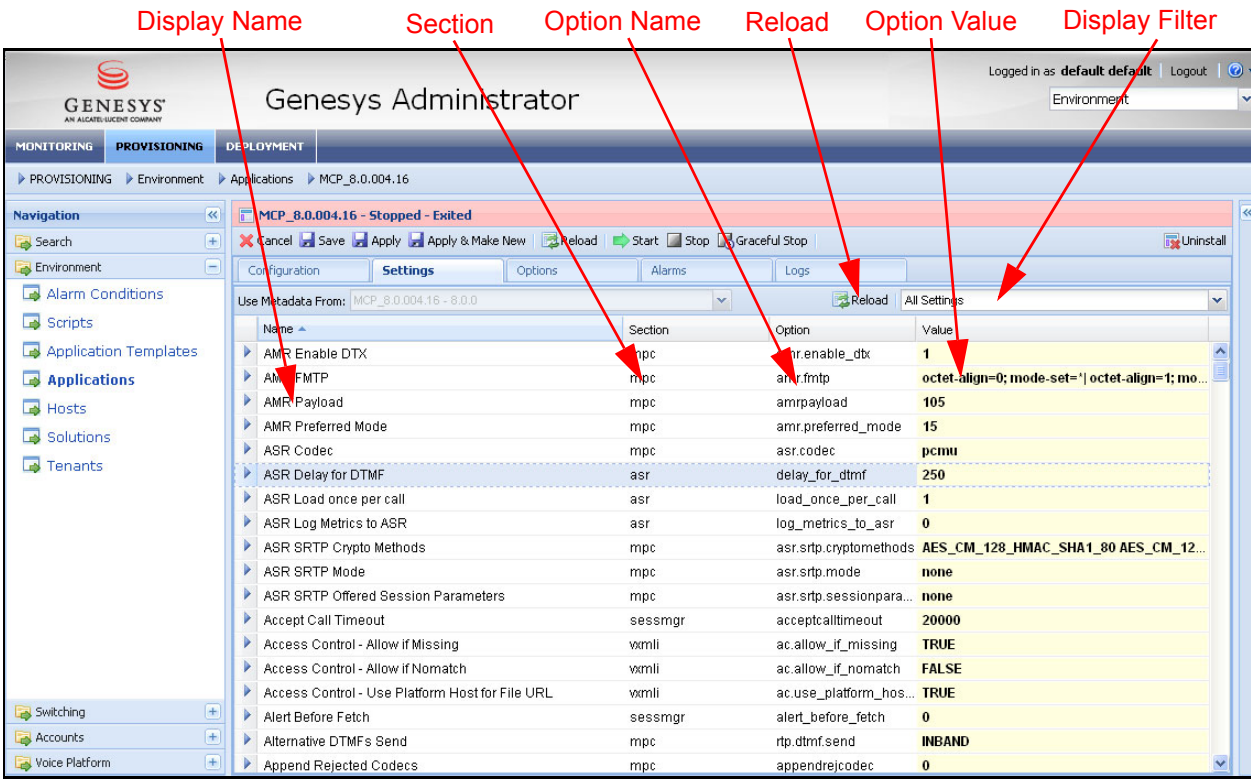


Figure 4: Settings Tab

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

- A plain-language display name.
- The configuration section that contains the option.
- The configuration option name, as it would appear on the `Options` tab.
- The current option value, either user-defined or default. User-defined values display in bold.

  User-defined values also appear on the `Options` tab.

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

- To filter the options that are displayed, select a different grouping from the `Display Filter` drop-down list.

  For example, the Media Control Platform `Application` provides the following option groupings:

  - Media Control Platform Main Settings
  - Logging
  - GVP Logging and Reporting
  - RemoteDial Application Module Settings
  - Conference Application Module Settings
  - Media Processing Settings
  - Session Manager Settings
  - SIP Settings
  - Speech Resource Management Settings
  - SNMP Settings
  - NGI Settings
  - All Settings
- To view more information about an option, click the arrow to the left of the display name. A description of the option appears (see Figure 5).

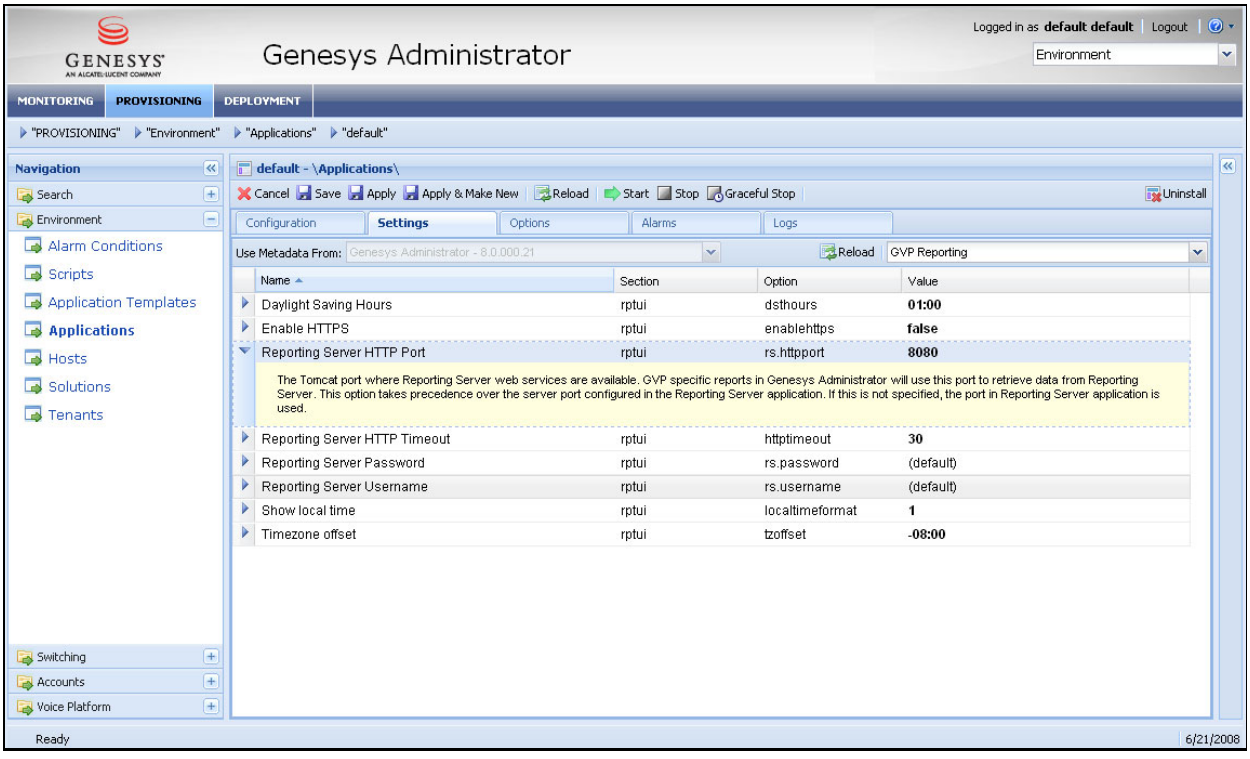


Figure 5: default Application Settings Tab, GVP Reporting Filter

3. To change an option setting:
   a. Click the `Name` or `Value` of the option that you want to change.

      A dialog box appears, which displays a full description of the option, its current value, valid values, default value (if applicable), and when changes takes effect.

      You can view the same information for all configuration options in the *Genesys Voice Platform 8.0 Configuration Options Reference.*
   b. Enter the new value in the `Value` field or, if applicable, select the option from the `Value` drop-down list. In cases where multiple values can be selected, hold down `Ctrl` to select multiple values.
   c. Click `OK`.
4. To save your changes, click `Save` and then `Apply`.

   Option values that you change on the `Settings` tab get updated on the `Options` tab as well.

   Similarly, options that you change on the `Options` tab get updated on the `Settings` tab. However, be aware that the `Settings` tab displays options and option names only as defined in the component metadata.
5. To update the metadata descriptions (from an updated Templates XML file), click `Reload`. This reloads the metadata file without affecting configured option values.

End of procedure

Task Summary: Configuring GVP

Table 7 provides an overview of the tasks to implement full GVP functionality in your deployment.

Table 7: 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 Communications and Routing" on page 86. • For secure communications, see "Enabling Secure Communications" on page 93. • For High Availability, see "Enabling High Availability" on page 134. See also component-specific requirements: • For the Media Control Platform, see Table 28 on page 158. • For the Call Control Platform, see Table 31 on page 184.
Provision the resources for the Resource Manager.	See "Configuring Logical Resource Groups" on page 126.
Provision the IVR Profiles.	See Chapter 6 on page 139.
Enable GVP Reporting.	• Configure the options in the `ems` configuration section of the Resource Manager, Media Control Platform, Call Control Platform, Fetching Module, and, if applicable, Cluster Manager `Application` objects. For more information, see "Configuring EMS Reporting" on page 103. • Configure the Reporting Server (see Chapter 10 on page 203). • If required, configure access control for Reporting services (see "Controlling Access to Reporting Services" on page 210). • On the `Monitoring` tab of the 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 display in the Genesys Administrator. For more information, see "Connecting to the Genesys Administrator" on page 213.
(Optional) Enable Automatic Speech Recognition (ASR) and Text-to-Speech (TTS).	See "Enabling ASR and TTS" on page 159.

Table 7: Configuring and Provisioning GVP (Continued)

Objective	Related Procedures and Actions
(Optional) Enable conferencing.	See "Enabling Conference Services" on page 102.
(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 "Configuring Logging" on page 111. • Session behavior and performance, see "Configuring Session Timers and Timeouts" on page 117. • Messaging, see "Customizing SIP Responses" on page 116 and Table 64 on page 279. • Call Control Platform device profiles, see "Configuring Device Profiles" on page 188. • Caching behavior, see "Configuring the Squid Caching Proxy" on page 200.

Chapter

4 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.

The configuration options tables provide parameter descriptions as well as 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 8.0 Configuration Options Reference*.

Configure the options in the Genesys Administrator on the `Provisioning > Environment > Applications > <GVP Component> > Settings` tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

Note: The configuration parameters for GVP processes are complex and provide a great deal of flexibility. Deploying GVP as described in the *Genesys Voice Platform 8.0 Deployment Guide* provides a fully functional, basic GVP deployment, with the minimum customizations required for GVP to operate in your environment. Ensure that you review the configuration options and fully understand their implications before you perform additional customizations.

Configuring SIP Communications and Routing

Table 8 summarizes the steps and parameters to configure the transport and routing mechanisms for SIP messaging within the GVP deployment.

Table 8: Task Summary: Configuring SIP Communications and Routing

Objective	Related Procedures and Actions
For each Resource Manager, Media Control Platform, and Call Control Platform `Application` in your deployment, configure the SIP transports for the supported transport protocols.	Configure the `sip.transport.<x>` options (see page 92). 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, UDP is the preferred protocol for all components (`sip.transport.0`). • To make TCP the preferred protocol, either rename (in other words, reorder) the respective `sip.transport` parameters, or else remove the default `sip.transport.0` UDP parameter so that the default `sip.transport.1` TCP parameter is the lowest defined `sip.transport.<x>`. • For secure SIP (SIPS) communications, specify a transport for TLS. For more information, see "Enabling Secure Communications" on page 93. **Note:** If you change the default preferred protocol for the Call Control Platform, there are additional steps you must take on the CCXML application side to ensure that the `Request-URI` specifies the correct protocol. For more information, see page 184.

Table 8: Task Summary: Configuring SIP Communications and Routing (Continued)

Objective	Related Procedures and Actions
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.secureroutesetset` option (see page 90 or 91). • For each Resource Manager, Media Control Platform, and Call Control Platform `Application` in your deployment, configure the required `sip.route.dest.<n>` entries (see page 89).
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, relevant options are in the `proxy` configuration section. • For the Media Control Platform and Call Control Platform, relevant options are in the `sip` configuration section.

Table 9 provides information about important SIP communications and routing options. Table 9 provides 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.

For information about all the available configuration options, see the *Genesys Voice Platform 8.0 Configuration Options Reference.*

Table 9: Selected SIP Communications and Routing Options

Section. Option Name	Description	Valid Values and Syntax
Resource Manager: • proxy.sip.route.default.<protocol> Media Control Platform and Call Control Platform: • sip.route.default.<protocol>	The default route for messages for the `<protocol>`, where `<protocol>` is: • `tcp`—The parameter defines the default route for Transmission Control Protocol (TCP) messages. • `tls`—The parameter defines the default route for Transport Layer Security (TLS) messages. • `udp`—The parameter defines the default route for User Datagram Protocol (UDP) messages. The value of this parameter is a number that corresponds to the transport interface index (`<x>`) for a transport defined by `sip.transport.<x>` (see page 92). The default transport is used when the component application finds no routes for the `<protocol>` in the routing table (see `sip.route.dest.<n>` on page 89). If the `sip.route.default.<protocol>` parameter is not set, the first UDP transport found in `sip.transport.<x>` becomes the default.	Any unsigned integer. **Default value:** Empty

Table 9: Selected SIP Communications and Routing Options (Continued)

Section. Option Name	Description	Valid Values and Syntax
Resource Manager: • proxy.sip.route.dest.<n> Media Control Platform and Call Control Platform: • sip.route.dest.<n>	Each <n> represents an entry in a routing table. The value of this parameter specifies the content of the entry in the routing table. The component application searches the routing table to identify the SIP transport and network interface to use for outbound calls. The order of the sip.route.dest.<n> parameters matters, because the component application searches the routing table linearly until it finds a <destination> entry that matches the requested, masked IP address. If the masked IP address matches a destination entry, that route is accepted. The <transport> part of the routing table entry then determines the transport to use. • In most cases, the first accepted route will be used. • If the protocol is specified or required (for example, TCP must be used when the message size is larger than sip.mtusize), the accepted route must also match the required protocol. If there is no routing table entry with a transport that matches the specified protocol, the default route of that protocol will be used. • If no route is found, the default route for the protocol in sip.transport.0 will be used. **Example:** sip.route.dest.0=138.120.72.0 255.255.255.0 1 0 Say a call is made to a host at 138.120.72.20. Applying the <netmask> of 255.255.255.0 to the outgoing IP address yields 138.120.72.0, which matches the defined <destination> in the route. Therefore, the transport in sip.transport.1 will be used. **Note:** In the Resource Manager configuration, there must be a proxy.sip.route.dest.<n> parameter defined for each <n> in the parameter proxy.sip.route.dests (see page 90).	A string representing <destination> <netmask> <transport> <metric> where: • <destination> is the subnet IP address of the network interface. • <netmask> is the network mask that the component application applies in evaluating whether the requested IP address matches the <destination> entry. • <transport> corresponds to the transport interface index (<x>) for the applicable transport, as defined by sip.transport <x> (see page 92). • <metric> is the routing metric. (Not used in GVP 8.0) **Default value:** Empty

Table 9: Selected SIP Communications and Routing Options (Continued)

Section. Option Name	Description	Valid Values and Syntax
(For Resource Manager only) proxy. sip.route.dests	An index list for entries in a routing table.	A space-delimited list of integers, starting from `0` and incrementing by `1` for each entry. **Example:** `0 1 2 3` specifies four entries in the routing table. **Default value:** Empty
(For Media Control Platform and Call Control Platform only) sip. routeset	A comma-separated list of SIP Proxy addresses that defines a route set for non-secure SIP outbound calls. If defined, this route set is inserted as the `ROUTE` header for all outgoing calls. This forces GVP to use the defined route set for SIP messages. Using the `lr` parameter with the URI (see syntax) forces the User Agent Client (UAC) to place the remote target URI into the `Request-URI` and to include the route set in the `ROUTE` header. The SIP port on SIP Proxies is usually `5060`. **Example:** `<sip:RM_host.yourdomain.com:5060;lr>,<sip:Proxy2.yourdomain.com:5060;lr>`—The Media Control Platform or Call Control Platform will send the outgoing request to the Resource Manager, which will, in turn, route the request to Proxy 2, which will redirect the message to its intended destination. **Note:** The route set does not apply to SIP `REGISTER` messages.	`<sip:<Resource Manager IP address>:<Resource Manager SIP port>;lr>[,<sip:<Next SIP Proxy or UA IP address>:<Proxy SIP port>;[lr]>,...]` **Note:** The outer angle brackets are required characters in the string. **Default value:** Empty

Table 9: Selected SIP Communications and Routing Options (Continued)

Section. Option Name	Description	Valid Values and Syntax
(For Media Control Platform and Call Control Platform only) sip.secureroutes et	A comma-separated list of SIP Proxy addresses that defines a route set for secure SIP outbound calls. If defined, this route set is inserted as the `ROUTE` header for all outgoing calls. This forces GVP to use the defined route set for secure SIP messages. Using the `lr` parameter with the URI (see syntax) forces the UAC to place the remote target URI into the `Request-URI` and to include the route set in the `ROUTE` header. The secure SIP port on SIP Proxies is usually `5061`. **Note:** The route set does not apply to SIP `REGISTER` messages.	• `<sips:<Resource Manager IP address>:<Resource Manager secure SIP port>;lr>[,<sips:<Next SIP Proxy or UA IP address>:<Proxy SIP port>;[lr]>,...]` • `<sip:<Resource Manager IP address>:<Resource Manager secure SIP port>;transport=tls;lr>[,<sip:<Next SIP Proxy or UA IP address>:<Proxy SIP port>;transport=tls;[lr]>,...]` **Note:** The outer angle brackets are required characters in the string. **Default value:** Empty

Table 9: Selected SIP Communications and Routing Options (Continued)

Section. Option Name	Description	Valid Values and Syntax
Resource Manager: • proxy. sip.transport.<x> • registrar. sip.transport.x • monitor. sip.transport.x Media Control Platform and Call Control Platform: • sip. transport.<x>	The parameters that define the transport layer for SIP stack and the network interfaces that are used to process SIP requests. `<x>` is the transport interface index that identifies the transport, so that you can specify different combinations of parameters for different protocols. The Resource Manager specifies separate SIP transports for messages relating to its role as SIP Proxy (`INVITE` messages), SIP Registrar (`REGISTER` messages), and monitor (`OPTIONS` messages). The default transport is the smallest non-empty transport interface index (`transport.<x>`). For Secure SIP, ensure that you specify Transport Layer Security (TLS) as the transport layer protocol (`<transport_type>` = `tls`), and include the following additional SIP transport parameters: • `cert=<TLS certificate path and file name>` (required) • `key=<TLS key path and file name>` (required) • `type=<type of secure transport>` (optional) Valid values: `TLSv1|SSLv2|SSLv3|SSLv23` Default value: `SSLv23` • `password=<password associated with the certificate and key pair>` (required only if the key file is password protected) **Note:** For each transport that you define with a specific IP address (in other words, the value of `<ip>` is not `any`), you must also configure at least one entry in the routing table (see `sip.route.dest.<n>` on page 89), or the interface will never be used.	`<transport_name> <transport_type>:<ip>: <port> [<parameters>]` where: • `<transport_name>` is any alphanumeric string. • `<transport_type>` is the transport layer protocol: `udp|tcp|tls`. • `<ip>` is the IP address of the network interface that accepts incoming SIP messages (the default value of `any` means all network interfaces). • `<port>` is the port number where SIP stack accepts incoming SIP messages. • `[<parameters>]` are any additional, optional SIP transport parameters. **Default values:** See Table 10 on page 93.

Table 10 summarizes the default values for the `sip.transport.<n>` parameter for the Resource Manager, Media Control Platform, and Call Control Platform.

Table 10: 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`
	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`
Media Control Platform **Note:** If all `sip.transport.x` values are empty, UDP, TCP, and TLS transports will all be enabled and respectively listen from ports 5060, 5060, and 5061 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`
Call Control Platform	sip.transport.0	`transport0 udp:any:5068`

Enabling Secure Communications

Table 11 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) communications.

For general information about secure communications in GVP, see "Secure Communications" on page 70.

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

Table 11: 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 Creating an SSL private key and certificate). 2. On the Resource Manager, Media Control Platform, and Call Control Platform `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 86 and 92, and the default values in Table 10 on page 93). 3. On the Media Control Platform and Call Control Platform `Applications`, specify secure routing for outbound calls (see information about the `sip.secureroutes et` option on page 91 and the `sip.route.dest.<n>` option on page 89). 4. Modify the CCXML applications, as required, to ensure that the `Request-URI` specifies TLS as the transport protocol. For more information, see page 184.
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 Creating an SSL key and self-signed certificate for use with IIS, page 97. 2. On each Fetching Module `Application` in your deployment, configure the Fetching Module process (pwproxy) to access files over HTTPS (see Configuring the Fetching Module for HTTPS, page 99). Configure additional `iproxy.ssl_*` options as required for your deployment. 3. Modify the Squid configuration file, if necessary, to configure "safe" and SSL ports, and to enforce SSL (see Modifying the Squid Configuration, page 200).

Table 11: Task Summary: Enabling SIPS, HTTPS, and SRTP in GVP (Continued)

Objective	Related Procedures and Actions
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.

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` or `offer`) 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`

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`

If necessary, set up the Reporting Server web server to use HTTPS.	• If the Reporting Server Tomcat is deployed primarily as a Servlet/JSP container behind another web server (such as Apache or Microsoft IIS), no further configuration is required. • If the Reporting Server Tomcat is deployed as a stand-alone web server, you must generate and deploy the SSL private key and certificate, and then modify the Tomcat configuration to use native SSL. For more information, see Configuring a stand-alone Tomcat web server for SSL, page 101. See also “Enabling HTTP Basic Authorization for Reporting” on page 211.
Configure Genesys Administrator to use HTTPS to access Reporting Server web services, for the GVP reports that display in the `Monitoring > Voice Platform` view.	• In the Genesys Administrator, on the `Provisioning > Environment > Applications > default > Settings` 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.

The following procedures support the tasks outlined in Table 11:

- Creating an SSL private key and certificate
- Creating an SSL key and self-signed certificate for use with IIS, page 97
- Configuring the Fetching Module for HTTPS, page 99

Procedure: Creating an SSL private key and certificate

Purpose: To provide an example of a way 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 provide an example of a way 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

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

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 Creating an SSL key and self-signed certificate for use with IIS, page 97.

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 98).
   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 98).
   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 6.

```
-----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/O0Ot3+cKhW1Ef1osh5fFxlWlhNrwv11mpkG0OZ8pVME=
-----END CERTIFICATE-----
```

Figure 6: Sample PEM Certificate File

2. Save the file.
3. In the Genesys Administrator, on the `Provisioning > Environment > Applications > <Fetching Module> > Settings` tab, modify the Fetching Module configuration:
   a. Verify that the value of the `iproxy.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 8.0 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 Modifying the Squid Configuration, page 200.

Procedure: Configuring a stand-alone Tomcat web server for SSL

Purpose: To provide an example of a way to configure SSL support on Tomcat, when Tomcat is deployed as a stand-alone web server for Reporting Server.

Start of procedure

1. Generate the private key and certificate for Tomcat.
   - To generate the certificate with `openssl`, execute the following command:

     ```
     openssl pkcs12 -export -infile mycert.crt -inkey mykey.key \
               -outfile mycert.p12 -name tomcat -CAfile myCA.crt \
               -caname root -chain
     ```

     For information about obtaining the OpenSSL Toolkit (openssl), see the Prerequisites item on page 97.
   - To generate a self-signed certificate with the Java `keytool`, execute the following command:

     ```
     $JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
     ```

2. Configure Tomcat to use native SSL.

   For example, add the following element in the `$CATALINA_HOME/conf/server.xml` file:

   ```
   <-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
   <!--
   <Connector className="org.apache.coyote.tomcat5.CoyoteConnector"
              port="8443" minProcessors="5" maxProcessors="75"
              enableLookups="true" disableUploadTimeout="true"
   ```

```
                acceptCount="100" debug="0" scheme="https" secure="true";
                clientAuth="false" sslProtocol="TLS"/>
    -->
```

End of procedure

Enabling Conference Services

Table 12 summarizes the steps and parameters to configure the GVP deployment to provide conference service.

Note: 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 (see, for example, "Selecting a Resource for Conference Services" on page 34).

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 other chapters in the Provisioning section of this guide, and see the *Genesys Voice Platform 8.0 Configuration Options Reference.*

Table 12: Task Summary: Configuring Conferencing

Objective	Related Procedures and Actions
Assign a conference resource to a logical resource group that provides conference service.	See "Configuring Logical Resource Groups" on page 126: 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` (see page 133). 2. Set the general conference maximums for the resource group (see the `confmaxsize` and `confmaxcount` options). 3. If the resource (for example, Media Control Platform or Call Control Platform) has not already been added to the Resource Manager connections, add it. For more information, see the *Genesys Voice Platform 8.0 Deployment Guide* chapter about post-installation activities.

Table 12: Task Summary: Configuring Conferencing (Continued)

Objective	Related Procedures and Actions
Create an IVR Profile for conference service.	Set the following required parameter: • `gvp.service-prerequisite.conference-id` (see page 151) 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 141.
If the deployment uses an external conferencing or bridge server, specify the external server IP address on the Media Control Platform.	See the Media Control Platform `sip.confserver` option.
Verify that conference-related settings on the Media Control Platform and Call Control Platform are suitable.	• For the Media Control Platform, review options in the `conference` section. • For the Call Control Platform, verify settings for the `Default Conference` device profile, and in the `mediacontroller` and `mediactrller` configuration sections.
(Optional) Customize the SIP response codes and Resource Manager behavior on error.	On the Resource Manager, customize the value of the `rm.conference-sip-error-respcode` option. For more information, see Table 64 on page 279.

Configuring EMS Reporting

Table 13 describes important parameters for EMS Reporting, which you configure in the `ems` section of the Resource Manager, Media Control Platform, Call Control Platform, Fetching Module, and, if applicable, Cluster Manager `Application` objects.

For general information about EMS Reporting in a GVP deployment, see "Logging and Reporting" on page 60.

For information about additional EMS configuration options, see the *Genesys Voice Platform 8.0 Configuration Options Reference.*

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

Note: The Resource Manager and Fetching Module do not support upstream metrics, which are application-level logs (relating to VoiceXML or CCXML applications) that are configured to be delivered to the Reporting Server through the Data Collection Sink (`DATAC`). Configuration parameters in the `ems` section that relate to `DATAC` do not apply to the Resource Manager or Fetching Module.

Table 13: Selected EMS Reporting Configuration Options—ems Section

Option Name	Description	Valid Values and Syntax
(For Media Control Platform and Call Control Platform only) dc.default.logfilter	The filter that determines which logs will be delivered upstream to the Reporting Server for Call Events reporting. **Example:** On the Media Control Platform, a value of `0,1|176-178|*|4|*|*` means that all `CRITICAL` (`0`) and `ERROR` (`1`) messages for the MCP code modules that have Module IDs in the range `176-178` will be sent to the sink, and all `INFO` (`4`) messages for all code modules will be sent as well.	`<Levels1>|<ModuleIDs1>|<SpecifierIDs1>[|<Levels2>|<ModuleIDs2>|<SpecifierIDs2>...]` where: • `<Levels>` is the GVP log level or a range of log levels. • `<ModuleIDs>` identifies the code modules within the tracing component. • `<SpecifierIDs>` identifies the specific log messages. • The pipes (`|`) are a required part of the syntax, and values within the pipes can be formatted as a range (`m-n`) or as individual, comma-separated numbers. For the valid Module and Specifier IDs, see Appendix A, "Module and Specifier IDs" on page 255. The wildcard character (`*`) means all. **Default values:** `0-2|*|*`

Table 13: Selected EMS Reporting Configuration Options—ems Section (Continued)

Option Name	Description	Valid Values and Syntax
(For Media Control Platform and Call Control Platform only) dc.default.metricsfilter	The filter that determines which metrics in the Data Collection Sink (DATAC) will be forwarded to the Reporting Server. If the gvp.log.metricsfilter parameter (see page 143) has been set in an IVR Profile, the IVR Profile value will override this default setting, for sessions that execute under this IVR Profile.	<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 8.0 Metrics Reference.* The wildcard character (*) means all. **Default values:** • Media Control Platform—0-15, 41, 52-55, 74, 136-141 • Call Control Platform—1001, 1009, 1012-1013, 1031, 1050, 1052, 1058-1059
log_sinks	A pipe (|)-separated list of the log sinks that will be loaded when the EMS Logging Service initializes. For more information about the log sinks, see "Log Sinks" on page 65.	[MFSINK]|[DATAC]|[TRAPSINK] **Default values:** • Resource Manager and Fetching Module—MFSINK|TRAPSINK • Media Control Platform and Call Control Platform—DATAC|MFSINK|TRAPSINK • Cluster Manager—MFSINK

Table 13: Selected EMS Reporting Configuration Options—ems Section (Continued)

Option Name	Description	Valid Values and Syntax
logconfig.<Sink Name>	The filter that determines which log messages are sent to the specified sink, where `<Sink Name>` is one of the following: • `MFSINK`—the MF Adaptation Sink • `DATAC`—the Data Collection Sink The filter is a pipe (|)-separated list of the following sets of parameters: • Log level: 0 = Critical 1 = Error 2 = Warning 3 = Note 4 = Info 5 = Debug • Code component that generates the log event. • The specific log message. This configuration option enables GVP to improve EMS Logging performance by prefiltering the log messages. However, Genesys recommends that you retain the default filter setting (all log messages), to ensure that you do not inadvertently omit capturing a type of log message that you later require for tracing purposes. **Note:** Management Framework can only log to file what is passed to it in accordance with the `logconfig.MFSINK` filter setting. **Changes take effect:** Immediately.	`<Levels1>|<ModuleIDs1>|<SpecifierIDs1>[|<Levels2>|<ModuleIDs2>|<SpecifierIDs2>...]` where: • `<Levels>` is the GVP log level or a range of log levels. • `<ModuleIDs>` identifies the code modules within the tracing component. • `<SpecifierIDs>` identifies the specific log messages. • The pipes (`|`) are a required part of the syntax, and values within the pipes can be formatted as a range (`m-n`) or as individual, comma-separated numbers. For the valid Module and Specifier IDs, see Appendix A, "Module and Specifier IDs" on page 255. The wildcard character (`*`) means all. **Default values:** See Table 14 on page 110.

Table 13: Selected EMS Reporting Configuration Options—ems Section (Continued)

Option Name	Description	Valid Values and Syntax
metricsconfig. <Sink Name>	The filter that determines which metrics are sent to the specified sink, where `<Sink Name>` is one of the following: • `MFSINK`—the MF Adaptation Sink • `DATAC`—the Data Collection Sink • `TRAPSINK`—the SNMP trap sink The filter is a comma-separated list of the specific metrics that will be queued and processed. This configuration option enables GVP to improve EMS Reporting performance by prefiltering the metrics. However, Genesys recommends that you do not change the default filter for any sink except `MFSINK`. For example, you may want to change the settings for the `ems.metricsconfig.MFSINK` parameter, in order to make a specific set of metrics available to Management Framework, so that these metrics can be sent to Message Server or written to file. **Note:** Reducing the scope of the `metricsconfig.TRAPSINK` filter on the Media Control Platform will impact the calculation of VAR summary statistics. **Changes take effect:** Immediately.	`<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 8.0 Metrics Reference.* The wildcard character (`*`) means all. **Example:** `5-8,50-53,70,71` means that metrics with IDs `5, 6, 7, 8, 50, 51, 52, 53, 70,` and `71` will be sent to the sink. **Default values:** See Table 14 on page 110.
(For Media Control Platform and Resource Manager only) ors.reportinginterval	The interval, in seconds, at which accumulated operational reports are submitted to the Reporting Server.	An integer in the range of 1–299. **Default value:** `60` (seconds)
(For Media Control Platform, Call Control Platform, and Resource Manager) rc.batch_size	The number of upstream messages that the Reporting Client queues before sending them on to the Reporting Server. A higher batch size reduces bandwidth constraints, but at the cost of sending data at larger intervals. See also `rc.cdr.batch_size` on page 108.	An integer in the range of 1–5000. **Default value:** `500`

Table 13: Selected EMS Reporting Configuration Options—ems Section (Continued)

Option Name	Description	Valid Values and Syntax
(For Media Control Platform, Call Control Platform, and Resource Manager) rc.cdr.batch_size	The number of call detail records (CDRs) to send in a message to the Reporting Server. A higher batch size lessens the demands on network bandwidth and also significantly reduces the transaction overhead on the Reporting Server database, but at the cost of a lag of up to 10 seconds in reporting updates. Batches time out after 10 seconds (in other words, the message is sent after 10 seconds even if the batch is not filled). A batch size of 1 means that CDRs are delivered in near real-time. As a rule of thumb, set the batch size based on the expected call throughput. For example, if a reporting delay of up to 5 seconds is acceptable, set the batch size to the number of CDR updates you expect to receive every 5 seconds, assuming 2 updates per call. Be aware, however, that there could still be a 10-second delay if actual call volumes are less than expected. For high call densities, Genesys recommends retaining the default value of `500`.	An integer in the range of 1–5000. **Default value:** `500`
(For Media Control Platform, Call Control Platform, and Resource Manager) rc.cdr.local_queue_path	The full path to the local SQLite3 database file that serves as the persistent queue for CDRs being submitted to the Reporting Server. Even if GVP component applications share a host, each component must have its own local queue file.	`<Path to file>` **Default value:** `$InstallationRoot$\config\cdrQueue.db`
(For Media Control Platform, Call Control Platform, and Resource Manager) rc.cdr.msg_broker_uri	The URI of the ActiveMQ message broker that implements Java Message Service (JMS) on the Reporting Server, for CDR reporting.	`tcp://<Reporting Server host name or IP address>:<port>` **Default value:** `tcp://$ReportingServer Hostname$:61616`

Table 13: Selected EMS Reporting Configuration Options—ems Section (Continued)

Option Name	Description	Valid Values and Syntax
(For Media Control Platform, Call Control Platform, and Resource Manager) rc.local_queue_path	The full path to the local database file that serves as the persistent queue for upstream logging to the GVP Reporting Client. Even if GVP components (for example, Resource Manager or Media Control Platform) share a host, each component has its own local queue file.	<Path to file> **Default value:** • For Resource Manager and Media Control Platform—$InstallationRoot$\config\reportingClientQueue.db • For Call Control Platform—$InstallationRoot$\config\upstreamQueue_CCP.db
(For Media Control Platform, Call Control Platform, and Resource Manager) rc.msg_broker_uri	The URI of the JMS Message Broker on the Reporting Server.	tcp://<Reporting Server host name or IP address>:<port> **Default value:** tcp://$ReportingServerHostname$:61616
(For Media Control Platform and Resource Manager only) rc.ors_local_queue_path	The full path to the local SQLite3 database file that serves as the persistent queue for OR statistics being submitted to the Reporting Server. Even if GVP components (for example, Resource Manager or Media Control Platform) share a host, each component has its own local queue file.	<Path to file> **Default value:** $InstallationRoot$\config\cdrQueue_rm.db
(For Media Control Platform and Resource Manager only) rc.ors.msg_broker_uri	The URI of the ActiveMQ message broker that implements Java Message Service (JMS) on the Reporting Server, for delivering OR statistics.	tcp://<Reporting Server host name or IP address>:<port> **Default value:** tcp://$ReportingServerHostname$:61616

Table 13: Selected EMS Reporting Configuration Options—ems Section (Continued)

Option Name	Description	Valid Values and Syntax
trace_flag	Flag that indicates whether debug-level logging is enabled. When enabled, debug-level logs will be processed and filtered like logs at other levels. Debug-level logging places a load on the logging infrastructure that degrades performance. Therefore, do not enable it in a production environment unless Genesys Support directs you to do so. **Changes take effect:** Immediately.	• `True` • `False` **Default value:** `False`

Table 14 summarizes the default values for the various logs and metrics filters for the Resource Manager (RM), Media Control Platform (MCP), Call Control Platform (CCP), Fetching Module (FM), and Cluster Manager (CM) `Application` objects.

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

Table 14: Default Log and Metrics Filters

Com-ponent	Option Name (in ems section)*				
	logconfig.DATAC	logconfig.MFSINK	metricsconfig.DATAC	metricsconfig.MFSINK	metricsconfig.TRAPSINK
RM	N/A	`0-3,5|*|*`	N/A	`0-15,19-41,43,52-56,72-74,76-81,127,129,130,132-141`	N/A
MCP	`0-2|*|*`	`0-3,5|*|*`	*	`0-15,19-41,43,52-56,72-74,76-81,127,129,130,132-141`	N/A
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`	*
* For descriptions of the configuration options, see Table 13 on page 104.					

Table 14: Default Log and Metrics Filters (Continued)

Com-ponent	Option Name (in ems section)*				
	logconfig.DATAC	logconfig.MFSINK	metricsconfig.DATAC	metricsconfig.MFSINK	metricsconfig.TRAPSINK
FM	N/A	0-3,5|*|*	N/A	0-15,19-41,43,52-56,72-74,76-81,127,129,130,132-141	N/A
CM	N/A	*|*|*	N/A	*	N/A
* For descriptions of the configuration options, see Table 13 on page 104.					

Configuring Logging

Table 15 describes the parameters for logging that are most commonly customized. Table 16 on page 115 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 the Genesys Administrator on the `Provisioning > Environment > Applications > <GVP Application> > Settings` tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

Changes take effect immediately.

The `Application Templates` do not expose all the logging parameters that are standard in Genesys applications. The `Settings` tab and its metadata (which are also available in the *Genesys Voice Platform 8.0 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 8.0 Configuration Options Reference Manual.*

Table 15: Selected Configuration Options—log Section

Parameter Name	Description	Valid Values and Syntax
all	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 are not stored in the Log Database.	• `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.
standard	A comma-separated list of the output destinations to which the application (GVP process) sends log events of the Standard level.	
interaction	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).	
trace	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).	
debug	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).	

Table 15: Selected Configuration Options—log Section (Continued)

Parameter 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. (Not applicable for Reporting Server.)

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 15: Selected Configuration Options—log Section (Continued)

Parameter Name	Description	Valid Values and Syntax
segment	(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. (Not applicable for Reporting Server.)
time-format	The format in which the log file presents the time when the application generated the log record.	• `time`—The time string is formatted according to 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 formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds. **Example:** `2001-07-24T04:58:10.123`

Table 15: Selected Configuration Options—log Section (Continued)

Parameter Name	Description	Valid Values and Syntax
verbose	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 16 provides the default values for options in the `log` configuration section that are commonly modified.

Table 16: Default Values for Selected log Options

Option Name	Default Value					
	Resource Manager	Media Control Platform	Call Control Platform	Fetching Module	Reporting Server	Cluster Manager
all	Empty	`..\logs\MCP`	`..\logs\ccp`	`$InstallationRoot$\logs\fm,stdout`	Empty	Empty
debug	`..\logs\ResourceMgr`	`..\logs\MCP`	`..\logs\ccp`	Empty	Empty	`..\logs\ClusterMgr`
expire	`20 (files)`	`10 (files)`	`20 (files)`	`10 (files)`	`false`	`20 (files)`
interaction	`..\logs\ResourceMgr,network`	`..\logs\MCP`	`..\logs\ccp,network`	`..\logs\mcp_metricsfile,network`	Empty	`..\logs\ClusterMgr,network`
message_format	`short`	`short`	[Not exposed in template]	`short`	`short`	`short`
segment	`10000 KB`	`10000 KB`	`10000 KB`	`10000 KB`	`false`	`10000 KB`
standard	`..\logs\ResourceMgr,network`	`..\logs\MCP`	`..\logs\ccp,network`	`..\logs\MCP,network`	`network`	`..\logs\ClusterMgr,network`

Table 16: Default Values for Selected log Options (Continued)

Option Name	Default Value					
	Resource Manager	Media Control Platform	Call Control Platform	Fetching Module	Reporting Server	Cluster Manager
time-format	time	time	time	time	time	time
trace	..\logs\ResourceMgr	..\logs\MCP	..\logs\ccp	Empty	Empty	..\logs\ClusterMgr
verbose	standard	interaction	standard	interaction	trace	standard

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, and Call Control Platform signal for certain events and conditions.

For more information about the SIP response codes that are generated, and how the following configurable options relate to them, see Table 64 on page 279. For more information about all the configuration options, see the *Genesys Voice Platform 8.0 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.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.outbound-call-forbidden-respcode
  - gvp.policy.outbound-call-forbidden-set-alarm
  - gvp.policy.outbound-usage-limit-exceeded-set-alarm
  - 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
  - To further customize the SIP response code 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 120).

Resource Manager Session Timers

Table 17 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, except under the conditions described in "Session-Expires Header" on page 29.

For more information about how the Resource Manager uses expiry timeouts, see "Session Management" on page 27.

Table 17: Session Timer Configuration Options

Section. Parameter Name	Description	Valid Values and Syntax
IVR Profile		
gvp.general. sip.sessiontimer	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 that you configure for this `sip.sessiontimer` parameter overrides session expiry timeouts that are set at the level of the tenant, the resource, and the Resource Manager.	Any positive integer. **Default value:** Empty
Environment Tenant		
gvp.general. sip.sessiontimer	The timeout value, in seconds, for the SIP session that executes for the 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 the IVR Profile for this tenant, the value that you configure for this `sip.sessiontimer` parameter overrides session expiry timeouts that are set at the level of the resource and the Resource Manager.	Any positive integer. **Default value:** Empty

Table 17: Session Timer Configuration Options (Continued)

Section. Parameter Name	Description	Valid Values and Syntax
Media Control Platform/Call Control Platform		
sip. min_se	The minimum value of the `Session-Expires` header, in seconds, that the SIP stack will accept from a UAC (User Agent Client).	An integer in the range of 90–3600. **Default value:** `90`
sip. sessionexpires	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` parameter.	An integer in the range of 90–3600. **Default value:** `1800`
Resource Manager		
proxy. sip.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`
proxy. sip.sessionexpires	The timeout value, 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 17: Session Timer Configuration Options (Continued)

Section. Parameter Name	Description	Valid Values and Syntax
registrar. sip.registrar.maxexpiry 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 the value that will be returned.	An integer in the range of 60–7200. **Default value:** `60`
registrar. sip.registrar.minexpiry time	The minimum expiry time, in seconds, of this registrar. If the client requests an expiry time smaller than this value, the request will be 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:
  - `sip.timer.ci_proceeding`—The timeout for a client transaction that is in progress. If a final response is not received within the timeout period, the client transaction is considered terminated. (Default is 120000 ms, or 120 seconds.)
- Media Control Platform:
  - `mpc.rtp.timeout`—The timeout for the RTP stream. (Default is 60000 ms.)
  - `sessmgr.acceptcalltimeout`—The timeout for the platform to accept inbound calls, after alerting is issued. (Default is 20000 ms.)
  - `sessmgr.maxincalltime`—The maximum call time for inbound calls. (Default is 0—disabled.)
  - `sip.hfdisctimer`—The timeout to terminate a SIP hookflash transfer. (Default is 5000 ms.)
  - `stack.connection.timeout`—The connection timeout for the MRCP Client stack to establish a TCP connection to the MRCP server. (Default is 10000 ms.)
  - `stack.client.timeout`—The connection timeout for the MRCP Client to receive a response from the MRCP server. (Default is 10000 ms.)
  - `vxmli.default.connecttimeout`—The default value of the `connecttimeout` attribute for bridge or consultation transfers, if not provided. (Default is 30000 ms.)
  - `vxmli.initial_request_fetchtimeout`—The fetch timeout for the initial VoiceXML page. (Default is 30000 ms.)

    - `vxmli.max_script_time`—The maximum time allowed for each script or ECMAScript expression to execute. (Default is `2000 ms`.)
- Call Control Platform:
    - `ccxmli.fetch.timeout`—The default timeout for the fetch of the initial page to complete. (Default is 30 seconds.)

Chapter

5 Configuring the Resource Manager

This chapter describes the Resource Manager configuration requirements in your Genesys Voice Platform (GVP) deployment.

It contains the following sections:

Task Summary: Configuring the Resource Manager

Table 18 summarizes the configuration steps and options to implement Resource Manager functionality in your GVP deployment.

Table 18: Configuring the Resource Manager

Objective	Related Procedures and Actions
Set up the Resource Manager to function as SIP Proxy, SIP Registrar, and resource monitor and manager.	See "Configuring SIP Communications and Routing" on page 86. 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.
(Optional) Enable High Availability.	• Configuring Resource Manager High Availability, page 134. • Setting up the cluster mode execution environment, page 136.
Provision the GVP resources.	See "Configuring Logical Resource Groups" on page 126.
Provision the IVR Profiles.	See Chapter 6 on page 139.
Configure conferencing.	See "Enabling Conference Services" on page 102.
Configure EMS Reporting.	See "Configuring EMS Reporting" on page 103.
Customize logging.	See "Configuring Logging" on page 111.
Customize session management behavior and performance.	See "Configuring Session Timers and Timeouts" on page 117. See also parameters in the `proxy` section that specify parameters such as numbers of threads and connections.
Customize Resource Manager messaging.	See "Customizing SIP Responses" on page 116 and Table 64 on page 279.

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 the Genesys Administrator on the `Provisioning > Environment > Applications > <Resource Manager> > Settings` tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

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:

- `ems` (see Table 13 on page 104)—Parameters determine EMS Reporting behavior.
- `log` (see "Configuring Logging" on page 111)—Parameters determine behavior for Management Framework logging.
- `rm`—Parameters determine the behavior of the Resource Manager in its role as manager of GVP services.
- `proxy`—Parameters determine the behavior of the Resource Manager in its role as SIP Proxy and session manager.
- `registrar`—Parameters determine the behavior of the Resource Manager in its role as SIP Registrar.
- `monitor`—Parameters support the Resource Manager in its role as manager of GVP resources.
- `cmserviceagent`—The only parameter (`cmport`) supports High Availability, by configuring the port for communicating with the Cluster Manager.

Table 19 provides information about important Resource Manager parameters that are not described in Chapter 4 on page 85. Table 19 provides parameter descriptions as well as 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 8.0 Configuration Options Reference.*

Table 19: Selected Resource Manager Configuration Options

Option Name	Description	Valid Values and Syntax
`rm` Section		
cluster_ip	(Required for warm active-standby mode, for High Availability) The virtual IP address that the external load balancer will use to provide transport-level redundancy. If no value is set, the Resource Manager is not running in any form of active-standby mode. For more information, see Configuring Resource Manager High Availability, page 134.	`<IP address>` **Default value:** Empty

Table 19: Selected Resource Manager Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
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 defaults to the next behavior to select the IVR Profile (see "Mapping the Call to an IVR Profile" on page 30). **Changes take effect:** Immediately.	• `To` • `Request-Uri` • `History-Info` **Default value:** `History-Info`
`monitor` Section		
sip.proxy.optionsinterval	The interval, in milliseconds, at which the Resource Manager sends `OPTIONS` messages to a healthy resource to determine if the resource is alive.	Any unsigned integer. **Default value:** `5000`
sip.proxy.unavailoptionsinterval	The interval, in milliseconds, at which the Resource Manager sends `OPTIONS` messages to a dead resource to determine if the resource is alive.	Any unsigned integer. **Default value:** `5000`

Configuring Logical Resource Groups

For each type of service that GVP provides (VoiceXML, CCXML, Conference, or Gateway), you must create and configure a logical resource group that 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, the group has a single member).

Use the Genesys Administrator `Provisioning > Voice Platform > Resource Management > <Resource Manager> > Manage RM Resources` wizard to create, modify, or view settings for the resource group and to specify the resources

that belong to each group. For details about using the `Manage RM Resources` wizard, see Configuring logical resource groups.

You identify the actual resource hosts and applications in the connections that you configure for the Resource Manager. For more information, see the *Genesys Voice Platform 8.0 Deployment Guide* chapter about post-installation activities.

The following procedure provides the detailed steps to use the `Manage RM Resources` 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.

Except where indicated otherwise, all changes to logical resource group configurations take effect immediately.

Prerequisites

- The GVP `Application` objects have been installed, as described in the *Genesys Voice Platform 8.0 Deployment Guide*.
- You are logged in to the Genesys Administrator. To access the Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. In the Genesys Administrator, go to the `Provisioning > Voice Platform > Resource Management` panel (see Figure 7).

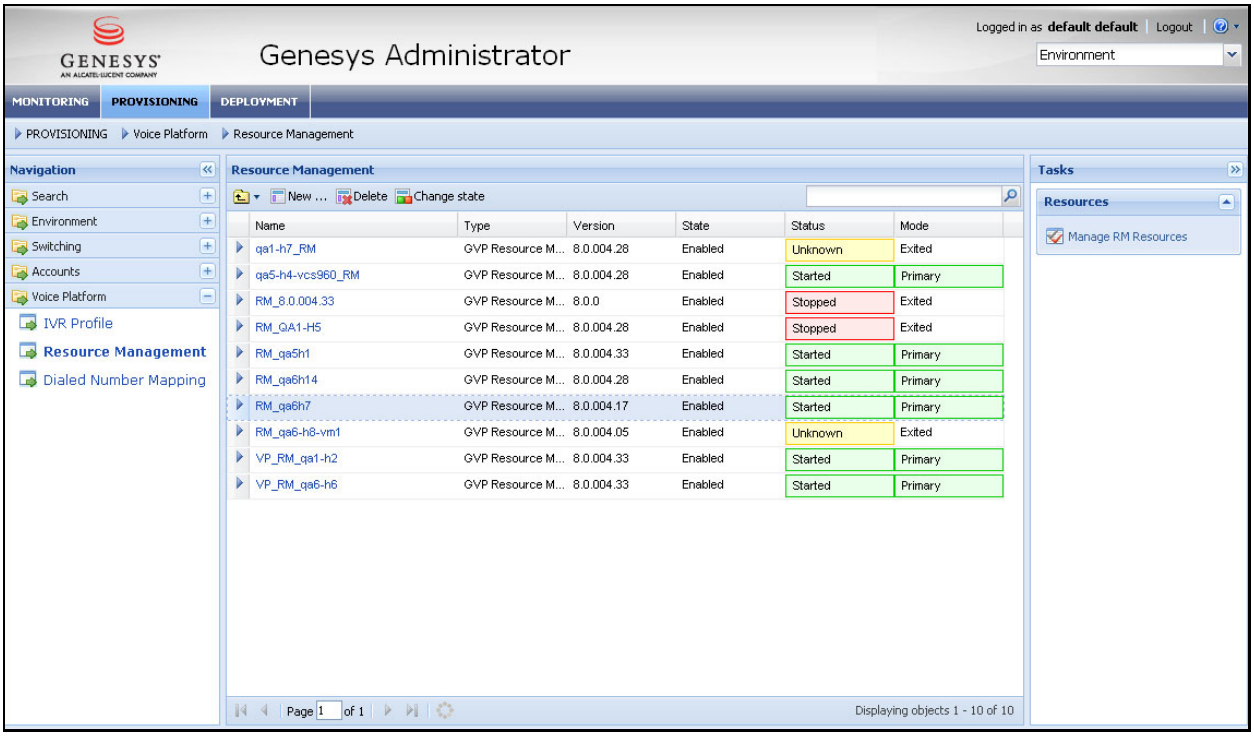


Figure 7: The Resource Management Panel

2. Select the Resource Manager for which you want to configure the logical resource group.

   Tip: Do not click the name of the Resource Manager, or you will link to the `Provisioning > Environment > Applications > <Resource Manager> > Configuration` tab. Rather, click anywhere else in the applicable row.

3. In the `Tasks` panel, click `Manage RM Resources`.

   The `Step 1: Group List` page of the `GVP Manage <Resource Manager> Resources` wizard displays.

4. On the `Step 1: Group List` page, do one of the following:
   - To create a new group, click `New`.

     The `New Group` dialog box displays.
   - To modify the configuration parameters for an existing group, select the group name and click `Modify`.

     The `Modify Group` dialog box displays.
   - To modify the resources that are assigned to an existing group, click `Next`. Continue at Step 6.

- To delete an existing group, click `Delete`. After you confirm the deletion, click through the remainder of the Wizard to `Finish`.

Note: Deleting a group does not delete the resources themselves from the GVP deployment. However, connections from the Resource Manager to members of the deleted resource group are removed. In GVP 8.0, a resource cannot be a member of more than one group. Deleting a group therefore means that connections to all the resources in that group are removed.

5. Use the text boxes and drop-down lists in the `New Group` or `Modify Group` dialog box to specify the configuration parameters for the group. You cannot change the name of an existing group. For more information about the logical group configuration parameters, including the types of services, see Table 20 on page 131.

   After you have specified all the required options, click `OK`. You are returned to the `Step 1: Group List` page.

6. To add resources to the group, access the `Step 2: Group Resources` page in one of the following ways:
   - With the required group selected on the `Step 1: Group List` page, click `Next`.
   - With the required group selected on the `Step 1: Group List` page, click `Step 2: Group Resources` in the `Steps` panel.

7. On the `Step 2: Group Resources` page, select or clear the check boxes in the `Assigned` column to add resources to or remove resources from the group. Except for `gateway` services, the list of available resources (the `Application` list) includes only those resources in your deployment that support the service type(s) you specified for the group in Step 5. The list of `gateway` resources will include all T-Servers (including SIP Servers) in the Genesys deployment. Ensure that you select only the applicable SIP Servers to add to the `gateway` resource group. You can also add Resource Access Point `Applications` to a `gateway` resource group.

   When you select a check box to add the resource, the `Address-of-record` and `Max. Ports` fields become active.

8. Specify the required resource-specific parameters:
   a. The `Address-of-record` (`aor` parameter) for the resource.

      The `aor` parameter is a comma-separated list of the public addresses for the physical resource, in the following format:

      `<protocol>:<ip1>:<port1>[,<protocol>:<ip2>:<port2>,...]`

      where:
      - `<protocol>` is `sip` or `sips`.
      - `<ip>` is the address of record (AOR) of the resource host.

- <port> is the SIP port on the resource host.

  For example: sip:ResourceHost.company.com:5060

b. The Max. Ports (port-capacity parameter) for the resource.

The port-capacity parameter specifies the maximum number of concurrent requests the resource is capable of handling. The value can be any unsigned integer.

9. Click Next, or else click Step 3: Confirmation in the Steps panel, to access the Step 3: Confirmation page.

10. To confirm and save the changes, click Finish.

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 (see page 132).

  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 group, specify the required SIP failure response codes in the noresource-response-code option. This option does not appear in the Resource Management wizard. Configure the option on the Provisioning > Environment > Applications > <Resource Manager> > Options tab.

Table 20 provides information about the Resource Manager parameters for logical groups. Configure these parameters using the Resource Management wizard (see Step 5 on page 129) or else on the Provisioning > Environment > Applications > <Resource Manager> > Options tab, in the configuration section for the applicable logical resource group.

Table 20: Logical Group Section Configuration Options

Parameter Name	Description	Valid Values and Syntax
capability	A list of name-value pairs that identify the capabilities that are supported by resources in the logical resource group. Items in the name-value pair list are separated by a semi-colon. 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 the resource group only if the values in this option (or a subset of the name-value pairs) exactly match the capability requirements that are specified in the applicable VoiceXML or CCXML application. For example, say the value of this option is set to `lang=en-US;grammar=grxml,gsl` for a VoiceXML group. • If the `voicexml-capability-requirement` attribute in the `gvp.policy` annex of the VoiceXML application is specified as "`lang=en-US`", then the Resource Manager will send the call to the group. • If the capability is specified in the VoiceXML application as "`lang=en-US,en-IN`", then the Resource Manager will either send the call to another VoiceXML group that does support the capability, or it will reject the call (`rm.resource-no-match-respcode`—see page 283).	`<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` (for a VoiceXML group) **Default value:** Empty
load-balance-scheme	The distribution algorithm that the Resource Manager will use to select a resource within this logical resource group.	• `round-robin` • `least-used` **Default value:** `round-robin`

Table 20: Logical Group Section Configuration Options (Continued)

Parameter Name	Description	Valid Values and Syntax
monitor-method	The method the Resource Manager will use to determine if the physical resources belonging to the logical resource group are alive and healthy.	• `option`—Resource Manager will use SIP `OPTIONS` messages. • `none`—Resource Manager will not monitor resource health. The Resource Manager assumes that resources in this group are always alive. **Default value:** None
noresource-response-code	The gateway resource failure response codes that will cause the Resource Manager to retry the request on other gateway resources in the group. The Resource Manager receives the responses from SIP Server. The parameter value is a list of response codes, separated by semi-colons. • If the Resource Manager receives a failure response that matches one of the codes in the list, it keeps retrying the request on other gateway resources in the logical resource group (in other words, normal Resource Manager behavior in relation to resource groups). • If the Resource Manager receives a failure response that does not match any of the codes in the list, it forwards the response to the UAC and does not retry the request. This parameter enables more fine-grained management of gateway resources, provided the gateway has been configured to differentiate its responses.	Any valid SIP response code. **Default value:** Empty **Example:** Say `<gateway resource group>.noresource-response-code=503;504`. The gateway has been configured to provide the following response codes: • `503` for requests that fail because capacity has been reached on the gateway resource • `486` for calls that fail because the callee is busy. A request fails because: • The gateway has reached its capacity (`503` response)—The Resource Manager retries the request on another gateway in the group. • The callee is busy (`486` response)—The Resource Manager simply forwards the response to the requesting Media Control Platform or Call Control Platform, without retrying the request.

Table 20: Logical Group Section Configuration Options (Continued)

Parameter Name	Description	Valid Values and Syntax
service-types	The types of service that are provided by resources in this logical resource group. To specify multiple types of service (for example, CCXML and Conference), hold down the `Ctrl` key while selecting additional service types. Resources can be assigned to the group only if they support all the service types that you specify in this parameter. **Changes take effect:** After restart	• `voicexml`—Voice application services provided by Media Control Platform resources. • `ccxml`—Call control application services provided by Call Control Platform resources. • `gateway`—Network gateway services provided by Resource Access Point resources. • `conference`—Conference services, which can be provided by Media Control Platform and Call Control Platform resources. • `external-sip`—SIP services provided by an external SIP proxy.
port-usage-type	Determines which SIP dialogs the Resource Manager will consider when calculating the current usage on each resource, for resource management purposes. Current usage is defined as the outstanding number of established SIP dialogs on a resource plus the current pending requests on the resource. The SIP dialogs that are included in the calculation are: • For `in-and-out`—SIP dialogs originated from and directed to the resource. • `outbound`—SIP dialogs directed to the resource. **Changes take effect:** After restart	• `in-and-out`—For gateway resources only (`service-type=gateway`). • `outbound`—For all non-gateway resources. **Default value:**
resource-confmaxsize	(Applicable only for groups where `service-type=conference`.) The maximum conference size (number of participants) supported by resources in this logical resource group.	Any unsigned integer. **Default value:**

Table 20: Logical Group Section Configuration Options (Continued)

Parameter Name	Description	Valid Values and Syntax
resource-confmaxcount	(Applicable only for groups where `service-type=conference`.) The maximum number of concurrent conferences supported by each resource in this logical resource group.	Any unsigned integer. **Default value:**

Enabling High Availability

GVP uses Windows Network Load Balancing (NLB) clustering to provide High Availability (HA) for the Resource Manager.

A Cluster Manager on each Resource Manager host monitors the Network Interface Cards (NICs) in cluster mode to determine when network errors occur.

The following procedure provides an overview of the steps to create an NLB cluster and to configure the Resource Manager `Application` to operate as a node in a cluster.

For more detailed instructions about creating and configuring the cluster, see the appendix about NLB clustering in the *Genesys Voice Platform 8.0 Deployment Guide.*

Procedure: Configuring Resource Manager High Availability

Purpose: To provide a high-level description of the steps to enable High Availability for the Resource Manager in a GVP deployment.

Prerequisites

- Two Resource Manager `Applications` have been configured to function in stand-alone mode on two NLB cluster node machines, which reside in the same subnet. Each NLB cluster node machine has at least two NICs.
- An IP address has been allocated to be used as the virtual IP address.

Start of procedure

1. Set up the Windows NLB service on each Resource Manager host.

   See the *Genesys Voice Platform 8.0 Deployment Guide* for details about the required values you must set for the cluster parameters (including the virtual IP address for the cluster), host parameters, port rules, and other properties.

2. In the Genesys Administrator, create and configure a GVP Cluster Manager `Application` object for each Resource Manager `Application` in the cluster. At a minimum, you must specify the following GVP Cluster Manager parameters:
   - A space-separated list of IDs of cluster members (`Cluster.members` option). The IDs correspond to the NLB unique host identifier (priority) number that you specified for each NLB cluster machine in the NLB service properties (Step 1). The default is `1 2`.
   - For each cluster member, the IP address and TCP port on which the cluster member can be reached (`Cluster.member.<x>` option).
     - The IP address is the private IP address of the host, not the virtual IP address for the cluster.
     - For each `member.<x>`, the port number must correspond to the port specified in the `Cmcomm.tcpbondinglocalport` option of the Cluster Manager for that member. The default port is `9801`.
   - The member ID (`Cluster.mymemberid` option), which is the ID of the Cluster Manager instance, and corresponds to an ID in the list of cluster members.
   - The path to the `NLB.bat` executable (`Cluster.NLBScriptPath` option). The default path is `<Cluster Manager installation directory>\bin\NLB.bat`.
   - If necessary, also specify the value for the TCP port where this cluster manager is listening for the Resource Manager to connect (`Cmservicemgrbase.serverport` option). This port must correspond to the port specified in the Resource Manager `cmserviceagent.cmport` option. The default port is `6001`.

3. (Optional) Specify the NICs that need to be monitored. Genesys recommends that you specify the list of NICs to be monitored if the machines also have other NICs that are not involved in the NLB cluster.
   a. On the `Options` tab of the Cluster Manager `Application`, create the `gvp` section.
   b. Specify the required `nic.eth<x>` options, starting with `nic.eth0`, as name-value pairs, where the value is the MAC address of the NIC card.

4. On each Cluster Manager `Application`, create a connection to Message Server, to enable logging.

5. Configure each Resource Manager `Application` to execute in cluster mode. Specify the following options:

Section.Option	Value
`rm.cluster_ip`	`<Virtual IP address for the cluster>`
`proxy.sip.transport.0`	`transport0 udp:any:5060`
`proxy.sip.transport.1`	`transport1 tcp:<Virtual IP address for the cluster>:5060`
`proxy.sip.transport.2`	`transport2 tls:<Virtual IP address for the cluster>:5061 cert=$InstallationRoot$\config\x509_certificate.pem key=$InstallationRoot$\config\x509_private_key.pem`

6. In a text editor, edit the `<Resource Manager installation directory>\bin\init.bat` file on each cluster machine:
   - Replace the IP addresses with the virtual IP address.
   - If necessary, replace the <member id> part of <IP address>: <member id> with the correct member ID, as defined in Step 2.
7. In a text editor, edit the `NLB.bat` file on each cluster machine:
   - Replace the IP address with the virtual IP address.

End of procedure

Next Steps

- Setting up the cluster mode execution environment

Procedure: Setting up the cluster mode execution environment

Purpose: To start the Resource Manager in HA mode.

Prerequisites

- The NLB cluster and the Resource Manager `Applications` have been configured for HA (see Configuring Resource Manager High Availability, page 134).

Start of procedure

1. On each cluster machine, execute the modified `<Resource Manager installation directory>\bin\init.bat`. This disables other load-balancing and ensures that Cluster Manager, which is monitoring Resource Manager health, is the entity enabling load-balancing.
2. Start the Cluster Manager–Resource Manager pairs on each NLB cluster node. (In other words, two instances of Cluster Manager and two instances of Resource Manager will be operating.)
3. Verify that each Cluster Manager–Resource Manager instance is running as a separate Genesys `Application`.

End of procedure

Chapter

6 Provisioning IVR Profiles

IVR Profiles are the Voice Extensible Markup Language (VoiceXML) and Call Control Extensible Markup Language (CCXML) applications that control interactions with external customers. This chapter describes how to provision IVR Profiles for 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 will use specified VoiceXML or CCXML applications to control interactions that use DNs for which GVP provides service.

Prerequisites

- The VoiceXML or CCXML applications have been created and are available on a network path accessible to GVP.

  For information about developing the applications, see the *Genesys Voice Platform 8 VoiceXML 2.1 Help* and the *Genesys Voice Platform 8 CCXML*

Reference Manual. For information about using Genesys Composer to develop the applications, see *Composer Voice 8 Help.*

- You are logged in to the Genesys Administrator. To access the Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. On the `Provisioning > Voice Platform > IVR Profile` tab of the Genesys Administrator, create an `IVR Profile` object.

   The IVR Profile name that you specify is used by the Resource Manager to identify the context of the session. For more information, see "Application Identifiers" on page 72.

   For more information about creating an `IVR Profile`, see the chapter about post-installation activities in the *Genesys Voice Platform 8 Deployment Guide.*

2. Configure the IVR Profile.
   a. Use the `Provisioning > Voice Platform > IVR Profile > <IVR Profile Property> > Settings` tab and `Options` tab to view and modify values for the IVR Profile configuration options.

      For detailed information about the IVR Profile configuration options, see "IVR Profile Configuration Options" on page 141.

   b. (Optional) To specify database retention policies for the IVR Profile, to override the Reporting Server defaults for the overall GVP deployment, create a new configuration section named `dbmp` on the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Options > Annex list` tab, and configure additional `rs.db.retention.<type of data>` options as required. For more information, see the description for the equivalent `rs.db.retention.<type of data>.default` options on page 209.

   c. (Optional) Use the `Provisioning > Voice Platform > IVR Profile > <IVR Profile Property> > Number Mapping` tab to map the IVR Profile to a DNIS range. You can also specify this mapping in other ways (see "Mapping IVR Profiles to Dialed Numbers" on page 151).

3. Repeat Steps 1 and 2 as required for all the voice and call control applications that you want to provision.

4. (Optional) On the `Provisioning > Voice Platform > Dialed Number Mapping` tab, map the IVR Profiles to DNIS ranges. For more information, see "Mapping IVR Profiles to Dialed Numbers" on page 151.

5. Specify the default IVR Profile for GVP:
   a. Go to the `Provisioning > Environment > Tenants > Environment [tenant] > Options > Annex list` tab.

   b. If necessary, create the `gvp.general` configuration section.

c. Specify the name of the default IVR Profile as the value for the `gvp.general.default-application` parameter.

6. On the `Provisioning > Environment > Tenants > Environment [tenant] > Options > Annex list` tab, specify other tenant settings for using IVR Profiles. For more information, see "Specifying Tenant Environment Settings" on page 153.

Note: GVP 8.0 supports single tenancy only. Therefore, the `Environment` tenant is the only tenant.

End of procedure

IVR Profile Configuration Options

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

Configure the IVR Profile parameters in the Genesys Administrator:

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

For more information about using the Genesys Administrator to add or modify configuration sections and options, see Viewing or modifying GVP configuration parameters, page 78.

The following tables describe the IVR Profile configuration options, by configuration section:

- gvp.general Section—Table 21 on page 142
- gvp.log Section—Table 22 on page 143
- gvp.policy Section—Table 23 on page 143
- gvp.policy.dialing-rules—Table 24 on page 149
- gvp.service-parameters Section—Table 25 on page 150
- gvp.service-prerequisite Section—Table 26 on page 151

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

gvp.general Section

Table 21 describes the parameters in the `gvp.general` section.

Table 21: IVR Profile Configuration Options—gvp.general Section

Parameter Name	Description	Valid Values and Syntax
application-confmax 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 for the conference resource itself (see `resource-confmaxsize` on page 133 and the Media Control Platform `conference.limit` parameter on page 165).	Any unsigned integer. **Default value:** `20`
service-type	The default type of service 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` **Default value:** `voicexml`
sip.sessiontimer	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 tenant, the Resource Manager, and the resource.	Any unsigned integer. **Default value:** Empty

gvp.log Section

Table 22 describes the parameter in the `gvp.log` section. This parameter enables you to specify the metrics filters that all resources will use for logging data, when they provide services for the IVR Profile.

Table 22: IVR Profile Configuration Options—gvp.log Section

Parameter Name	Description	Valid Values and Syntax
metricsfilter	The filter that determines which metrics in the Data Collection Sink (DATAC) will be forwarded to the Reporting Server. If this parameter is set, the value will override the default DATAC filter for the component (see ems.dc.default.metricsfilter on page 105), for sessions that execute under this IVR Profile. 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 8.0 Metrics Reference.* The wildcard character (*) means all. **Default value:** Empty

gvp.policy Section

Table 23 describes the parameters in the gvp.policy section. 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.

Table 23: IVR Profile Configuration Options—gvp.policy Section

Parameter Name	Description	Valid Values and Syntax
conference-allowed	Specifies whether a Resource Manager session is allowed to use a conference.	• True • False **Default value:** True
dialing-rule-forbidden-respcode	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 23: IVR Profile Configuration Options—gvp.policy Section (Continued)

Parameter Name	Description	Valid Values and Syntax
dialing-rule-forbidden-set-alarm	Specifies whether an alarm will be raised for the corresponding policy violation.	• `True` • `False` **Default value:** `False`
external-sip-allowed	Specifies whether a Resource Manager session is allowed to use an external SIP service.	• `True` • `False` **Default value:** `True`
inbound-usage-limit	The number of times this IVR Profile may be concurrently in use for an inbound call.	Any unsigned integer. **Default value:** Empty
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 164 for more information about the implications of this setting.	• `per-call` • `per-utterance` **Default value:** `per-call`
mcp-max-log-level	The Maximum Log Level allowed for a Media Control Platform application. The Resource Manager passes this value to the Media Control Platform in a `Request-URI` parameter.	An alphanumeric string. **Default value:** Empty

Table 23: IVR Profile Configuration Options—gvp.policy Section (Continued)

Parameter Name	Description	Valid Values and Syntax
mcp-sendrecv-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
outbound-call-allowed	Specifies whether a Resource Manager session is allowed to make an outbound call. An outbound call in this context is a call through a gateway or to a SIP device not managed by the Resource Manager.	• True • False **Default value:** True
outbound-usage-limit	The number of times this IVR Profile may be concurrently in use for an outbound call.	Any unsigned integer. **Default value:** Empty
<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 Items in the name-value pair list are separated by a semi-colon. 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 specified in this option (see the <Logical Group>.capability option in Table 20 on page 131).	<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 23: IVR Profile Configuration Options—gvp.policy Section (Continued)

Parameter Name	Description	Valid Values and Syntax
<service>-forbidden-respcode	The SIP response code that is sent in the SIP response when a request for the service is rejected because the service is not allowed in the session (`gvp.policy.<service>-allowed=false`). Valid values for `<service>` are: • `conference` • `external-sip` • `outbound-call` • `transfer` • `voicexml-dialog`	• `<sipcode>;<desc>` • `<sipcode>` where: • `<sipcode>` is an integer in the range of 400–699. • `<desc>` is any string. **Default value:** `403`
<service>-forbidden-set-alarm	Specifies whether an alarm will be raised for the corresponding policy violation. Valid values for `<service>` are: • `ccxml` • `conference` • `external-sip` • `outbound-call` • `transfer` • `voicexml-dialog`	• `True` • `False` **Default value:** `False`
<service>-usage-limit	The number of times the specified service may be invoked in the context of this IVR Profile. Valid values for `<service>` are: • `ccxml` • `conference` • `voicexml`	Any unsigned integer. **Default value:** Empty
<service>-usage-limit-exceeded-respcode	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`	• `<sipcode>;<desc>` • `<sipcode>` where: • `<sipcode>` is an integer in the range of 400–699. • `<desc>` is any string. **Default value:** `503`

Table 23: IVR Profile Configuration Options—gvp.policy Section (Continued)

Parameter Name	Description	Valid Values and Syntax
<service>-usage-limit-exceeded-set-alarm	Specifies whether an alarm will be raised for the corresponding policy violation. Valid values for `<service>` are: • `ccxml` • `conference` • `external-sip` • `inbound` • `outbound` • `voicexml`	• `True` • `False` **Default value:** `False`
<service>-usage-limit-per-session	The number of times the specified service may be invoked in the context of this instance of a Resource Manager session. Valid values for `<service>` are: • `ccxml` • `conference` • `voicexml`	Any unsigned integer. **Default value:** Empty
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`
usage-limit-exceeded-respcode	The SIP response code that is sent in the SIP response when a call is rejected because the usage limits 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)
usage-limit-exceeded-set-alarm	Specifies whether an alarm will be raised for the corresponding policy violation.	• `True` • `False` **Default value:** `False`
usage-limits	The number of times this IVR Profile may be concurrently in use. If a value is not specified (the default), there are no usage limits for the IVR Profile.	Any unsigned integer. **Default value:** Empty

Table 23: IVR Profile Configuration Options—gvp.policy Section (Continued)

Parameter 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`
voicexml-dialog-allowed	Specifies whether a Resource Manager session is allowed to use a VoiceXML service.	• `True` • `False` **Default value:** `True`

gvp.policy.dialing-rules

Table 24 describes the parameters in the `gvp.policy.dialing-rules` subsection.

Table 24: IVR Profile Configuration Options—gvp.policy.dialing-rules Section

Parameter Name	Description	Valid Values and Syntax

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*`
		<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

gvp.service-parameters Section

Table 25 describes the parameters in the gvp.service-parameters section. The Resource Manager uses these values to add, modify, or delete Request-URI parameters in the SIP requests that it forwards.

Table 25: IVR Profile Configuration Options—gvp.service-parameters Section

Parameter Name	Description	Valid Values and Syntax
<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: • `ccxml` • `conference` • `external` • `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`.

gvp.service-prerequisite Section

Table 26 describes the parameters in the `gvp.service-prerequisite` section. These parameters provide information that the Resource Manager needs when it uses a default IVR Profile.

Table 26: IVR Profile Configuration Options—gvp.service-prerequisite Section

Parameter Name	Description	Valid Values and Syntax
alternatevoicexml	(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
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 containing 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 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

Mapping IVR Profiles to Dialed Numbers

Dialed numbers are the DNs 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 126).

- If your GVP configuration includes a mapping of IVR Profiles to DNs, 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 DNs, the Resource Manager will use the default IVR Profile that you specify for the tenant (see `gvp.general.default-application` on page 154).

There are two ways to map IVR Profiles to DNs:

- Create rules on the `Provisioning > Voice Platform > Dialed Number Mapping` tab. All IVR Profiles in the GVP environment are available for selection.
- Create rules for a particular IVR Profile on the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Number Mapping` tab.

Whichever method you use, the mapping is updated in both contexts, as well as on the `Provisioning > Environment > Tenants > <tenant> > Options > Annex list` tab (`gvp.dnis-range.<range>` parameters).

The following procedure describes how to create mapping rules on the `Voice Platform` tab.

Procedure: Mapping IVR Profiles to DNs

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

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 Deployment Guide*. For more information about configuring the IVR Profile, see "IVR Profile Configuration Options" on page 141.
- You are logged in to the Genesys Administrator. To access the Genesys Administrator, go to the following URL:

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

Start of procedure

1. Go to the `Provisioning > Voice Platform > IVR Profile > <IVR Profile> > Number Mapping` tab.
2. In the menu bar of the tab, click `New Rule`.

   The `Specify Dialed Number Mapping Rule` dialog box displays (see Figure 8).

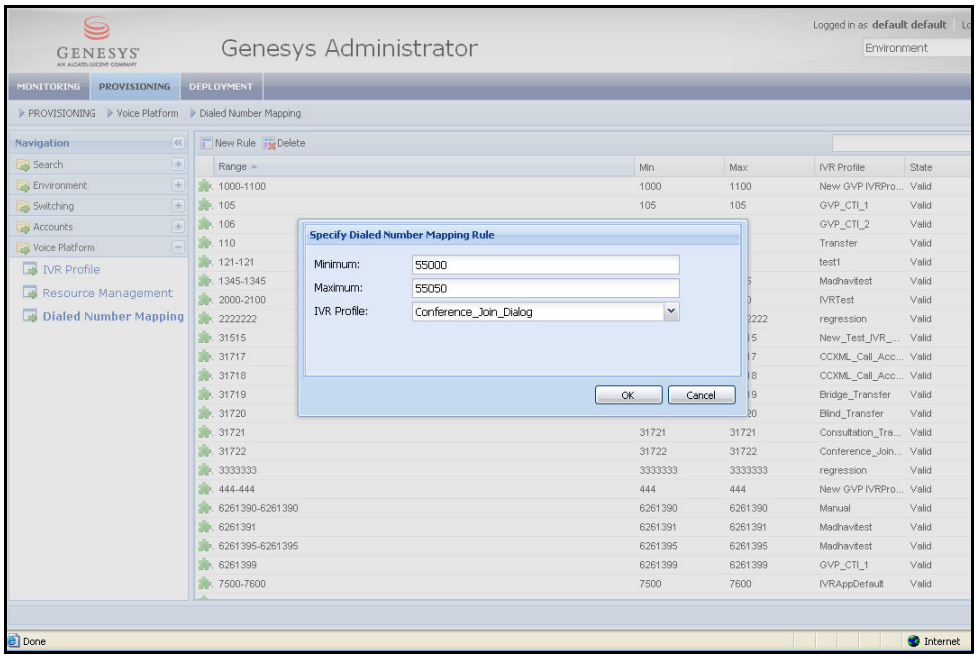


Figure 8: Specifying a Dialed Number Mapping Rule

3. In the `Minimum` and `Maximum` text boxes, enter the start and end numbers of the DN range. For a single DN, rather than a range, enter the same number in both text boxes.
4. From the `IVR Profile` drop-down list, select the required IVR Profile.
5. Click `OK`.

   The list of mapping rules on the number mapping tab is updated to include the new rule. The display also includes a column (`State`) and icon that indicate the status of the rule—valid or invalid.

   A range is valid if each DN in the range is not already specified as part of another mapping.
6. If you were specifying the rule on the IVR Profile `Number Mapping` tab, click `Apply`, then `Save`.

End of procedure

Specifying Tenant Environment Settings

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 `Environment` tenant parameters in the Genesys Administrator on the `Provisioning > Environment > Tenants > Environment [tenant] > Options > Annex list` tab.

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

The following tables describe the `Tenant` configuration options:

- gvp.dnis-range Section
- gvp.general Section—Table 27
- gvp.policy Section—see page 155 and Table 23 on page 143
- gvp.service-parameters Section—see page 155 and Table 25 on page 150

gvp.dnis-range Section

The parameter list in this section shows the mapping of dialed numbers to IVR Profiles. The list is automatically updated when you use one of the methods to map a DNIS range to an IVR Profile (see "Mapping IVR Profiles to Dialed Numbers" on page 151).

Do not create or modify the mapping from this section.

gvp.general Section

Table 27 describes the parameters in the `gvp.general` section. These parameters specify general configuration information for the Resource Manager in the tenant's environment.

Table 27: Tenant Configuration Options—gvp.general Section

Parameter Name	Description	Valid Values and Syntax
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

Table 27: Tenant Configuration Options—gvp.general Section (Continued)

Parameter Name	Description	Valid Values and Syntax
sip.sessiontimer	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 “Session Management” on page 27.	Any positive integer. **Default value:** Empty
usage-limits	The number of times a Resource Manager session can be concurrently in use in the context of any IVR Profile.	Any unsigned 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.

For more information about the configuration options in the `gvp.policy` section, see Table 23 on page 143.

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 25 on page 150.

Chapter

7 Configuring the Media Control Platform

The Media Control Platform is the Genesys Voice Platform (GVP) component that provides media-centric services. This chapter provides information about configuring the Media Control Platform and, if required, provisioning 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

Table 28 summarizes the configuration steps and options to implement Media Control Platform functionality in your GVP deployment.

Table 28: Configuring the Media Control Platform

Objective	Related Procedures and Actions
Integrate the Media Control Platform with the Resource Manager.	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` (see page 93) • `sip.routeset` or `sip.secureroutеset` (see page 90) 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 Communications and Routing" on page 86.

(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` or `offer`) in the `mpc.srtp.mode` parameter (see page 166). 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`
	For more information about secure communications in GVP, see "Secure Communications" on page 70.

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 159.
Configure conferencing.	See "Enabling Conference Services" on page 102.
Configure EMS Reporting.	See "Configuring EMS Reporting" on page 103.
Configure logging.	See "Configuring Logging" on page 111.

Table 28: Configuring the Media Control Platform (Continued)

Objective	Related Procedures and Actions
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.stack.connection.timeout` (default is `10000 ms`) • See also "Configuring Session Timers and Timeouts" on page 117. • 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 the `mediamgr.*` options in the `mcp` configuration section, to verify that they are optimal for your deployment.
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 8.0 Configuration Options Reference*). Some of the important `vxmli` options are described in Table 29 on page 164. • 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 71 on page 295.
Customize session management behavior and performance.	See "Configuring Session Timers and Timeouts" on page 117.
Customize Media Control Platform messaging.	See "Customizing SIP Responses" on page 116 and Table 64 on page 279.

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 `Third Party Server`.

Prerequisites

- You are logged in to the Genesys Administrator. To access the 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
      - MRCPv2_ASR
      - MRCPv1_TTS
      - MRCPv2_TTS

   b. Import metadata into the `Application Template`.
   c. On the `Provisioning > Environment > Applications` tab, create and name the new `Third Party Server Application`, based on the required `Application Template`.

   For detailed information about importing `Application Templates` and metadata, and creating `Applications` from the templates, see the appendix about pre-installation activities in the *Genesys Voice Platform 8.0 Deployment Guide.*
2. On the `Provisioning > Environment > Applications > <MRCP Server> > Settings` tab, configure the options in the `provision` configuration section, as required for your deployment.
   a. At a minimum, you must specify values for the following options:
      - `vrm.client.resource.address`—The IP address of the speech resource.
      - `vrm.client.resource.uri`—The URI to the speech resource.

- vrm.client.resource.port—The port on the speech resource for communication with the Media Control Platform MRCP Client. The default value is 4900 for MRCPv1, and 5060 for MRCPv2.

b. If necessary, modify the vrm.client.resource.name parameter to add the name of your speech resource vendor. Modify this parameter:

- For ASR, if your vendor is not SPEECHWORKS. Continue at Step 3.
- For TTS, if your vendor is not REALSPEAK. Continue at Step 4.

For other important configuration options that you may need to modify, see "Important MRCP Server Configuration Options" on page 178.

3. If you added an ASR speech vendor (Step 2b), you must also update the hotkey grammar directory and path.

a. On the Provisioning > Environment > Applications > <MRCP Server> > Settings tab, modify the following parameters to use the name of your ASR vendor (vrm.client.resource.name):

- vrm.client.HotKeyBasePath=/vggrammarbase/<vendor name>/hotkey
- vrm.client.HotKeyLocalPath=$InstallationRoot$/grammar/<vendor name>/hotkey

For more information about these parameters, see page 181.

b. Click Save or Apply to save the MRCP server configuration.

c. On the Media Control Platform host, go to the grammar directory in the path specified in vrm.client.HotKeyLocalPath.

d. Create a new directory with the <vendor name> specified in vrm.client.resource.name.

e. In the <vendor name> directory, create a new directory called hotkey.

f. Copy the hotkey grammar file from $InstallationRoot$\grammar\<vendor name>/hspeechworks\hotkey to the new hotkey directory.

4. 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.

5. On the Provisioning > Environment > Applications > <Media Control Platform> > Configuration tab, create the connection between the Media Control Platform and the MRCP server. For more details, see the procedure to assign the MRCP server to the Media Control Platform, in the chapter about post-installation activities in the *Genesys Voice Platform 8.0 Deployment Guide.*

End of procedure

Important Media Control Platform Configuration Options

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

Configure the options on the Genesys Administrator on the Provisioning > Environment > Applications > <Media Control Platform> > Settings tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

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

- asr
- calllog
- conference—Parameters determine the default behavior of the Conference application module, for NETANN-initiated conference calls.
- email—Parameters enable you to configure e-mail address information for maintainer e-mails.
- ems (see Table 13 on page 104)—Parameters determine EMS Reporting behavior for call detail records (CDRs) and metrics.
- log (see "Configuring Logging" on page 111)—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.
- 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.
- `stack`—Parameters relate to the MRCP stack and determine the way the Media Control Platform manages connections to the external MRCP server.
- `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 29 provides information about important Media Control Platform parameters that are not described in Chapter 4 on page 85. Table 29 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 8.0 Configuration Options Reference.*

Table 29: Selected Media Control Platform Configuration Options

Option Name	Description	Valid Values and Syntax
`asr` Section		
load_once_per_call	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.	• `0`—Single session not enabled. • `1`—Single session enabled. **Default value:** `1` (only one VRM session for the entire call)
`conference` Section		
audio_format	The audio codec for conference.	• `pcmu` • `pcma` • `g726-16` • `g726-24` • `g726-32` • `g726-40` • `l16` • `gsm` **Default value:** `pcmu`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
limit	The maximum number of participants allowed for a conference initiated by the conferencing application.	An integer in the range 2–32. **Default value:** `32`
`mpc` Section		
appendrejcodec	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`
codec	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` • `g726` • `g729` • `gsm` • `h263` • `h263-1998` • `pcmu` • `pcma` • `telephone-event` • `tfci` **Default value:** `pcmu pcma g726 gsm h263 h263-1998 telephone-event`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
codecpref	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`
default_audio_format	The default audio format for the Call Manager.	• `ALAW` • `ULAW` **Default value:** `ULAW`
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. **Default value:** `none`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
`sip` Section		
(**Note:** For additional important options in this configuration section, see also "Configuring SIP Communications and Routing" on page 86.)		
confserver	(Mandatory if a conferencing server is deployed in the network) The address of the GVP conferencing server. **Note:** Underscores are not valid characters in the IP address.	`<IP address>[:<port>]` where: • `<IP address>` is the IP address of the conferencing server host. • `<port>` is the SIP port number. You must specify this value if the conferencing server uses a non-default SIP port (default is `5060`). **Default value:** Empty
defaultblindxfer	The default transfer method for SIP, for blind transfers. For more information about the transfer types and methods, see "Transfers" on page 45.	• `HKF`—Hookflash • `REFER`—`REFER`-based transfer • `BRIDGE`—Bridge-based transfer • `REFERJOIN`—Consultative `REFER` transfer • `MEDIAREDIRECT`—Media redirect transfer **Default value:** `REFER`
defaultbridgexfer	The default transfer method for SIP, for bridge-type transfers. For more information about the transfer types and methods, see "Transfers" on page 45.	• `BRIDGE`—Bridge-based transfer • `MEDIAREDIRECT`—Media redirect transfer **Default value:** `BRIDGE`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
defaultconsultxfer	The default transfer method for SIP, for consult-type transfers. For more information about the transfer types and methods, see "Transfers" on page 45.	• `HKF`—Hookflash • `BRIDGE`—Bridge-based transfer • `REFERJOIN`—Consultative `REFER` transfer • `MEDIAREDIRECT`—Media redirect transfer **Default value:** `REFERJOIN`
defaultgw	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
defaulthost	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

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
deferoutalerting	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`
dnis_correlationid_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_correlationid_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)
enablesendrecv 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`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
hfdisctimer	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`
hfprefix	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 containing one or more of the following characters: `0–9 , ! * none` **Default value:** `!`
hfstopdial	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 containing one or more of the following characters: `0–9 !` **Default value:** `!`
hftype	Specifies the type of hookflash transfer for SIP.	• `0`—Wait for disconnection. • `1`—Force disconnection. **Default value:** `0`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
in.<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, then none is ignored. **Example:** From To Via **Default values:** • For BYE requests: Reason • For INFO and INVITE requests: *
in.invite.params	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.contenttype	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 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
localrtpaddr	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 multihomed 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)
out.<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`, then `none` is ignored. **Example:** `From To Via` **Default value:** `*`
out.<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`, then `none` is ignored. **Default value:** `RequestURI`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
outcalluseoriggw	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`
referxferhold	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 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
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 deregister when it shuts down. **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`.	`<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. • `<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. • `<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

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
sendalert	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`
sipinfoallowedcontenttypes	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:** Empty
transfermethods	A space-separated list of the supported transfer methods for SIP. For more information about the transfer methods, see “Transfer Methods” on page 46.	• `HKF`—Hookflash • `REFER`—`REFER`-based transfer • `REFERJOIN`—Consultative `REFER` transfer • `MEDIAREDIRECT`—Media redirect transfer • `none`—No transfer methods for SIP **Default value:** `REFER REFERJOIN MEDIAREDIRECT`

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
vxmlinvite	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`
warningheaders	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`
xfer.copyheaders	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 rescanned 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 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
vrm Section		
client.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
client.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
client.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
client.universals.uri	The URI convention that the NGI uses to specify the universals grammars.	builtin:grammar/universals **Default value:** builtin:grammar/universals
vxmli Section		
conformance.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

Table 29: Selected Media Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
debug.enabled	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`
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 the Genesys Administrator on the `Provisioning > Environment > Applications > <MRCP Server> > Settings` tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

The configurable MRCP server options are in the `provision` configuration section. Table 29 provides information about these options. Table 29 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 8.0 Configuration Options Reference.*

Table 30: Selected MRCP Server Configuration Options

Option Name	Description	Valid Values and Syntax
ASR and TTS		
vrm.client.ConnectPerSetup	(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`
vrm.client.resource.address	The IP address of the speech resource.	`<IP address>` **Default value:** `0.0.0.0`
vrm.client.resource.name	The name of the speech resource vendor.	<vendor_name> **Default value:** • For ASR: `SPEECHWORKS` • For TTS: `REALSPEAK`
vrm.client.resource.port	The port on the speech resource that will be used for communication with the Media Control Platform MRCP Client.	`<port>` where `<port>` is an integer in the range of `1`-`60000`. **Default value:** • For ASR servers: `5060` • For TTS servers: `4900`

Table 30: Selected MRCP Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
vrm.client.resource.uri	The URI to the speech resource.	• For MRCPv1 ASR: `rtsp://<IP address>:<port>/media/speechrecognizer` • For MRCPv1 TTS: `rtsp://<IP address>:<port>/media/speechsynthesizer` • For MRCPv2: `sip:mresources@<IP address>:<port>` **Default value:** • For MRCPv1 ASR: `rtsp://0.0.0.0:4900/media/speechrecognizer` • For MRCPv1 TTS: `rtsp://0.0.0.0:4900/media/speechsynthesizer` • For MRCPv2: `sip:mresources@0.0.0.0:5060`
ASR Only		
vrm.client.DisableHotWord	Specifies whether the platform will treat 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`

Table 30: Selected MRCP Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
vrm.client.HotKey BasePath	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 specified in `vrm.client.resource.name` on page 179.	`/vggrammarbase/<vendor name>/hotkey` **Default value:** `/vggrammarbase/speechworks/hotkey`
vrm.client.HotKey LocalPath	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:** `$InstallationRoot$/grammar/speechworks/hotkey`

Chapter

8 Configuring the Call Control Platform

This chapter provides information about configuring the Call Control Platform and provisioning the device profiles in your Genesys Voice Platform (GVP) deployment.

It contains the following sections:

Task Summary: Configuring the Call Control Platform

Table 31 summarizes the configuration steps and options to implement Call Control Platform functionality in your GVP deployment.

Table 31: Configuring the Call Control Platform

Objective	Related Procedures and Actions
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` (see page 188) • `mediacontroller.bridge_server` (see page 187) • `sip.transport.x` (see page 92) • `sip.routeset` or `sip.securerouteset` (see page 90) For additional, relevant configuration options and actions, see "Configuring SIP Communications and Routing" on page 86 and "Enabling Secure Communications" on page 93.

(Required only if you made TCP or TLS the preferred default transport protocol [see page 86])
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 8.0\MCP_80\helloaudio.vxml'" hints="hints"/>`

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 188.
Configure conferencing.	See "Enabling Conference Services" on page 102.
Configure EMS Reporting.	See "Configuring EMS Reporting" on page 103.
Configure logging.	See "Configuring Logging" on page 111.

Table 31: 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 `1000`) • `ccxmli.num_session_processing_threads` (default is `5`) • `ccxmli.max_num_sessions` (default is `1000`) • `ccxmli.max_conn_per_session` (default is `100000`) • `ccxmli.max_dialog_per_session` (default is `100`) • `ccxmli.max_conf_per_session` (default is `100000`) See also "Configuring Session Timers and Timeouts" on page 117.
Customize session management behavior and performance.	See "Configuring Session Timers and Timeouts" on page 117.
Customize Call Control Platform messaging.	See "Customizing SIP Responses" on page 116 and Table 64 on page 279.

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 the Genesys Administrator on the `Provisioning > Environment > Applications > <Call Control Platform> > Settings` tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

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 13 on page 104)—Parameters determine EMS Reporting behavior for call detail records (CDRs) and metrics.
- `log` (see "Configuring Logging" on page 111)—Parameters determine behavior for Management Framework logging.
- `mediacontroller` and `mediactrller`—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).

Table 32 provides information about important Call Control Platform parameters that are not described in Chapter 4 on page 85. Table 32 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 8.0 Configuration Options Reference*.

Table 32: Selected Call Control Platform Configuration Options

Option Name	Description	Valid Values and Syntax
`ccpccxml` Section		
default_uri	The URI for the default CCXML application.	`<URI path to file>` **Default value:** `file://$InstallationRoot$\config\default.ccxml`
sip.allowedunknown headers	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`	A string specifying a header name. **Default value:** Empty

Table 32: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
`ccxmli` Section		
platform.save_`<file type>`_files	`<file type>` is either `ccxml` or `script`. 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`
`mediacontroller` Section		
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
bridge_server.profile	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 188.	`<Device profile name>` **Default value:** `Default Conference`

Table 32: Selected Call Control Platform Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
full_<media type>_codec	A space-separated list of the `<media type>` 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 connectionless SDP. `<media type>` is either `audio` or `video`. For the codecs that the Media Control Platform and, therefore, the Call Control Platform support, see "Codec Negotiation" on page 43.	`<payload>|<codec>|<MIME-type>|<rate>|<number of channels>` **Default values:** • Audio— `0|pcmu|audio/basic|8000|1 8|pcma|audio/x-alaw-basic|8000|1 4|g723|none|8000|1 18|g729|audio/g729|8000|1 12|qcelp|audio/QCELP|8000|1 111|aurora|none|8000|1 3|gsm|audio/x-gsm|8000|1 105|AMR|audio/AMR|8000|1 101|telephone-event|none|8000|1` • Video— `34|h263|video/H263|90000|1 98|H264|video/H264|90000|1`
inbound_allowed_media	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.	• `audio` • `audio video` • `video` **Default value:** `audio`
sipproxy	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
`sip` Section		
See Table 9 on page 88.		

Configuring Device Profiles

The Call Control Platform uses device profiles to determine the behavior of the devices with which it interacts, in order to produce the most appropriate SIP messages. (For more information about how the Call Control Platform uses device profiles, see "Device Profiles" on page 53.)

Device Profile Configuration File

Device profiles are defined in the `<Call Control Platform Installation Directory>\config\ccpccxml_provision.dat` file, which is installed when you install the Call Control Platform.

For the format and syntax of device profile entries in the configuration file, see Provisioning Device Profiles for the Call Control Platform, page 194.

For information about the properties that have been defined for the `CCXML Device` class of profiles, see Table 33.

For information about the values that have been predefined for the default `CCXML Device` class of profiles, see Appendix D, “Default Device Profiles,” on page 304.

CCXML Device Class

Table 33 describes the properties that define the `CCXML Device` class of device profiles.

Table 33: CCXML 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=0.0.0.0`). • `non-routable`—Use SDP with a non-routable connection (for example, `c=1.1.1.1`). • `none`—The device does not support connectionless SDP.
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 CCXML Device supports a `sendonly` media line and a `recvonly` media line on separate connections. • `false`—The CCXML 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 CCXML Device supports receiving the `recvonly` attribute in multiple SDP media lines. • `false`—The CCXML Device does not support receiving the `recvonly` attribute in multiple SDP media lines.

Table 33: CCXML 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 CCXML Device supports receiving SDP without media lines. • `false`—The CCXML 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 CCXML Device supports the offer-answer model. • `false`—The CCXML 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 CCXML Device supports receiving `INVITE` requests that do not contain an SDP offer. • `false`—The CCXML Device does not support receiving `INVITE` requests that do not contain an SDP offer.
options-support	Support for SIP `OPTIONS`. Indicates if the device supports the SIP `OPTIONS` request. The device includes its SDP capabilities in the body of the `OPTIONS` response.	• `true`—The CCXML Device supports receiving `OPTIONS` requests. • `false`—The CCXML Device does not support receiving `OPTIONS` requests.

Table 33: CCXML Device Profile Class Properties (Continued)

Property	Description	Valid Values
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 CCXML Device supports receiving the `recvonly` attribute in the SDP media line. • `false`—The CCXML Device does not support receiving the `recvonly` attribute in the SDP media line.
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 CCXML Device supports receiving media from the UA to which media is being sent, and only from that UA. • `false`—The CCXML 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 CCXML Device supports receiving the `sendonly` attribute in the SDP media line. • `false`—The CCXML Device does not support receiving the `sendonly` attribute in the SDP media line.

Table 33: CCXML Device Profile Class Properties (Continued)

Property	Description	Valid Values
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`). • `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.

Table 33: CCXML Device Profile Class Properties (Continued)

Property	Description	Valid Values
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). • nomedia-SDP—If the value of both nomedia-SDP-support and options-support is true, an INVITE with no-media SDP will be sent to the endpoint. If the value of either or both nomedia-SDP-support and options-support is false, 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.

Customizing Device Profiles

VP Call Control Platform 8.0 is 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. For details about the default device profile attributes, see Appendix D on page 304.

Your deployment may require the Call Control Platform to 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 can use to identify the device, Genesys recommends that you create a new device profile.
- If the SIP request does not include headers that the Call Control Platform can use 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 to the Call Control Platform, 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 *Select Profile,* and match the incoming `INVITE` to the device profile selection. Then review the parameter values for that profile (see Table 74 on page 305) to identify the parameters you need to change.

The following procedure describes how to modify the device profile provisioning file.

Procedure: Provisioning Device Profiles for the Call Control Platform

Purpose: To modify the `ccpccxml_provision.dat` file to enable the Call Control Platform to interact with non-default SIP devices.

Prerequisites

- The Call Control Platform 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 `ccpccxml_provision.dat` file, in case you later want to restore the original settings.
2. Open the `<Call Control Platform Installation Directory>\config\ccpccxml_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 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.

  The Call Control Platform attempts to match the value of this property to CCXML hints in outbound connections, dialogs, and conferences.
- `<Device Profile Class Name>` is the class of device profiles to which this device profile belongs.

  The only class that has been defined by default is `CCXML Device`. For the properties and default values that have been defined for the `CCXML Device` class, see "CCXML Device Class" on page 189.
- `<# 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 will attempt to match, to assign the device profile for inbound connections. Spaces are not allowed.

  The value for the predefined `CCXML Device` class is `User-Agent`.
- `<Regex>` is the expression that the Call Control Platform 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.

End of procedure

Chapter

9 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

Table 34 summarizes the configuration steps and options to configure the Fetching Module and Squid Caching Proxy in your GVP deployment.

Table 34: 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 200. 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 8.0 Deployment Guide.*

Important Fetching Module Configuration Options

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

Configure the options in the 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 Viewing or modifying GVP configuration parameters, page 78.

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 13 on page 104)—Parameters determine EMS Reporting behavior for call detail records (CDRs) and metrics.
- `log` (see "Configuring Logging" on page 111)—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).

Table 35 provides information about some important Fetching Module parameters in the `iproxy` section. Table 35 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 8.0 Configuration Options Reference*.

Table 35: Selected Fetching Module Configuration Options

Option Name	Description	Valid Values and Syntax
`iproxy` Section		
<protocol>_proxy	`<protocol>` is either HTTP or HTTPS. 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_substr	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 Communications" on page 93.	

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 (`C:\squid\etc\squid.conf`) in a text editor.

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`

   For more information about how Squid uses the refresh-pattern rules to determine the freshness or staleness of an object, see “Squid Expiry Time Algorithm” on page 316.

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
     ```

   - Restart the SquidNT service.

   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.

Chapter

10 Configuring the Reporting Server

This chapter provides information about configuring the Reporting Server. It contains the following sections:

Task Summary: Configuring the Reporting Server

Table 36 summarizes the configuration steps and options to set up the Reporting Server and to customize EMS Reporting behavior in your GVP deployment.

Table 36: Configuring the Reporting Server

Objective	Related Procedures and Actions
Verify directory paths for: • Java Message Service (JMS) for CDR 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`). • `transaction`.

Table 36: Configuring the Reporting Server (Continued)

Objective	Related Procedures and Actions
Configure logging.	See "Configuring Logging" on page 111. Note that the Reporting Server does not support time-based log rollover.
Configure the maximum size of reports for different levels of granularity (5-minute period, hour, day, week, month).	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 (see page 209), 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 (see page 208), for the maximum number of pages per report. The default is `10`.
Configure database retention policies.	If necessary, modify settings for the options in the `dbmp` configuration section. For more information, see "Configuring Database Retention Policies" on page 206.
Configure Reporting Server behavior in general.	See "Important Reporting Server Configuration Options" on page 207.
(Optional) Configure HTTP Basic Authorization to secure access to Reporting services.	See "Controlling Access to Reporting Services" on page 210.
Verify that the Genesys Administrator displays GVP reports requested from the `Monitoring > Voice Platform` navigation panel.	If necessary, configure or modify the connection between the Genesys Administrator and the Reporting Server. For more information, see "Connecting to the Genesys Administrator" on page 213.

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

- 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 210) are appropriate for your reporting purposes and environment.

Table 37 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.

Table 37: Default Maximum Units, by Granularity Level

Aggregation Period	Maximum Number of Units
5 minutes	288 (5-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 (see page 209)—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 specificVoiceXML or CCXML application.

Table 38 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.

Table 38: 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
Call log events (upstream logs)	N/A	rs.db.retention.events.default	> 0	7
Operational data	5-minute	rs.db.retention.operations.5min.default	> 0	1
	Daily	rs.db.retention.operations.daily.default	> 30	90
	Hourly	rs.db.retention.operations.hourly.default	> 0	7
	Monthly	rs.db.retention.operations.monthly.default	> 30	1095 (36 months)
	Weekly	rs.db.retention.operations.weekly.default	> 30	364 (52 weeks)

Table 38: Default Database Retention Periods (Continued)

Type of Data	Granularity	Option Name in dbmp Section	Minimum Valid Value (integer)	Default Value, in days
VAR summary statistics (Call Summary and IVR Action statistics)	5-minute	rs.db.retention.var.5min.default	> 0	1
	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)

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 the Genesys Administrator on the `Provisioning > Environment > Applications > <Reporting Server> > Settings` tab. For the detailed steps to configure option settings, see Viewing or modifying GVP configuration parameters, page 78.

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

- `cdr`—Parameters determine behavior for processing and reporting on call detail records (CDRs).
- `dbmp`—Parameters determine database retention policies.
- `log` (see "Configuring Logging" on page 111)—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 periodic tasks.
- `transaction`—Parameters provide the directory paths for the Atomikos distributed transactions processing engine.

Table 39 provides information about important Reporting Server parameters that are not described in Chapter 4 on page 85. Table 39 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 8.0 Configuration Options Reference.*

Table 39: Selected Reporting Server Configuration Options

Option Name	Description	Valid Values and Syntax
`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). • 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 210). The timeout process uses the value of the `call-timeout` parameter to identify calls that have timed out since the process last ran.	An integer in the range of `1-1440`. **Default value:** `180` (3 hours)
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 39: 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`
`dbmp` Section		
rs.db.retention.<type of data>.default	The number of days for which `<type of data>` will be retained in the Reporting database, where `<type of data>` is: • `cdr`—CDR data • `events`—call log events (upstream logs) data • `operations.5min`—5-minute operational data • `operations.daily`—daily operational data • `operations.hourly`—hourly operational data • `operations.monthly`—monthly operational data • `operations.weekly`—weekly operational data • `var.5min`—5-minute Call Summary and IVR Profile Summary data • `var.daily`—daily Call Summary and IVR Profile Summary data • `var.hourly`—hourly Call Summary and IVR Profile Summary data • `var.monthly`—monthly Call Summary and IVR Profile Summary data • `var.weekly`—weekly Call Summary and IVR Profile Summary data **Note:** You can specify non-default `dbmp.rs.db.retention.<type of data>` options for individual IVR Profiles. If configured, the IVR Profile `rs.db.retention.<type of data>` value overrides the value configured in the Reporting Server `dbmp.rs.db.retention.<type of data>.default` option.	See Table 38 on page 206.

Table 39: Selected Reporting Server Configuration Options (Continued)

Option Name	Description	Valid Values and Syntax
`reporting` Section		
rs.query.limit.<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 • `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 38 on page 206. **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.
`schedule` Section		
quartz.rs.calltimeout	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 on page 208).	See `http://www.opensymphony.com/quartz/api/org/quartz/CronExpression.html` and `http://quartz.sourceforge.net/javadoc/org/quartz/CronTrigger.html`. **Default value:** `0 50 * * * ?`
quartz.rs.dbMaintenancePeriod	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.	See `http://www.opensymphony.com/quartz/api/org/quartz/CronExpression.html`. **Default value:** `0 0 1 * * ?`

Controlling Access to Reporting Services

GVP 8.0 leverages the HTTP Basic Authorization features of Apache Tomcat to authenticate users and control access to Reporting Web Services.

GVP 8.0 does not support selective, role-based access for different categories of Reporting services.

The following procedure describes how to configure Tomcat and Management Framework so that the Genesys Administrator will silently provide authentication information to the Reporting Server when it invokes Reporting Web Services.

Procedure: Enabling HTTP Basic Authorization for Reporting

Purpose: To enable HTTP Basic Authorization for Reporting, by modifying the configurations of Tomcat and the Management Framework user interface (UI).

Start of procedure

1. Shut down the Apache Tomcat process.
2. Define the user name and password that the Genesys Administrator will be required to provide for authentication, in order to access Reporting services.
   a. Open the `<Tomcat Home Directory>\conf\users.xml` file.
   b. Edit the file, to specify the required role, user name, and password.

      The lines in bold text in the following example show the modifications for the Reporting Web Services client role, `EMS Reporting`.

      (Contrary to type conventions in the remainder of this guide, italic text indicates placeholders for user-specified values, *`user-name`* and *`password`*. The angle brackets are a required part of the XML syntax.)

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <!-- create for EMS Reporting web service clients -->
  <role rolename="EMS Reporting"/>

  <!-- create a user assigned to the EMS Reporting role -->
  <user username="user-name" password="password" roles="EMS Reporting"/>
</tomcat-users>
```

3. Update the web service configuration.
   a. Open the web application configuration file:
      `<Tomcat Home Directory>\webapps\ems-rs\WEB-INF\web.xml`
   b. Edit the file, to specify that authentication is required and that the `EMS Reporting` role has access to Reporting services.

      The lines in bold text in the following example show the modifications to grant access to `EMS Reporting` users.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    version="2.5">

    <servlet>
       <servlet-name>RestServlet</servlet-name>
       <servlet-class>
          com.noelios.restlet.ext.servlet.ServerServlet
       </servlet-class>
       <load-on-startup>1</load-on-startup>
    </servlet>

    <!-- The EMS-REPORTING service -->
    <context-param>
       <param-name>org.restlet.application</param-name>
       <param-value>
          com.gvp.rpt.service.ReportingServiceContext
       </param-value>
    </context-param>

    <servlet-mapping>
       <servlet-name>RestServlet</servlet-name>
       <url-pattern>/*</url-pattern>
    </servlet-mapping>

    <!-- indicate that all endpoints within this web application are
authenticated, with access granted to users of the "EMS Reporting" role -->
    <security-constraint>
       <web-resource-collection>
          <web-resource-name>Reporting Services</web-resource-name>
          <url-pattern>/*</url-pattern>
       </web-resource-collection>
       <auth-constraint>
          <role-name>EMS Reporting</role-name>
       </auth-constraint>
    </security-constraint>

    <!-- enable HTTP Basic authorization -->
    <login-config>
       <auth-method>BASIC</auth-method>
       <realm-name>EMS Reporting</realm-name>
    </login-config>

    <!-- list any roles that are part of the above security constraint -->
    <security-role>
       <description>Reporting Services role</description>
       <role-name>EMS Reporting</role-name>
```

```
        </security-role>
    </web-app>
```

4. Restart Apache Tomcat.

   Apache Tomcat will now authenticate all web service requests, including those from the Management Framework Reporting UI (the Genesys Administrator).

5. Update the EMS Reporting user name and password in the Management Framework Reporting UI configuration. The relevant options are in the `rptui` configuration section of the `default Application` object.
   a. In the Genesys Administrator, go to the `Provisioning > Environment > Applications > default > Settings` tab.
   b. From the display filter drop-down list, select `GVP Reporting`.

      The `Settings` tab displays the configuration options for the GVP Reporting UI (`rptui` section).
   c. Modify the `rptui.rs.username` and `rptui.rs.password` options, to specify the user name and password you configured for Tomcat authorization (see Step 2 on page 211).
   d. Click `Save` or `Apply`.
6. Restart the web server, or else log out of the Genesys Administrator and then log back in.
7. Verify that GVP reports display in the Genesys Administrator when you request them from the `Monitoring > Voice Platform` navigation panel..

End of procedure

Connecting to the Genesys Administrator

The following procedure describes the steps to configure the connection between the Genesys Administrator GVP Reporting UI (GVP Reports) and the Reporting Server.

Procedure: Configuring the connection between the Reporting Server and the Genesys Administrator

Purpose: To configure Management Framework to enable GVP reports to be displayed in the Genesys Administrator.

Prerequisites

- You are logged in to the Genesys Administrator. To access the Genesys Administrator, go to the following URL:
  `http://<Genesys Administrator host>/wcm`

Start of procedure

1. In the Genesys Administrator, go to the `Provisioning > Environment > Applications > default > Settings` tab.
2. From the display filter drop-down list, select `GVP Reporting`.

   The `Settings` tab displays the configuration options for the GVP Reports UI (`rptui` section).
3. Verify that the configuration option values are suitable for your deployment.
   - If you have configured Tomcat for HTTP Basic Authorization on the Reporting Server, ensure that you specify the user name and password (`rs.username` and `rs.password`) to enable GVP Reports to provide authentication information to Tomcat. For more information, see Enabling HTTP Basic Authorization for Reporting, page 211.
   - To enable HTTPS communication between the Reporting Server and the Genesys Administrator, set the `enablehttps` option to `true`.

   For more information about the `rptui` configuration options, see Table 40.
4. Click `Save`.
5. On the `Provisioning > Environment > Applications > default > General` tab, add a connection to the Reporting Server. For more information, see the procedure about creating a connection to a server in the *Genesys Voice Platform 8.0 Deployment Guide*.
6. Restart the web server, or else log out of the Genesys Administrator and then log back in.

End of procedure

Table 40 provides information about the options in the `rptui` configuration section of the `default Application` object. Table 40 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 the Genesys Administrator.

Table 40: Reporting UI Configuration Options—default Application

Option Name	Description	Valid Values and Syntax
dsthours	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`
enablehttps	Specifies whether GVP Reports will use HTTP or HTTPS to access Reporting Web Services.	• `true`—HTTPS will be used. • `false`—HTTP will be used. **Default value:** `false`
httptimeout	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)
localtimeformat	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 `tzoffset`).	• `true (1)`—Date and time values will display in local time. • `false`—Date and time values will display in GMT. **Default value:** `true`
rs.httpport	The Tomcat port where Reporting Web Services are available. GVP Reports uses this port to retrieve data from the Reporting Server. The value of this option overrides the server port configured in the Reporting Server `Application`. If this option is not specified in the `default Application`, the port configured in the Reporting Server `Application` is used (default is `8080`).	An integer in the range of 1030–65535. **Default value:** `8080`
rs.password	The password to be provided to the Reporting Server, together with the configured user name (`rs.username`), if the Reporting Server web server has been configured to authenticate user credentials. For more information, see Enabling HTTP Basic Authorization for Reporting, page 211.	An alphanumeric string. **Default value:** Empty

Table 40: Reporting UI Configuration Options—default Application (Continued)

Option Name	Description	Valid Values and Syntax
rs.username	The user name to be provided to the Reporting Server, together with the configured password (`rs.password`), if the Reporting Server web server has been configured to authenticate user credentials. For more information, see Enabling HTTP Basic Authorization for Reporting, page 211.	An alphanumeric string. **Default value:** Empty
tzoffset	The time offset, in hours and minutes, that will be applied to convert GMT to local time (see `localtimeformat`), in the timezone 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`

Part

3 Monitoring GVP

This part of the guide describes the available real-time and historical reports in the Genesys Administrator.

This information appears in the following chapters:

Chapter

11 Reporting Overview

This chapter describes how to use the Genesys Administrator to create real-time and historical reports.

It contains the following sections:

Genesys Administrator

The Genesys Administrator is the 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 8.0 Genesys Administrator Help* file.

Running a Report

The following procedure explains how to run a report using Genesys Administrator.

Procedure: Running a Report

Purpose: To run a report by using the 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 creating reports.
- The name of the 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 9) with the `Monitoring` tab active.

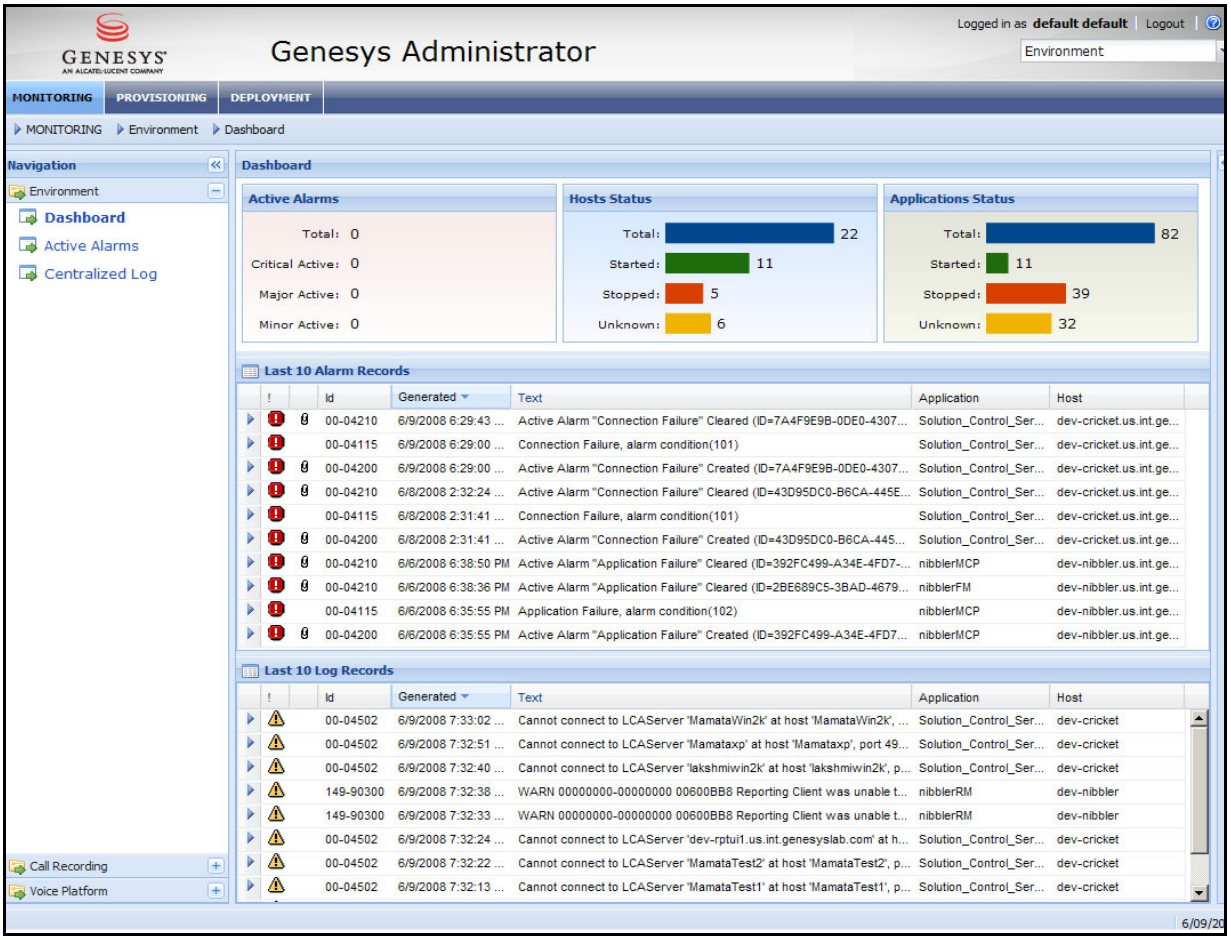


Figure 9: Genesys Administrator

4. From the Navigation panel, select `Voice Platform`.
   The reporting categories become visible (see Figure 10).

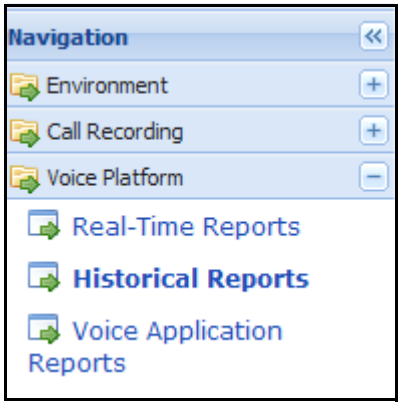


Figure 10: Genesys Administrator Voice Platform Reporting Categories

5. Select the required reporting category.

   The list of available reports is visible in the right-hand panel of Genesys Administrator (for example, see Figure 11, the Historical Report List).

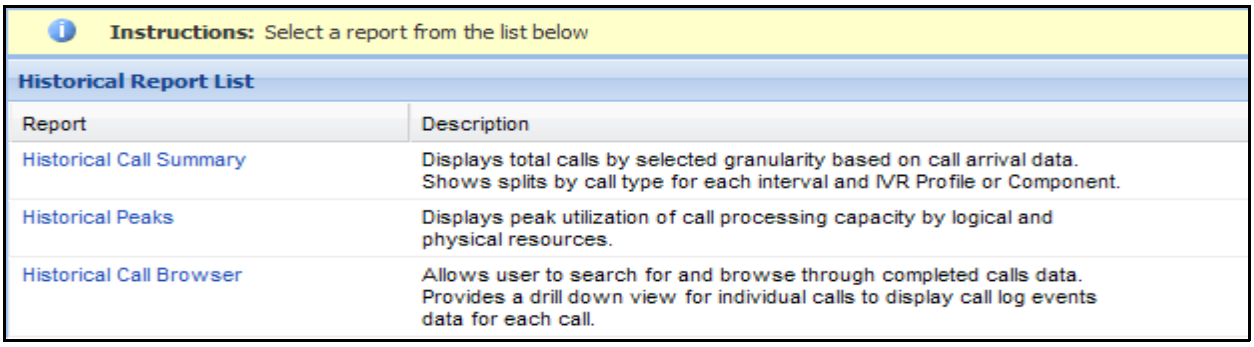


Figure 11: Historical Report List

6. Select the required report from the list of available reports.

   The `Configure Filters` screen appears with the possible filter criteria (see Figure 12). A different set of filters appear for each report type.

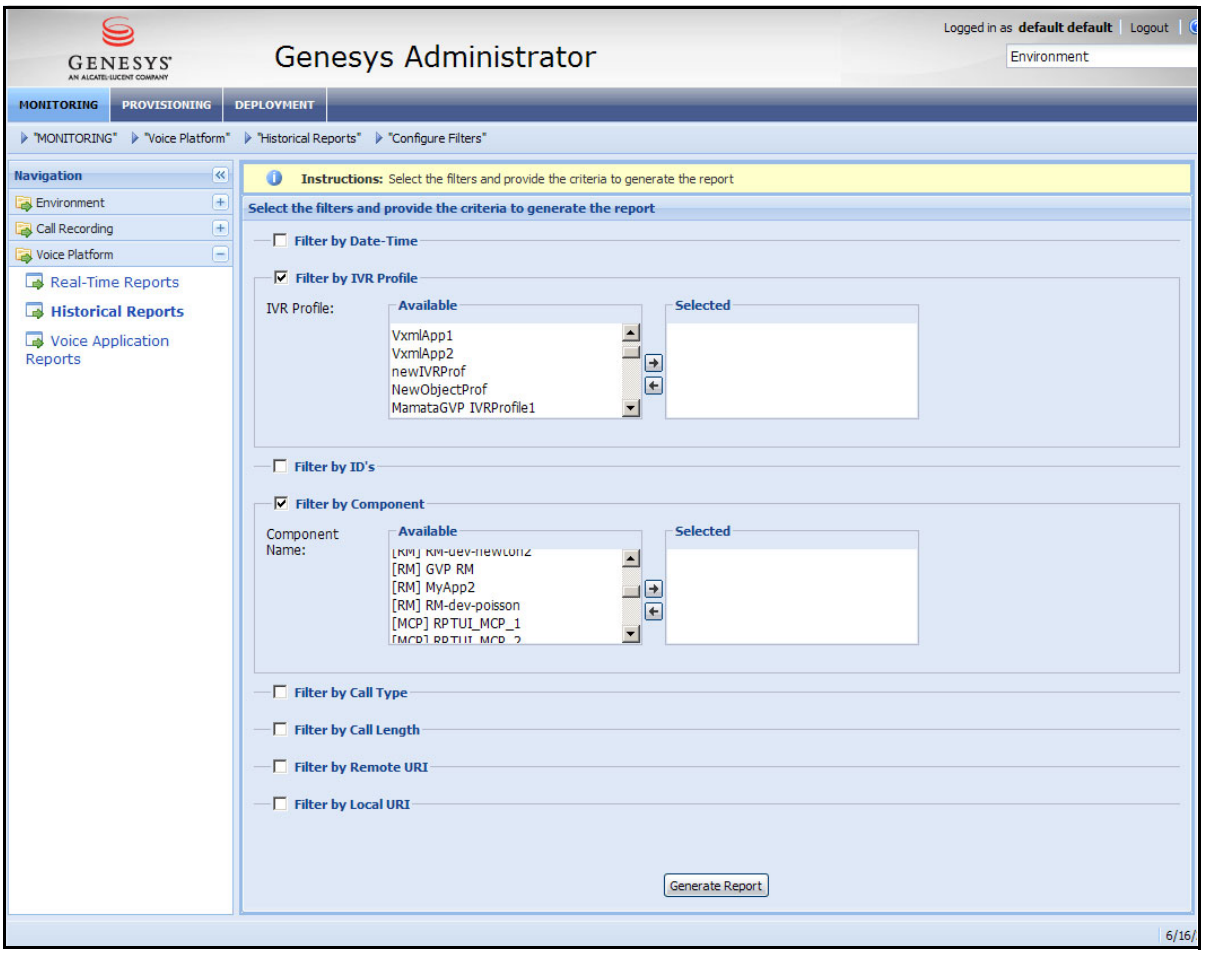


Figure 12: Configure Filters Screen for the Historical Call Browser Report

7. Select the required filters. Table 41, “Reporting Filters,” on page 224 describes each filter.
   - Table 42 on page 228 lists the real-time reporting supported filters.
   - Table 45 on page 234 lists the historical reporting supported filters, and the valid filter combinations for each report.
   - Table 50 on page 244 lists the VAR reporting supported filters, and the valid filter combinations for each report.
8. Click `Generate Report`.

   The `Report Results` screen appears (see Figure 13).

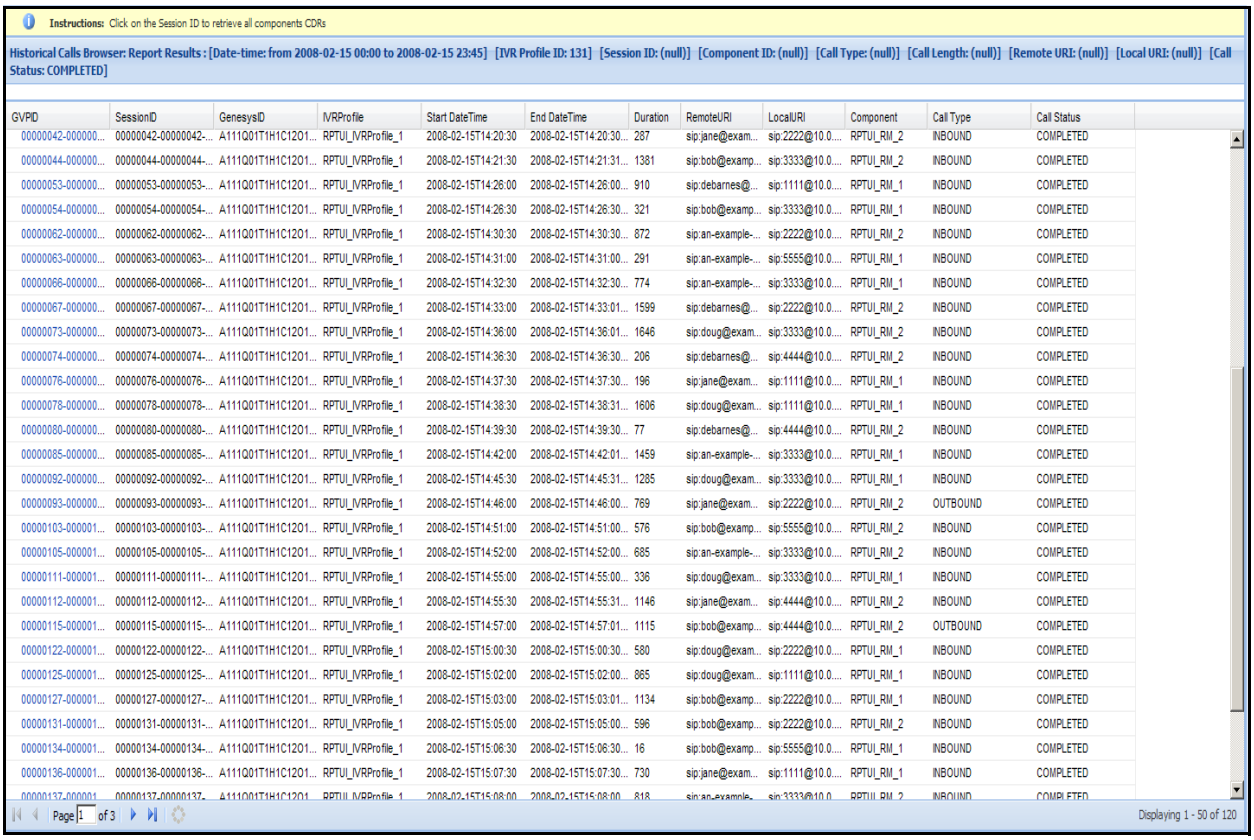


Figure 13: Example of Historical Call Browser Report Results

9. If the report has drill down functionality, select the required record to view the details of that record (see Figure 14).

Compone...	GVP ID	Genesys ID	Session ID	IVRProfile	Compon
CCP	00000116-00000116-00000116-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000065-00000065-00000065-0000...	RPTUI_IVR...	RPTUI_CC
MCP	00000116-00000116-00000116-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000064-00000064	RPTUI_IVR...	RPTUI_MC

Figure 14: Example of Historical Call Browser Drill-Down Results

End of procedure

Report Filters

Table 41 describes the filter criteria that you can use to retrieve call detail records, IVR action data, or summary data for the Real-Time, Historical, and Voice Application reports.

Table 41: Reporting Filters

Filter Name	Description
`Filter by Date-Time`	Filters the data by start date, end date, start time, and end time. The results will display calls that started on or after the start time and ended before the end time.
`Filter by IVR Profile`	Filters the data by IVR Profile. You can choose more than one IVR Profile for some of the reports. For more information, see the individual reports. For more information on IVR Profiles, see Chapter 6 on page 139.
`Filter by Call End State`	Filters the data by Call End States. The possible Call End States are: • `Application End`—The voice 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.
`Filter by Call Result`	Filters the data 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 call was rejected by MCP. • `Unknown`—Some unknown reason caused the call to end abruptly.

Table 41: Reporting Filters (Continued)

Filter Name	Description
Filter by ID's	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 ID—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.
Filter by Component	Filters the data by the component. A component is a provisioned Resource Manager, MCP, or CCP application. You can choose more than one component for some of the reports. For more information, see the individual reports. **Note:** All selected components must be of the same type—RM, CCP, or MCP.
Filter by Granularity Level	Presents the data at various levels of aggregation: • Five Minutes • Hour • Day • Week • Month
Filter by 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. • NewCall—Applicable for CCP components only. • Create-CCXML—Applicable for CCP components only. • External—Applicable for CCP components only. • Unknown—Applicable for RM components only.

Table 41: Reporting Filters (Continued)

Filter Name	Description
`Filter by 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.
`Filter by Remote URI`	Filters the data by the full URI of the remote party that is involved in the session. **Note:** Accepts the `*` wildcard.
`Filter by Local URI`	Filters the data by the URI of the local service. **Note:** Accepts the `*` wildcard.

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`
- `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 204.

Chapter

12 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, change 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 example, more transaction overhead on the server side), but the result is more *real-time*.

The following procedure describes how to run a real-time report.

Procedure: Running a Real-Time Report

Purpose: To run a real-time report by using the Genesys Administrator.

Start of procedure:

- Follow the instructions to create a report (see Running a Report, page 219).
- For Step 7 in those instructions, see Table 42 for the available real-time report filters.

End of procedure

Table 42: Real-Time Report Filters

Filter	Available Real-Time Reports
	Active Call List
Filter by IVR Profile	✓
Filter by Component	✓

Active Call List

The Active Call List report (see Figure 15) displays the list of calls that currently are being processed by GVP. It includes also 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, of 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.

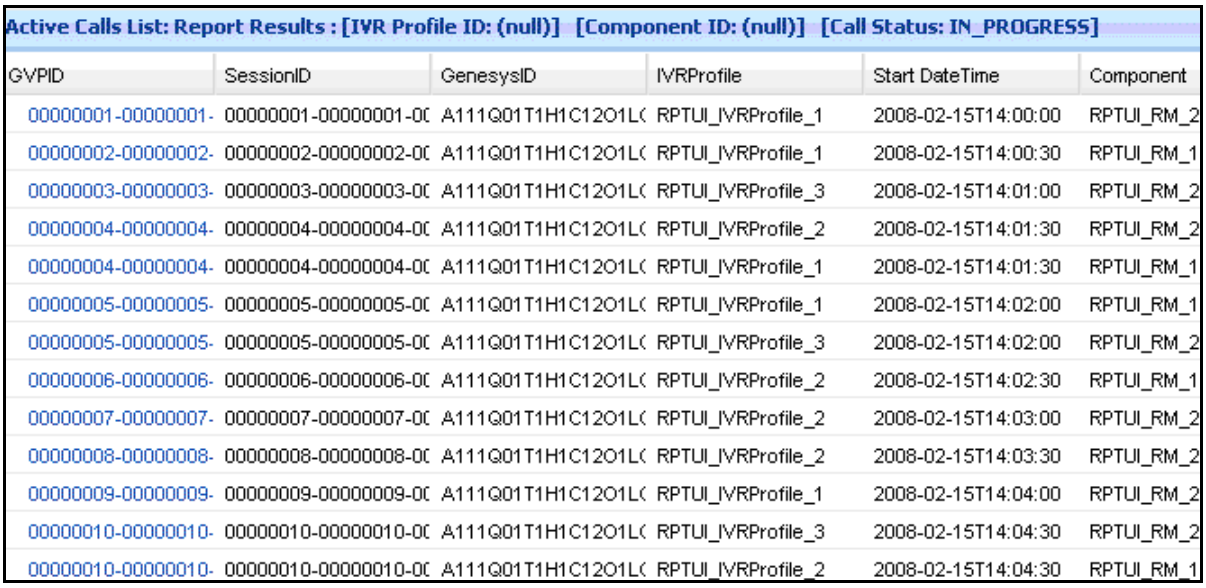
The Reporting Server processes timed out calls once hourly (by default), and marks calls as timed out when they have been in progress for more than a configured period of time (by default 3 hours).

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.

You can select multiple IVR Profiles and multiple Components.

For more information on the details of completed calls, see "Historical Call Browser" on page 238.



Active Calls List: Report Results : [IVR Profile ID: (null)] [Component ID: (null)] [Call Status: IN_PROGRESS]

GVPID	SessionID	GenesysID	IVRProfile	Start DateTime	Component
00000001-00000001-	00000001-00000001-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_1	2008-02-15T14:00:00	RPTUI_RM_2
00000002-00000002-	00000002-00000002-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_1	2008-02-15T14:00:30	RPTUI_RM_1
00000003-00000003-	00000003-00000003-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_3	2008-02-15T14:01:00	RPTUI_RM_2
00000004-00000004-	00000004-00000004-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_2	2008-02-15T14:01:30	RPTUI_RM_2
00000004-00000004-	00000004-00000004-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_1	2008-02-15T14:01:30	RPTUI_RM_1
00000005-00000005-	00000005-00000005-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_1	2008-02-15T14:02:00	RPTUI_RM_1
00000005-00000005-	00000005-00000005-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_3	2008-02-15T14:02:00	RPTUI_RM_2
00000006-00000006-	00000006-00000006-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_2	2008-02-15T14:02:30	RPTUI_RM_1
00000007-00000007-	00000007-00000007-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_2	2008-02-15T14:03:00	RPTUI_RM_2
00000008-00000008-	00000008-00000008-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_2	2008-02-15T14:03:30	RPTUI_RM_2
00000009-00000009-	00000009-00000009-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_1	2008-02-15T14:04:00	RPTUI_RM_2
00000010-00000010-	00000010-00000010-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_3	2008-02-15T14:04:30	RPTUI_RM_2
00000010-00000010-	00000010-00000010-0(	A111Q01T1H1C12O1L(	RPTUI_IVRProfile_2	2008-02-15T14:04:30	RPTUI_RM_1

Figure 15: Active Call List Report

Table 43 describes the fields for the summary level for the Active Call List report.

Table 43: Active Call List Report Summary Fields

Field	Description
GVPID	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 72.
SessionID	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 72.

Table 43: Active Call List Report Summary Fields (Continued)

Field	Description
GenesysID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see “Session Identifiers” on page 72.
IVRProfile	The name of the IVR Profile that is selected.
Start DateTime	The start date and start time of the call.
Component	The name of the Resource Manager that is selected.
Call Type	The type of the call. Valid call types are the following: • INBOUND • OUTBOUND • BRIDGED • UNKNOWN • NEW CALL • CREATE-CCXML • EXTERNAL
RemoteURI	The remote Uniform Resource Identifier.
LocalURI	The local Uniform Resource Identifier.

The drill-down report breaks down the detail recorded of the selected call according to component type (see Figure 16).

Compone...	GVP ID	Genesys ID	Session ID	Appl
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 16: Active Call List Drill Down Report

Table 44 describes the fields for the Active Call List drill down report.

Table 44: Active Call List Drill Down 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)
GVPID	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 72.
GenesysID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see "Session Identifiers" on page 72.
SessionID	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 72.
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 DateTime	The start date and start time of the call.
End DateTime	The end date and end time of the call if the call has completed.

Table 44: Active Call List Drill Down Report Fields (Continued)

Fields	Description
Call Status	The current state of the call. Valid call states are the following: • IN-PROGRESS • TIMED OUT • COMPLETED
Call Type	The type of call. Valid call types are the following: • INBOUND • OUTBOUND • BRIDGED • UNKNOWN • NEW CALL • CREATE-CCXML • EXTERNAL

Chapter

13 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, based on a set of selected criteria.

The Historical Call Summary and Historical Peaks reports display the data in both a pictorial graph and a table. The graph provide 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.

The following procedure describes how to run a historical report.

Procedure: Running a Historical Report

Purpose: To run a historical report by using the Genesys Administrator.

Start of procedure:

- Follow the instructions to run a report (see Running a Report, page 219).
- For Step 7 in those instructions, see Table 45 for the available historical filters and the valid filter combinations.

End of procedure

Table 45: Historical Filters and Valid Filter Combinations

Filter	Available Historical Reports		
	Call Summary	Peaks	Call Browser
Filter by Date-Time	✓	✓	✓
Filter by IVR Profile	✓	✓	✓
Filter by ID's			✓
Filter by Component	✓	✓	✓
Filter by Granularity Level	✓	✓	
Filter by Call Type			✓
Filter by Call Length			✓
Filter by Remote URI			✓
Filter by Local URI			✓
Filter Combinations	• Date-Time, IVR Profile, Granularity • Date-Time, Component, Granularity	• Date-Time, IVR Profile, Granularity • Date-Time, Component, Granularity	• None **Note:** If you filter by the Session ID, you also need to provide a Component.

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. The retention policy can be overridden on the IVR Application by specifying the corresponding configuration in the IVR Profile (minus the `.default` suffix):

- `rs.db.retention.operations.5min.default`
- `rs.db.retention.operations.hourly.default`
- `rs.db.retention.operations.daily.default`

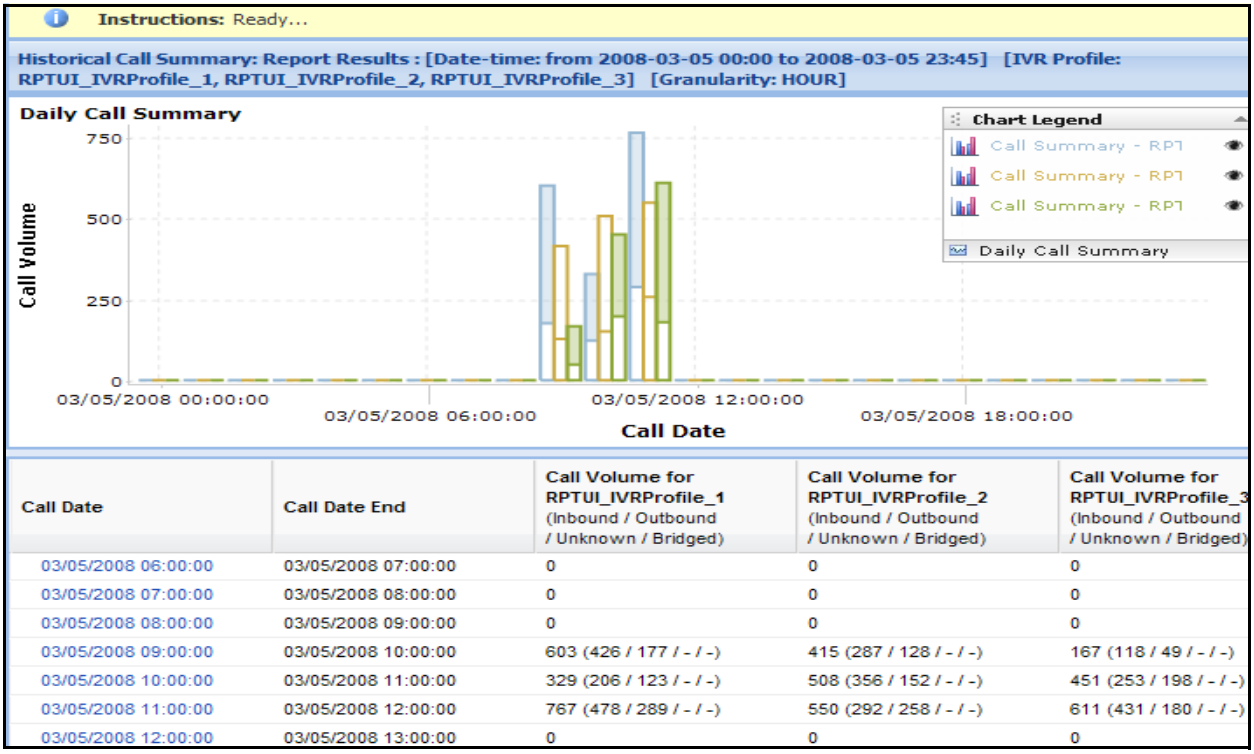
- `rs.db.retention.operations.weekly.default`
- `rs.db.retention.operations.monthly.default`

For more information on this and other data retention parameters, see “Configuring Database Retention Policies” on page 206.

Historical Call Summary

The Historical Call Summary report (see Figure 17) lists a summary of call arrival data that is submitted by each component for a given period of time, IVR Profile, and/or Component.

Note: You can select multiple IVR Profiles or multiple Components.



Call Date	Call Date End	Call Volume for RPTUI_IVRProfile_1 (Inbound / Outbound / Unknown / Bridged)	Call Volume for RPTUI_IVRProfile_2 (Inbound / Outbound / Unknown / Bridged)	Call Volume for RPTUI_IVRProfile_3 (Inbound / Outbound / Unknown / Bridged)
03/05/2008 06:00:00	03/05/2008 07:00:00	0	0	0
03/05/2008 07:00:00	03/05/2008 08:00:00	0	0	0
03/05/2008 08:00:00	03/05/2008 09:00:00	0	0	0
03/05/2008 09:00:00	03/05/2008 10:00:00	603 (426 / 177 / - / -)	415 (287 / 128 / - / -)	167 (118 / 49 / - / -)
03/05/2008 10:00:00	03/05/2008 11:00:00	329 (206 / 123 / - / -)	508 (356 / 152 / - / -)	451 (253 / 198 / - / -)
03/05/2008 11:00:00	03/05/2008 12:00:00	767 (478 / 289 / - / -)	550 (292 / 258 / - / -)	611 (431 / 180 / - / -)
03/05/2008 12:00:00	03/05/2008 13:00:00	0	0	0

Figure 17: Historical Call Summary Report

Table 46 describes the fields for the Historical Call Summary report.

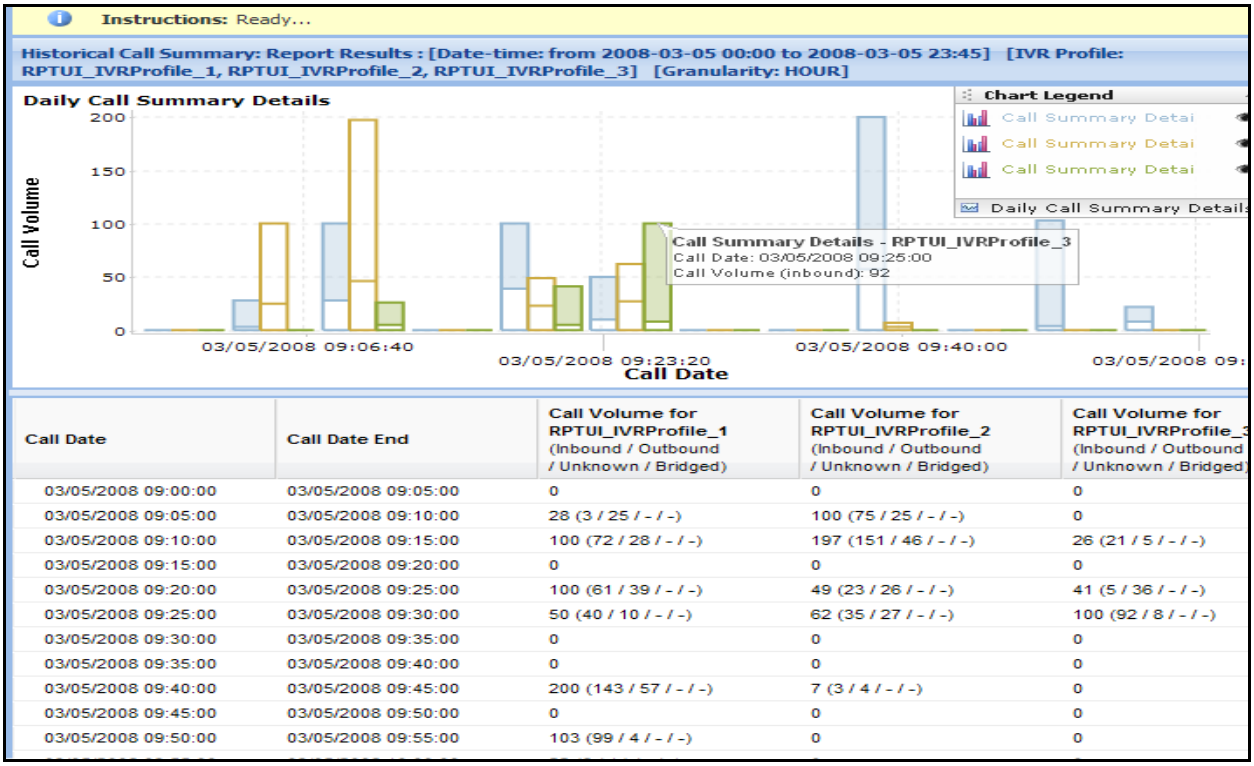
Table 46: Historical Call Summary Report Fields

Field	Description
`Call Date`	The start date and start time for the calls.

Table 46: Historical Call Summary Report Fields (Continued)

Field	Description
Call Date End	The end date and end time for the calls.
Call Volume (Inbound/Outbound /Unknown/Bridged)	The number of calls according to call type for the respective call date.

The Daily Call Summary Details report (see Figure 18) displays the call distribution details of the selected day in five-minute intervals.



Call Date	Call Date End	Call Volume for RPTUI_IVRProfile_1 (Inbound / Outbound / Unknown / Bridged)	Call Volume for RPTUI_IVRProfile_2 (Inbound / Outbound / Unknown / Bridged)	Call Volume for RPTUI_IVRProfile_3 (Inbound / Outbound / Unknown / Bridged)
03/05/2008 09:00:00	03/05/2008 09:05:00	0	0	0
03/05/2008 09:05:00	03/05/2008 09:10:00	28 (3 / 25 / - / -)	100 (75 / 25 / - / -)	0
03/05/2008 09:10:00	03/05/2008 09:15:00	100 (72 / 28 / - / -)	197 (151 / 46 / - / -)	26 (21 / 5 / - / -)
03/05/2008 09:15:00	03/05/2008 09:20:00	0	0	0
03/05/2008 09:20:00	03/05/2008 09:25:00	100 (61 / 39 / - / -)	49 (23 / 26 / - / -)	41 (5 / 36 / - / -)
03/05/2008 09:25:00	03/05/2008 09:30:00	50 (40 / 10 / - / -)	62 (35 / 27 / - / -)	100 (92 / 8 / - / -)
03/05/2008 09:30:00	03/05/2008 09:35:00	0	0	0
03/05/2008 09:35:00	03/05/2008 09:40:00	0	0	0
03/05/2008 09:40:00	03/05/2008 09:45:00	200 (143 / 57 / - / -)	7 (3 / 4 / - / -)	0
03/05/2008 09:45:00	03/05/2008 09:50:00	0	0	0
03/05/2008 09:50:00	03/05/2008 09:55:00	103 (99 / 4 / - / -)	0	0

Figure 18: Historical Call Summary Details Report

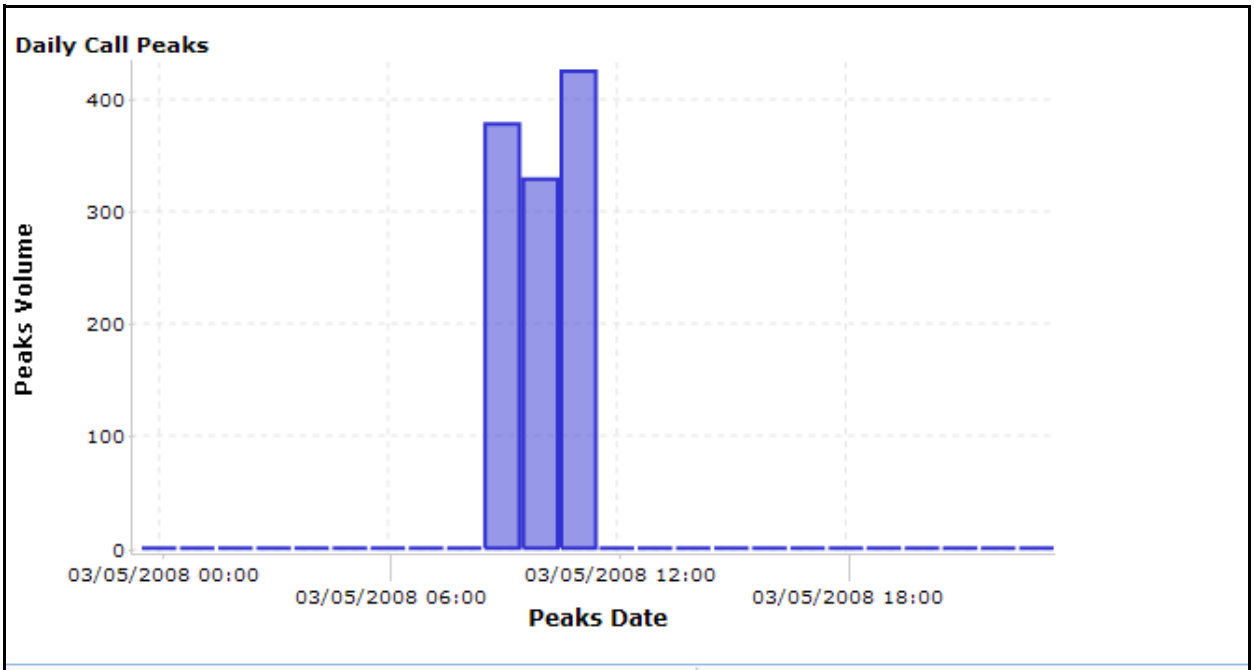
The Historical Call Summary report provides a summary row at the bottom of the grid that displays the grand totals for each column.

Historical Peaks

The Historical Peaks report (see Figure 19) provides the peak volume of calls that is observed during a given period of time.

Note: You can select multiple Components; however, you can select only one IVR Profile.

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.



Peaks Date	Peaks Volume
03/05/2008 00:00:00	0
03/05/2008 01:00:00	0
03/05/2008 02:00:00	0
03/05/2008 03:00:00	0
03/05/2008 04:00:00	0
03/05/2008 05:00:00	0
03/05/2008 06:00:00	0
03/05/2008 07:00:00	0
03/05/2008 08:00:00	0
03/05/2008 09:00:00	377
03/05/2008 10:00:00	328

Figure 19: Historical Peaks Report

Table 47 describes the fields for the Historical Peaks report.

Table 47: Historical Peaks Reports Fields

Field	Description
`Peaks Date`	The date and time of the call for the given granularity period.
`Peaks Volume`	The number of calls for the given time period.

Historical Call Browser

The Historical Call Browser report (see Figure 20) 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, of 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).

Notes: You can select multiple IVR Profiles and multiple Components.

Data is returned if no filter is selected.

GVPID	SessionID	GenesysID	IVRProfile	Start Date...	End Date...	Duration	RemoteURI	LocalURI	Component	Call T
00000001...	0000000...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1159	sip:doug@exam...	sip:1111@10.0....	RPTUI_RM_2	INBO
00000002...	0000000...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1355	sip:an-example-...	sip:1111@10.0....	RPTUI_RM_1	INBO
00000004...	0000000...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	437	sip:debarnes@...	sip:2222@10.0....	RPTUI_RM_1	INBO
00000005...	0000000...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1211	sip:jane@exam...	sip:1111@10.0....	RPTUI_RM_1	OUT
00000009...	0000000...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1714	sip:debarnes@...	sip:2222@10.0....	RPTUI_RM_2	INBO
00000011...	0000001...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1581	sip:debarnes@...	sip:2222@10.0....	RPTUI_RM_2	INBO
00000013...	0000001...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1729	sip:an-example-...	sip:1111@10.0....	RPTUI_RM_2	INBO
00000015...	0000001...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1059	sip:an-example-...	sip:5555@10.0....	RPTUI_RM_1	INBO
00000017...	0000001...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1117	sip:debarnes@...	sip:2222@10.0....	RPTUI_RM_1	INBO
00000018...	0000001...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	273	sip:bob@examp...	sip:1111@10.0....	RPTUI_RM_1	INBO
00000021...	0000002...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1398	sip:jane@exam...	sip:4444@10.0....	RPTUI_RM_1	INBO
00000024...	0000002...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	660	sip:debarnes@...	sip:5555@10.0....	RPTUI_RM_1	INBO
00000025...	0000002...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1630	sip:doug@exam...	sip:4444@10.0....	RPTUI_RM_1	INBO
00000027...	0000002...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1126	sip:an-example-...	sip:2222@10.0....	RPTUI_RM_1	INBO
00000032...	0000003...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	820	sip:bob@examp...	sip:2222@10.0....	RPTUI_RM_2	INBO
00000034...	0000003...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1406	sip:bob@examp...	sip:3333@10.0....	RPTUI_RM_1	INBO
00000041...	0000004...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	549	sip:debarnes@...	sip:2222@10.0....	RPTUI_RM_1	OUT
00000042...	0000004...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	287	sip:jane@exam...	sip:2222@10.0....	RPTUI_RM_2	INBO
00000044...	0000004...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	1381	sip:bob@examp...	sip:3333@10.0....	RPTUI_RM_2	INBO
00000053...	0000005...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	910	sip:debarnes@...	sip:1111@10.0....	RPTUI_RM_1	INBO
00000054...	0000005...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	321	sip:bob@examp...	sip:3333@10.0....	RPTUI_RM_1	INBO
00000062...	0000006...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	872	sip:an-example-...	sip:2222@10.0....	RPTUI_RM_2	INBO
00000063...	0000006...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	291	sip:an-example-...	sip:5555@10.0....	RPTUI_RM_1	INBO
00000066...	0000006...	A111Q01T1H...	RPTUI_IVRProfile...	2008-02-...	2008-02-...	774	sip:an-example-...	sip:3333@10.0....	RPTUI_RM_1	INBO

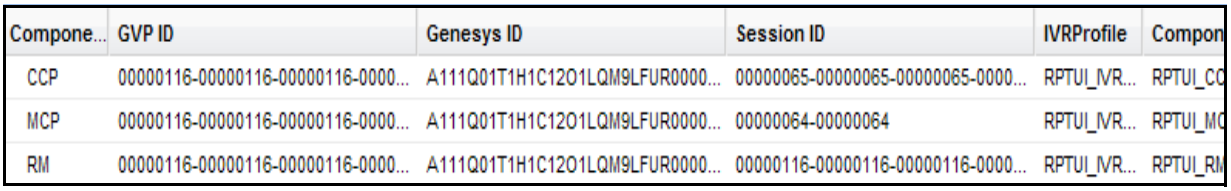
Figure 20: Historical Call Browser Report

Table 48 describes the fields for the Historical Call Browser report.

Table 48: Historical Call Browser Report Fields

Field	Description
GVPID	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 72.
SessionID	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 72.
GenesysID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see "Session Identifiers" on page 72.
IVRProfile	The name of the IVR Profile that is selected.
Start DateTime	The start date and start time of the call.
End DateTime	The end date and end time of the call.
Duration	The length of the time of the call in milliseconds.
RemoteURI	The remote Uniform Resource Identifier.
LocalURI	The local Uniform Resource Identifier.
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 • CREATE-CCXML • EXTERNAL
Call Status	The state of the call. Valid call states are the following: • COMPLETED • TIMED OUT

The drill-down report 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 (see Figure 21). There can be multiple call detail records for each component if there was more than one leg in the call.



Compone...	GVP ID	Genesys ID	Session ID	IVRProfile	Compon
CCP	00000116-00000116-00000116-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000065-00000065-00000065-0000...	RPTUI_IVR...	RPTUI_CC
MCP	00000116-00000116-00000116-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000064-00000064	RPTUI_IVR...	RPTUI_MC
RM	00000116-00000116-00000116-0000...	A111Q01T1H1C12O1LQM9LFUR0000...	00000116-00000116-00000116-0000...	RPTUI_IVR...	RPTUI_RM

Figure 21: Historical Call Browser Drill-Down Report

Table 49 describes the fields for the Historical Call Browser Drill-Down report.

Table 49: Historical Call Browser Report Drill-Down

Field	Description
Component Type	The type of component. Valid types are: • CCP (Call Control Platform). • MCP (Media Control Platform). • RM (Resource Manager).
GVPID	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 72.
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 72.
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 72.
Application ID	The name of the IVR Profile as seen in Genesys Administrator or Configuration Manager.
Component ID	The name of the Component application as seen in Genesys Administrator or Configuration Manager.
Start DateTime	The start date and start time of the call.
End DateTime	The end date and start time of the call.

Table 49: Historical Call Browser Report Drill-Down (Continued)

Field	Description
Call Status	The state of the call. Valid call states are the following: • COMPLETED • TIMED OUT
Call Type	The type of call. Valid call types are the following: • INBOUND • OUTBOUND • BRIDGED • UNKNOWN • NEW CALL • CREATE-CCXML • EXTERNAL

Chapter

14 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.

The following procedure describes how to run a VAR report.

Procedure: Running a Voice Application Report

Purpose: To run a Voice Application Report by using the Genesys Administrator.

Start of procedure:

- Follow the instructions to create a report (see Running a Report, page 219).
- For Step 7 in those instructions, see Table 50 for the available Voice Application Report filters and the valid filter combinations.

End of procedure

Table 50: Voice Application Filters and Valid Filter Combinations

Filter	Available Voice Application Reports			
	Call Browser	Call Completion Summary	IVR Action Usage	Last IVR Action Used
Filter by Date-Time	✓	✓	✓	✓
Filter by IVR Profile	✓	✓	✓	✓
Filter by Call End State	✓			
Filter by Call Result	✓			
Filter by ID's	✓			
Filter by Component	✓			
Filter by Granularity Level		✓	✓	✓
Filter Combinations	none	• DateTime, IVRProfile, Granularity **Note:** You can select only one IVRProfile at a time.	• DateTime, IVRProfile, Granularity	• DateTime, IVRProfile, Granularity **Note:** You can select only one IVRProfile at a time.

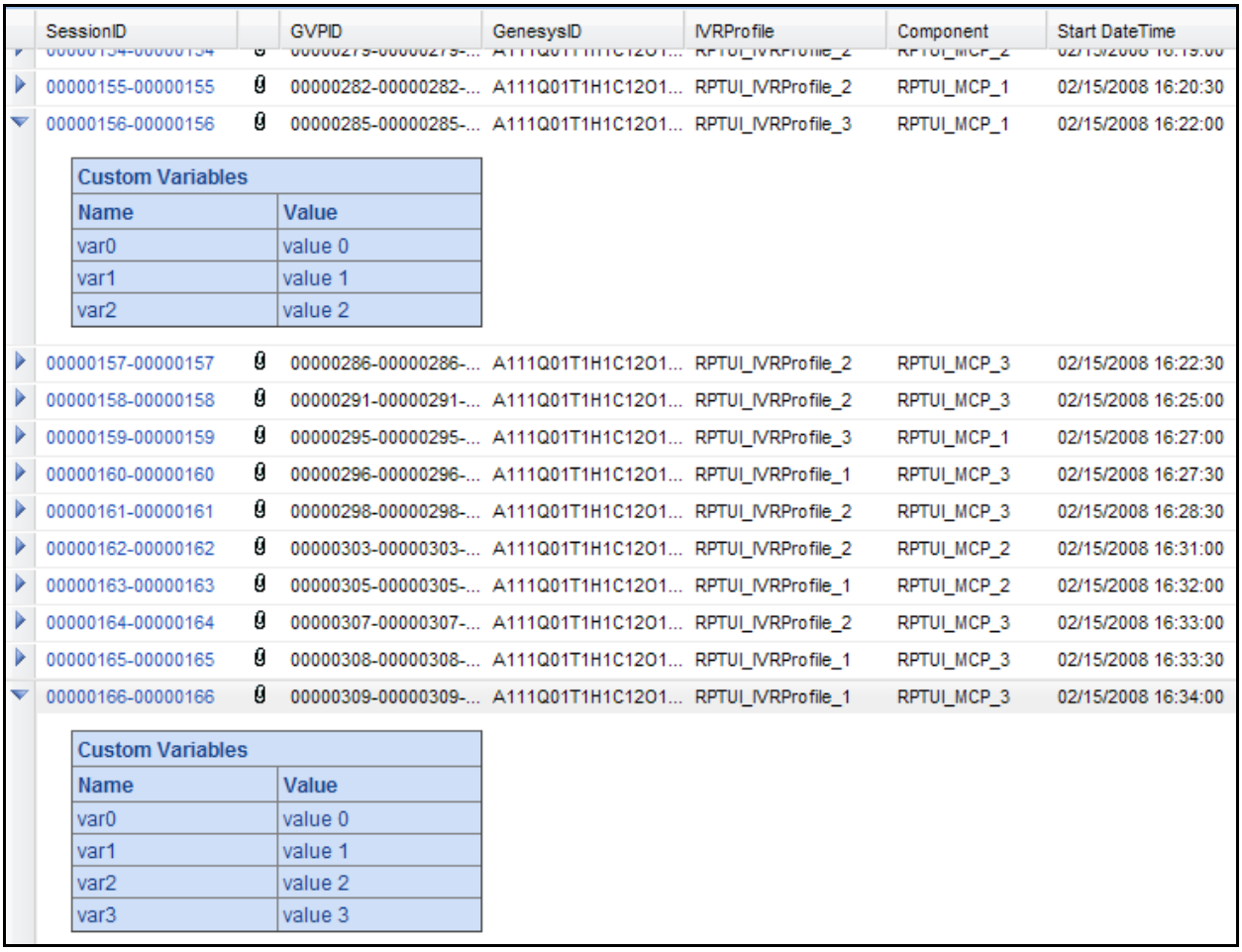
VAR Call Browser

The VAR Call Browser report (see Figure 22) provides the ability to search and browse call data that relates to VAR reporting. It displays a list of all calls that occurred for a selected period of time.

Notes: The VAR Call Browser report displays MCP data only.

Data is returned if no filter is selected.

Any call detail record that contains custom variables will have a paper clip icon in the second column.



SessionID	GVPID	GenesysID	IVRProfile	Component	Start DateTime
00000155-00000155	00000282-00000282-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_2	RPTUI_MCP_1	02/15/2008 16:20:30
00000156-00000156	00000285-00000285-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_3	RPTUI_MCP_1	02/15/2008 16:22:00

Custom Variables	
Name	Value
var0	value 0
var1	value 1
var2	value 2

00000157-00000157	00000286-00000286-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_2	RPTUI_MCP_3	02/15/2008 16:22:30
00000158-00000158	00000291-00000291-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_2	RPTUI_MCP_3	02/15/2008 16:25:00
00000159-00000159	00000295-00000295-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_3	RPTUI_MCP_1	02/15/2008 16:27:00
00000160-00000160	00000296-00000296-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_1	RPTUI_MCP_3	02/15/2008 16:27:30
00000161-00000161	00000298-00000298-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_2	RPTUI_MCP_3	02/15/2008 16:28:30
00000162-00000162	00000303-00000303-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_2	RPTUI_MCP_2	02/15/2008 16:31:00
00000163-00000163	00000305-00000305-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_1	RPTUI_MCP_2	02/15/2008 16:32:00
00000164-00000164	00000307-00000307-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_2	RPTUI_MCP_3	02/15/2008 16:33:00
00000165-00000165	00000308-00000308-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_1	RPTUI_MCP_3	02/15/2008 16:33:30
00000166-00000166	00000309-00000309-...	A111Q01T1H1C12O1...	RPTUI_IVRProfile_1	RPTUI_MCP_3	02/15/2008 16:34:00

Custom Variables	
Name	Value
var0	value 0
var1	value 1
var2	value 2
var3	value 3

Figure 22: VAR Call Browser Report

Table 51 describes the fields for the VAR Call Browser report.

Table 51: VAR Call Browser Report Fields

Field	Description
SessionID	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 72.
GVPID	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 72.

Table 51: VAR Call Browser Report Fields (Continued)

Field	Description
GenesysID	The Genesys CallUUID that is generated by T-Server or SIP Server. For more information on the Genesys CallUUID, see "Session Identifiers" on page 72.
IVRProfile	The name of the IVR Profile that is selected.
Component	The name of the Resource Manager application that is selected.
Start DateTime	The start date and start time of the call.
End DateTime	The end date and end time of the call.
Duration	The length of the time of the call in milliseconds.
RemoteURI	The remote Uniform Resource Identifier.
LocalURI	The local Uniform Resource Identifier.
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.
Result	The end result of the call, as reported by the application. Valid results are: • SUCCESS—The call was processed successfully. • FAILED—A failure occurred that prevented the call from being processed properly—for example, a database error or network error. • REJECTED—The MCP rejected the call—for example, the VoiceXML application could not be fetched from the application server. • UNKNOWN—Some unknown reason caused the call to end abruptly.
Reason	A string identifier, of no more than 256 characters in length, that explains the results.

Table 51: VAR Call Browser Report Fields (Continued)

Field	Description
Notes	A string identifier, of no more than 256 characters in length, that has additional information that relates to the call.
Custom Variables	The name and value of any custom variable that associates name/value pairs to the call.

Call Completion Summary

The Call Completion Summary report (see Figure 23) displays the total number of calls and the percentage of calls, grouped by Call End Action data. The Call End Action displays the IVR Results and the IVR Result Reasons.

Note: The Call Completion Summary report displays MCP data only.

Call Completion Time	Call End State	Call End Re...	Reason	Total Calls	% of Calls	Avg. Call Len...
2008-03-05T17:0...				37472		
	USER_END			6438	17.18%	60.0
		SUCCESS		4788	74.37%	
		FAILED		1188	18.45%	
		REJECTED		231	3.59%	
		UNKNOWN		231	3.59%	
	APPLICATION_END			15867	42.34%	60.0
		SUCCESS		11770	74.18%	
		FAILED		2935	18.5%	
		REJECTED		581	3.66%	
		UNKNOWN		581	3.66%	
	SYSTEM_ERROR			6459	17.24%	60.0
		SUCCESS		4797	74.27%	
		FAILED		1194	18.49%	
		REJECTED		234	3.62%	
		UNKNOWN		234	3.62%	
	UNKNOWN			8708	23.24%	60.0
		SUCCESS		6467	74.27%	
		FAILED		1611	18.5%	
		REJECTED		315	3.62%	

Figure 23: Call Completion Summary Report

Table 52 describes the fields for the Call Completion Summary report.

Table 52: 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.
Call 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.
Call End Result	The end result of the call, as reported by the application. Valid results are: • SUCCESS—The call was processed successfully. • FAILED—A failure occurred that prevented the call from being processed properly—for example, a database error or network error. • REJECTED—The MCP rejected the call—for example, the VoiceXML application could not be fetched from the application server. • UNKNOWN—Some unknown reason caused the call to end abruptly.
Reason	A string identifier, of no more than 256 characters in length, that explains the result.
Total Calls	The total number of calls that ended for the time duration (granularity) that is selected.
% of Calls	The percentage of calls for each level of granularity with respect to the next higher grouping.
Avg. Call Length	The average length, in milliseconds, of the time of the call.

IVR Action Usage

The IVR Action Usage report (see Figure 24) displays statistics on individual IVR Actions that are used for a given IVR Profile within a given time period.

Note: The IVR Action Usage report displays MCP data only.

IVRProfile	IVR Action	Usage Count	% actions that w...	Calls that use...	% age of ...
RPTUI_IVRProfile_1					
	Random Action #1	4049	33.27%	4049	4.33%
	Random Action #10	3821	34.13%	3821	4.08%
	Random Action #2	3785	34.58%	3785	4.05%
	Random Action #3	3511	32.95%	3511	3.75%
	Random Action #4	4151	33.41%	4151	4.44%
	Random Action #5	3702	32.96%	3702	3.96%
	Random Action #6	4143	33.02%	4143	4.43%
	Random Action #7	3442	31.26%	3442	3.68%
	Random Action #8	3941	33.01%	3941	4.21%
	Random Action #9	3363	32.74%	3363	3.59%

Figure 24: IVR Action Usage Report

Table 53 describes the fields of the IVR Action Usage report.

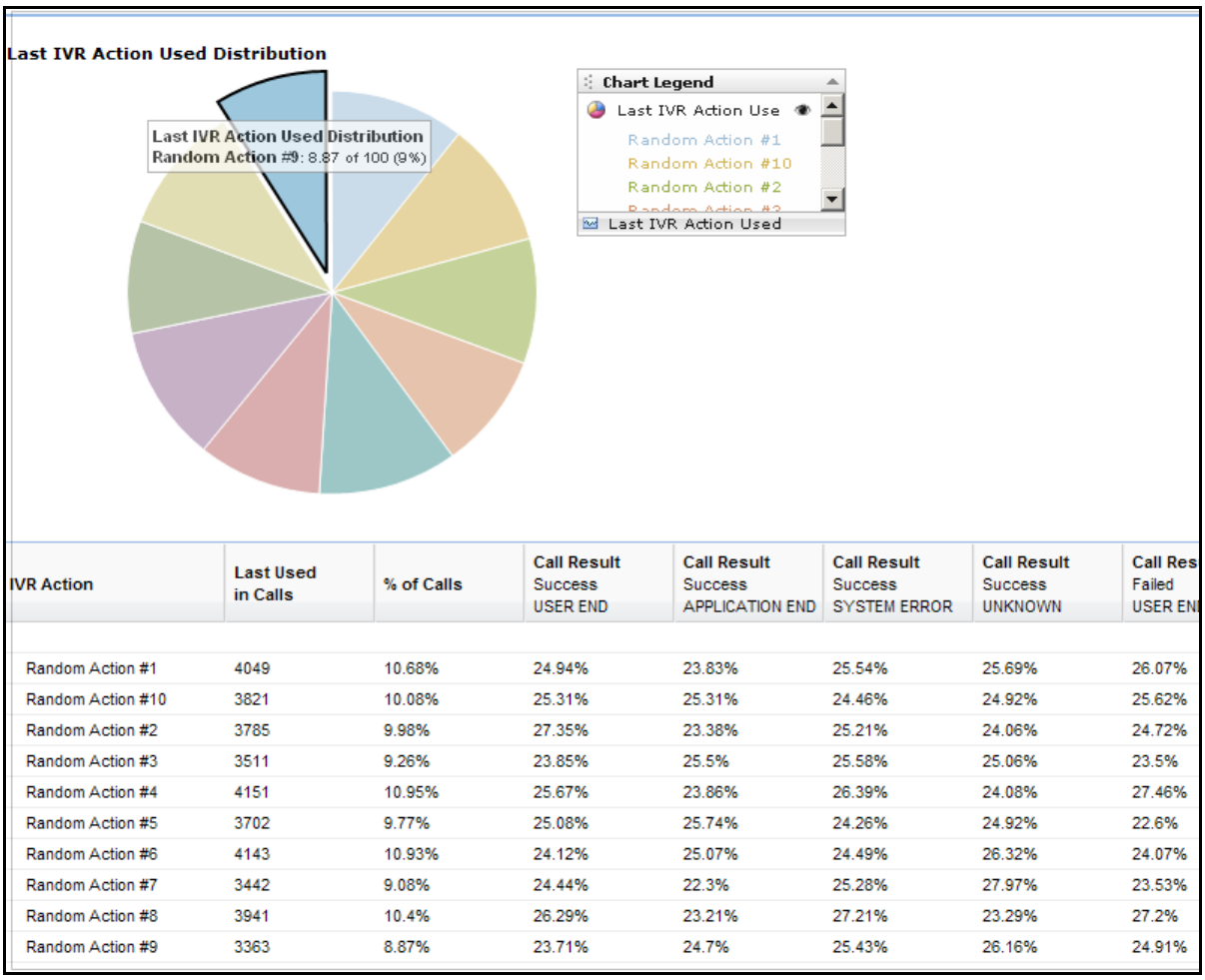
Table 53: IVR Action Usage Report Fields

Field	Description
IVRProfile	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 that were successful	The percentage of actions that were successful.
Calls that used this action	The number of calls that used this IVR Action at least once.
% age of Calls that used this action	The percentage of total calls that used this IVR Action at least once.

Last IVR Action Used

The Last IVR Action Used report (see Figure 25) displays the details of the last IVR Actions that were used during the end of a call.

Note: The Last IVR Action Used report displays MCP data only.



IVR Action	Last Used in Calls	% of Calls	Call Result Success USER END	Call Result Success APPLICATION END	Call Result Success SYSTEM ERROR	Call Result Success UNKNOWN	Call Res Failed USER EN
Random Action #1	4049	10.68%	24.94%	23.83%	25.54%	25.69%	26.07%
Random Action #10	3821	10.08%	25.31%	25.31%	24.46%	24.92%	25.62%
Random Action #2	3785	9.98%	27.35%	23.38%	25.21%	24.06%	24.72%
Random Action #3	3511	9.26%	23.85%	25.5%	25.58%	25.06%	23.5%
Random Action #4	4151	10.95%	25.67%	23.86%	26.39%	24.08%	27.46%
Random Action #5	3702	9.77%	25.08%	25.74%	24.26%	24.92%	22.6%
Random Action #6	4143	10.93%	24.12%	25.07%	24.49%	26.32%	24.07%
Random Action #7	3442	9.08%	24.44%	22.3%	25.28%	27.97%	23.53%
Random Action #8	3941	10.4%	26.29%	23.21%	27.21%	23.29%	27.2%
Random Action #9	3363	8.87%	23.71%	24.7%	25.43%	26.16%	24.91%

Figure 25: Last IVR Action Used Report

Table 54 describes the fields for the Last IVR Action Used report.

Table 54: 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 in which the last IVR Action was used.

Table 54: Last IVR Action Used Report Fields (Continued)

Field	Description
% of Calls	The percentage of the total number of calls in which this IVR Action was used.
Call Result	The percentage of calls that used the Last IVR Action, with the given End Action. Valid Call Results are: • SUCCESS—The call was processed successfully. • FAILED—A failure occurred that prevented the call from being processed properly—for example, a database error or network error. • REJECTED—The MCP rejected the call—for example, the VoiceXML application could not be fetched from the application server. • UNKNOWN—Some unknown reason caused the call to end abruptly. Valid End Actions: • 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.

Part

4 Appendixes

This part of the manual contains miscellaneous reference information in the following appendixes:

- Appendix A, “Module and Specifier IDs,” on page 255
- Appendix B, “SIP Response Codes,” on page 279
- Appendix C, “Media Control Platform Reference Information,” on page 289
- Appendix D, “Default Device Profiles,” on page 304
- Appendix E, “Specifications and Standards,” on page 307
- Appendix F, “Caching Reference Information,” on page 315

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 and the Call Control Platform generate, including metric IDs and descriptions, see the *Genesys Voice Platform 8.0 Metrics Reference*.

Media Control Platform

Table 55 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 265.

Table 55: 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
SESSMGR	Call Manager API	28	SESSMGR

Table 55: Media Control Platform Application Module Names and IDs (Continued)

Module Name	Description or Comment	Module ID	Specifiers (link)
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

Table 56 lists the Media Control Platform specifier names and IDs.

Table 56: Media Control Platform Specifier Names and IDs

Specifier ID	Specifier Name
MTMPC	
1001	CMLOGMOD_MTMPC_INITFAILED
2001	CMLOGMOD_MTMPC_CONNERROR
LMBase	
1001	CMLOGMOD_LMBASE_IDGENDIRUNACCBLE

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
4001	CMLOGMOD_SESSMGR_INBOUNDDTMF
4003	CMLOGMOD_SESSMGR_NOINOUTLINES
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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
1026	CMLOGMOD_SMMAIN_CANTCREATEVGNETLIB
1027	CMLOGMOD_SMMAIN_CANTINITVGNETLIB
1028	CMLOGMOD_SMMAIN_FAILINITCFGOBJ
1029	CMLOGMOD_SMMAIN_CANTSTARTCMGR
2002	CMLOGMOD_SMMAIN_FAILLOADAPPMODLIB
2003	CMLOGMOD_SMMAIN_FAILINITAPPMOD
2004	CMLOGMOD_SMMAIN_NOVLDAPPMODINLIB
2005	CMLOGMOD_SMMAIN_LIBNODEFMAKEAPPMOD
2006	CMLOGMOD_SMMAIN_VXMLAPPMODNOTLOAD
CMUTIL	
2001	CMLOGMOD_CMUTIL_TELNUMLONG
2002	CMLOGMOD_CMUTIL_TELNUMINVCHAR
2003	CMLOGMOD_CMUTIL_POSTDIALLONG
2004	CMLOGMOD_CMUTIL_POSTDIALINVCHAR

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
APPMODULE	
1001	CMLOGMOD_APPMODULE_FAILSTRTWORKNGTHRD
2001	CMLOGMOD_APPMODULE_FAILREGAPP
2002	CMLOGMOD_APPMODULE_FAILREGAPPMOD
2003	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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
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
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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
3003	MPCLOGMOD_CONTROL_CONNMODIFYFAILED
3004	MPCLOGMOD_CONTROL_SENDDTMFNOTALLOWED
3005	MPCLOGMOD_CONTROL_INVALIDCONFIGPARAM
MEDIA	
2001	MPCLOGMOD_MEDIA_RECORDOPENFAILED
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

Table 56: Media Control Platform Specifier Names and IDs (Continued)

Specifier ID	Specifier Name
DSP	
3002	MPCLOGMOD_DSP_NOTRANSCODER
VGULOGMOD_MAIN	
1002	VGLOG_CANT_OPEN_DLL
2003	VGLOG_SOCKET_SEND_FAILED
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
3001	VGLOG_RTSP_SOCKET_EVENT
5001	VGLOG_RTSP_SOCKET_CLOSE

Next Generation Interpreter Module ID and Specifiers

The Module ID for the Next Generation Interpreter (NGI) application is 192. Table 57 describes the specifiers for the NGI application module.

Table 57: 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)
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

Table 57: NGI Specifiers (Continued)

Specifier ID	Specifier Name	Level
1014	NGI_LOG_PROMPT_FETCH_TIMEOUT	Error
1015	NGI_LOG_PROMPT_FETCH_ERROR	Error

Call Control Platform

Table 58 lists the Call Control Platform Application Module names and IDs.

Table 58: 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 59 describes the specifiers for the main Call Control Platform module (Module ID = 151). These events are related to a connection, dialog, or conference.

Table 59: 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.
151 (continued)	**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 60 describes the Media Controller events.

Table 60: 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`.
	Info Events	
	1024	Connection timeout.
	1025	Dialog Unsupported MIME Type.
	1026	Conference created.

Table 60: 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 61 describes the CCXML interpreter events at the INFO level.

Table 61: CCXMLI Log_4 INFO Events

Module ID	Specifier ID	Description
152	1024	CCXMLI initialized.
	1025	CCXMLI 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.

Resource Manager

The Module ID for the Resource Manager application is 148.

Table 62 describes the specifiers for the Resource Manager application module.

Table 62: Resource Manager Specifiers

Specifier ID	Specifier Name
257	GVPLOG_RM_UNRECOVERABLEERR
513	GVPLOG_RM_CONFIGERR
514	GVPLOG_RM_CCPSS7ERR
515	GVPLOG_RM_SOCKETERR

Table 62: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
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
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

Table 62: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
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
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

Table 62: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
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
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

Table 62: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
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
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
1025	GVPLOG_RM_CCPSS7STATE
1026	GVPLOG_RM_COMMNOTICE

Table 62: Resource Manager Specifiers (Continued)

Specifier ID	Specifier Name
1027	GVPLOG_RM_CLUSTERNOTICE
1028	GVPLOG_RM_CCPPROXYSTATE
1029	GVPLOG_RM_STARTUP
1030	GVPLOG_RM_SHUTDOWN
1031	GVPLOG_RM_RMSERVICEAGENTSTATUS
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
2003	VGLOG_SOCKET_SEND_FAILED
2305	GVPLOG_RM_GENERIC_TRACE

Fetching Module

The Module ID for the Fetching Module application is 80.

Table 63 describes the specifiers for the Fetching Module application module.

Table 63: 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.

Table 63: Fetching Module Specifiers (Continued)

Specifier ID	Specifier Name	Description
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.
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 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 72 on page 299.

SIP Responses within GVP

Table 64 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 64 also lists the configuration options to customize those responses, where applicable.

The Resource Manager handles SIP responses from other components in accordance with rules described on page 36.

Table 64: SIP Response Codes

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
100	Trying	MCP CCP RM*	The immediate response to a valid `INVITE` request.	*For CCP-initiated outbound calls.

Table 64: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
180	Ringing	MCP	The default intermediate response to an `INVITE` request.	`sip.sendalert` (see page 175)
		CCP	Intermediate response 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`
183	Session Progress	MCP	The non-default intermediate response, which includes SDP information.	`sip.sendalert` (see page 175)
		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 64: 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 media bridging changes are required, 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.	
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 71 on page 295.
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 to be used.
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.	

Table 64: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
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.

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.	
405	Method Not Allowed	RM	`REFER`, `OPTIONS`, `SUBSCRIBE`, or `INFO` message was sent outside of an existing SIP dialog.	

Table 64: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
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 all requests are accepted.)	
420	Bad Extension	MCP	The MCP rejects an incoming call that requires `100rel` (SIP Provisional Message Reliability).	
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	Conference call fails because of insufficient resource port capacity or because conference has reached maximum size.	`rm.conference-sip-error-respcode`
		RM	The Resource Manager is not able to select a resource to which to forward the request, 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`

Table 64: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
480	Temporarily Unavailable (continued)	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 to be used.
		CCP	The platform is not in `READY` state, and is therefore rejecting all `INVITE` and `OPTIONS` requests.	`ccpccxml.defaultrejectcode`
487	Request Terminated	MCP	A `CANCEL` or `BYE` is received before the final response to the `INVITE` was 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 warning text one of the following: • `Unable to generate an offer`—The `INVITE` contained no SDP. • `Unable to generate an answer`—All other situations.

Table 64: 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` 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 page 36).	
		MCP	Unable to create a media session to handle the call.	The response includes an explanatory `Warning` header (see `Warning` header information in Table 71 on page 295).
		MCP	The MCP is unable to fetch or parse the VoiceXML document.	
503	Service Unavailable	RM	The Resource Manager is in suspend mode when a new session request arrives.	`rm.suspend-mode-respcode`

Table 64: 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 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`
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`

Table 64: SIP Response Codes (Continued)

SIP Response		Sent By	Situations	Configurable Options and Notes
Code	Phrase			
[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

C 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

Table 65 lists the supported audio-only file formats for playing prompts.

Table 65: 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/basic*	Audio with `.au` header	8-bit mono	G.711 ulaw, G.711 alaw, PCM, ADPCM (depends on file header information)
.ulaw	audio/basic*	Raw audio	8-bit mono	G.711 ulaw
.alaw	audio/x-alaw-basic	Raw audio	8-bit mono	G.711 alaw
.g729	audio/g729	Raw audio		G.729
.pcm	audio/pcm audio/x-pcm	Raw audio	8-bit unsigned mono	Linear PCM
.adpcm24	audio/x-g726-24	Raw audio	24 kb/sec	ADPCM (G.726)
.adpcm	audio/x-g726 audio/x-adpcm audio/adpcm audio/x-adpcm8	Raw audio	32 kb/sec	ADPCM (G.726)
.adpcm40	audio/x-g726-40	Raw audio	40 kb/sec	ADPCM (G.726)
.pcm8	audio/L8 audio/pcm8 audio/x-pcm8	Raw audio	8-bit unsigned mono	Linear PCM
.pcm16	audio/L16	Raw audio	16-bit signed mono	Linear PCM
Note: The sample rate is always 8000 Hz. If a non-8000 Hz audio file is detected, a warning message will be issued, and the prompt will be played as if the sampling rate is 8000 Hz. A configurable Media Control Platform parameter, `mpc.mediamgr.strictsamplingrate`, enables you to prevent non-8000 Hz audio files from being played.				

Table 65: Supported Audio File Formats—Play (Continued)

Expected File Extension	MIME-type	File Format	Sample Size	Encoding
.wav	audio/wav audio/x-wav	Audio with .wav header		G.711 ulaw, G.711 alaw, PCM, 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	Raw audio		gsm 6.10
.amr	audio/amr	Raw audio		AMR
.3gp	audio/3gpp	Audio stored in 3GP container		AMR
Note: The sample rate is always 8000 Hz. If a non-8000 Hz audio file is detected, a warning message will be issued, and the prompt will be played as if the sampling rate is 8000 Hz. A configurable Media Control Platform parameter, `mpc.mediamgr.strictsamplingrate`, enables you to prevent non-8000 Hz audio files from being played.				

* The Media Control Platform examines the audio data to determine whether an audio/basic file is actually of `.au` format or `.ulaw` format.

Video-Only Formats—Play

Table 66 lists the supported video-only file formats for playing prompts.

Table 66: 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

Combined Audio and Video Formats—Play

Table 67 lists the supported audio/video file formats for playing prompts.

Table 67: 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 (depends on file header information)
.3gp	video/3gpp	• Audio: 8000 Hz • Video: 30 fps (recommended)	Audio/video stored in 3GP container	• Audio: AMR • Video: h263, h263-1998 (depends on file header information)

Audio-Only Formats—Record

Table 68 lists the supported audio-only file formats for recording.

Table 68: Supported Audio File Formats—Record

MIME-type	Recorded File Format	Sample Size	Encoding	File Extension
audio/x-vox audio/vox	Raw audio	8-bit mono	G.711 ulaw, G.711 alaw (depends on platform configuration)	.vox
audio/basic	Raw audio	8-bit mono	G.711 ulaw	.ulaw
audio/x-alaw-basic	Raw audio	8-bit mono	G.711 alaw	.alaw
audio/g729	Raw audio		G.729	.g729
audio/pcm audio/x-pcm	Raw audio	8-bit unsigned mono	Linear PCM	.pcm
audio/x-g726-24	Raw audio	24 kb/sec	ADPCM (G.726)	.adpcm24

Notes:

- Only the 8000 Hz audio sampling rate is supported.
- Genesys Voice Platform (GVP) 8.0 does not support `.au` and `.nist` file recording.

Table 68: Supported Audio File Formats—Record (Continued)

MIME-type	Recorded File Format	Sample Size	Encoding	File Extension
audio/x-g726 audio/x-adpcm audio/adpcm audio/x-adpcm8	Raw audio	32 kb/sec	ADPCM (G.726)	.adpcm
audio/x-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, 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
audio/x-gsm	Raw audio		gsm 6.10	.gsm
audio/amr	Raw audio		AMR	.amr
audio/3gpp	Audio stored in 3GP container		AMR	.3gp

Notes:

- Only the 8000 Hz audio sampling rate is supported.
- Genesys Voice Platform (GVP) 8.0 does not support `.au` and `.nist` file recording.

Video-Only Formats—Record

Table 69 lists the supported video-only file formats for recording.

Table 69: 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

Combined Audio and Video Formats—Record

Table 70 lists the supported audio/video file formats for recording.

Table 70: 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, gsm, g726, none. **Default:** ulaw or alaw (depends on platform configuration). video_codec: h263, h263-1998. **Default:** h263 g726_encoding_rate: 16 kb, 24 kb, 32 kb, or 40 kb. **Default:** 32 kb.	.avi
video/3gpp;codec=<audio_ codec> ;videocodec=<video_codec>	Audio/video stored in 3GP container	audio_codec: amr, none. **Default:** amr video_codec: h263, h263-1998. **Default:** h263	.3gp
Note: Only the 8000 Hz audio sampling rate is supported.			

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`, or from the `Request-URI` of a `REFER` that triggers an outbound call.

The following configuration options can be set dynamically:

asr Section

load_once_per_call*
delay_for_dtmf
log_metrics_to_asr

sessmgr Section

maxincalltime
ECS_Fallback
join_fallback
record.start.beep.filename
inbandxferprefix
inbandxfertimeout
alert_before_fetch
mediaswitch_on_alert
acceptcalltimeout

mpc Section

codec
codecpref
fcr.defaultdtmfhandling
transmitmultiplecodec
appendrejcodec
rtp.dtmf.receive
rtp.dtmf.send

sip Section

warningheaders
sendalert
sendrecvevents

* 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.

SIP Headers

Table 71 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 71 for any custom headers, or they will be ignored.

Table 71: 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 SIP OPTIONS request, lists all the methods supported by the component.
Call-ID	RFC 3261 (20.8)	Standard support.

Table 71: SIP Headers Known to GVP (Continued)

SIP Header Name	Standard/Specification (Section)	Description
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 `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.
Min-Expires	RFC 4028 (5)	Minimum session timer.
Max-Forwards	RFC 3261 (20.22)	Standard support.
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.

Table 71: SIP Headers Known to GVP (Continued)

SIP Header Name	Standard/Specification (Section)	Description
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. The value can be set by the application or by configuration. If the `INVITE` contains `Record-Route` headers, `Record-Route` values override the configured routeset for all requests within the dialog.
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.
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`

Table 71: SIP Headers Known to GVP (Continued)

SIP Header Name	Standard/Specification (Section)	Description
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)
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, VoiceXML or CCXML application).

Handling Error Responses for Outbound Calls

Table 72 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 B, "SIP Response Codes," on page 279.

Table 72: 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			
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			
405	Method Not Allowed	unsupported	CM_DISCREASON_UNSUPPORTED	`error.unsupported.transfer.blind/consultation/bridge` event during `<transfer>`
488	Not Acceptable Here			
501	Not Implemented			

Table 72: 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			
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 73 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 73 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 8.0 VoiceXML 2.1 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 73: 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** • *`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 whitespace in the *`result`* is ignored. • *`reason`* content beyond 256 characters will be truncated.
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** • *`notes`* cannot be empty. • Content beyond 4 KB will be truncated.
ivr_action_start	`<log label="com.genesyslab.var.ActionStart">`*`actionID`*`[|parentID=`*`PID`*`]</log>` where: • *`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** • *`actionID`* and *`PID`* are any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. • Action IDs are case-sensitive. • Whitespace is ignored. • An active action is an action that has started and has 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 73: 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: • *actionID* is the ID of the VoiceXML application action being ended. • *result* is one of `SUCCESS|FAILED|UNKNOWN`, indicating the result of the action. The default is `UNKNOWN`. • *reason* 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** • *actionID* is any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. • *actionID* is case-sensitive. • *result* and *reason* values are not case-sensitive. • Whitespace 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.

Table 73: VAR <log> Tags and Metrics (Continued)

Metric	<log> Tag Label Syntax and Valid Values
ivr_action_notes	`<log label="com.genesyslab.var.ActionNotes">`*`actionID`*`|`*`notes`*`</log>` where: • *`actionID`* is the ID of the VoiceXML application action. • *`notes`* are up to 4 KB (4096 bytes) of free-form notes associated with the action. **Notes** • *`actionID`* is any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. Preceding and trailing whitespace is ignored. • *`notes`* cannot be empty. • *`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.
custom_var	`<log label="com.genesyslab.var.CustomVar">`*`name`*`|`*`value`*`</log>` where: • *`name`* is the name of the custom variable. • *`value`* is the value of the custom variable. **Notes** • *`name`* is any valid UTF8 string, to a maximum of 64 characters, that does not contain spaces or pipes. Preceding and trailing whitespace is ignored. • *`value`* is any valid UTF8 string, to a maximum of 256 characters. Whitespace 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

D Default Device Profiles

This appendix summarizes the settings for the default device profiles that are provisioned for the Call Control Platform. These device profiles are defined in the `<Call Control Platform Installation Directory>\config\ccpccxml_provision.dat` file.

Table 74 summarizes the 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 Table 74.
- `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. For descriptions of the properties, see Table 33 on page 189.

Table 74: Default Device Profile Settings

Item		Value for										
		Cisco Gateway	**Default Inbound**	**Default Outbound**	**Default Conference**	**Default Dialog**	**Audio-codes Gateway**	**Convedia Media Server**	**X-Lite**	**Brooktrout Snowshore**	**GVP MCP**	**Audio-codes MP104**
Entry ID		1	2	3	4	5	6	7	8	9	10	11
Precedence		1	0	0	0	0	2	3	4	5	6	7
Profile Name		Cisco Gateway	Default Inbound	Default Outbound	Default Conference	Default Dialog	Audio-codes Gateway	Convedia Media Server	X-Lite	Brooktrout Snowshore	GVP MCP	Audio-codes MP104
Number of properties		10	10	10	10	10	10	10	10	10	10	10
Properties	sendonly-support	false	true	true	true	true	true	true	true	false	true	true
	recvonly-support	false	true	true	true	true	true	true	true	false	true	true
	distinct-send-recv-support	false	true	true	true	true	false	false	true	false	true	false
	multiple-recvonly-support	false	true	true	true	true	false	false	true	false	true	true
	restricts-media-source	false	true	true	true	true	true	true	true	true	true	false
	connectionless-sdp-type	hold	hold	non-routable	non-routable	non-routable	hold	hold	non-routable	hold	non-rout-able	hold
	offer-answer-support	false	true	true	true	true	false	true	false	true	true	true
	nomedia-SDP-support	true	true	true	false	false	false	true	false	false	false	false
	offer-less-invite-support	true	true	true	true	true	true	true	true	true	true	false
	options-support	false	false	false	false	false	false	false	false	false	false	false
User-Agent		Cisco	Inbound	Outbound	Conference	Dialog	Audiocodes	Convedia	X-Lite	Brooktrout	GVP MCP	Audio codes MP104

Appendix

E Specifications and Standards

This appendix describes the specifications and standards that GVP supports.

It contains the following sections:

Specifications

The following specifications are published and maintained by the W3C Voice Browser Working Group:

- VoiceXML Specification—*W3C Voice Extensible Markup Language (VoiceXML) 2.1, W3C Recommendation 19 June 2007* and *W3C Voice Extensible Markup Language (VoiceXML) 2.0, W3C Recommendation 16 March 2004*
- Speech Synthesis Markup Language Specification—*W3C Speech Synthesis Markup Language (SSML) Version 1.0, Recommendation 7 September 2004*
- Speech Recognition Grammar Specification—*Speech Recognition Grammar Specification Version 1.0, W3C Recommendation 16 March 2004*
- CCXML Specification—*W3C Voice Browser Call Control: CCXML Version 1.0, W3C Working Draft 29 June 2005*

Related Standards

GVP is based on open standards. As a result, the platform provides complete or subset support for many Requests for Comments (RFCs) that are published and maintained by the Internet Engineering Task Force (IETF). For more information, see `http://www.ietf.org`.

The IETF standards that GVP supports include:

- RFC 1738 Uniform Resource Locators
- RFC 1808 Relative Uniform Resource Locators
- RFC 1867 Form-based File Upload in HTML
- RFC 2109 HTTP State Management Mechanism
- RFC 2190 RTP Payload Format for H.263 Video Streams
- RFC 2388 Returning Values from Forms: multipart/form-data
- RFC 2326 Real Time Streaming Protocol (RTSP)
- RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax
- RFC 2429 RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H263+)
- RFC 2616 Hypertext Transfer Protocol – HTTP/1.1 (subset)
- RFC 2806 URLs for Telephone Calls
- RFC 2833 RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals
- RFC 2964 Use of HTTP State Management
- RFC 2965 HTTP State Management Mechanism
- RFC 2976 The SIP INFO Method
- RFC 3261 SIP: Session Initiation Protocol
- RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP)
- RFC 3267 Real Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs
- RFC 3515 The SIP REFER Method
- RFC 3550 RTP: A Transport Protocol for Real-Time Applications
- RFC 3986 Uniform Resource Identifier (URI): Generic Syntax
- RFC 4240 Basic Network Media Services with SIP—GVP provides support for conference service and dialogs
- RFC 4568 Session Description Protocol (SDP) Security Descriptions for Media Streams

- Most extensions from proposed IETF draft *SIP Interface to VoiceXML Media Services* (`http://tools.ietf.org/id/draft-burke-vxml-02.txt`)—see "Burke Draft Support"
- Some extensions from proposed IETF draft *Basic Network Media Services with SIP* (`http://www.ietf.org/rfc/rfc4240.txt`)

The platform also includes complete support for many network-related and other protocols (for example, TCP/IP, SNMP). Contact Genesys for additional details.

Burke Draft Support

This section describes various aspects of the Burke Draft (`http://www.ietf.org/internet-drafts/draft-ietf-mediactrl-vxml-01.txt`), and the features supported by GVP 8.0.

The Burke Draft describes the SIP interface to VoiceXML media services. The following Burke Draft sections are covered:

- 2—"VoiceXML Session Establishment and Termination"
- 3 —"Media Support" on page 311
- 4 — "Returning Data to the Application Server" on page 312
- 5 — "Outbound Calling" on page 313
- 6 — "Call Transfer" on page 313

Table 75 lists and describes the Burke Draft requirements and which ones GVP 8.0 supports.

Table 75: Burke Draft Support Requirements, by Section

Section	Requirement	Current Support
VoiceXML Session Establishment and Termination		
2.1	Support for the following service identification parameters: • `voicexml` • `maxage` [RFC2161] • `maxstale` [RFC2616] • `method` • `postbody` • `ccxml` [RFC4627] • `aai` [RFC4627]	Supported
	Support for incorrectly formed requests with the 4xx class response.	Not Supported

Table 75: Burke Draft Support Requirements, by Section (Continued)

Section	Requirement	Current Support
2.1 (continued)	Support for repeated init-parameters rejected with the `400 Bad Request` response.	Supported
	Support for parameter URL-encoding.	Supported
2.2	Support for `100 Trying`, followed by a `200 OK` response upon receipt of a `INVITE` when a document has been fetched. After the `ACK` is received, the application executes.	Supported
	Support for optimization before sending the `200 OK` response.	Not Supported
	Support for the inability to accept `INVITE` requests, and to respond as defined by [RFC3261] with the exception of the following error conditions: • If the request does not conform to the specification, return a `400 Bad Request` response. • If the request does not include a `voicexml` parameter, and the default page is not configured, return a `400 Bad Request` and a `399 Warning` header.	Not Supported
	Support for returning a `500 Server Internal Error` with a `399 Warning` header if the document cannot be fetched or parsed.	Supported
	Support for large transport appropriate messages (such as TCP) when an `INVITE` request exceeds the MTU of the underlying network.	Supported
2.3	Support for not exiting VoiceXML application until a `re-INVITE` with port information is sent if starting a `Dialog-INVITE` without media; a `200 OK` with offered media; an `ACK` with media, but the media ports are set to zero (`0`); or an `INVITE` with SDP without media lines followed by a regular `INVITE`, `200 OK`, or `ACK` flow.	Supported
	Support for a `re-INVITE` that disables media stream without affecting the executing VoiceXML application once the application is running.	Supported
2.4	Support for the following session variables: • `session.connection.local.uri` • `session.conection.remote.uri` • `session.connection.redirect` • `session.connection.protocol.sip.headers` • `session.connection.aai` • `session.connection.ccxml`	Supported

Table 75: Burke Draft Support Requirements, by Section (Continued)

Section	Requirement	Current Support
2.4 (continued)	Support for evaluating the following session variables: • `session.connection.protocol.name` to `sip`. • `session.connection.protocol.version` to `2.0`	Supported
	Support for using the `session.connection.protocol.sip.requesturi` variable as an associative array formed from the URI parameters.	Supported
	Support for using the `session.connection.protocol.sip.media` variable as an array where each array element is an object with the following properties: • `type` • `direction` • `format` **Note:** This parameter will be updated as the media values involved in the session change.	Supported
2.5	Support for sending a `200 OK` in response to a `BYE`, and then generating a `connection.disconnect.hangup` event.	Supported
	Support for providing the value of the `Reason` header verbatim through the `_message` variable if a `Reason` header [RFC3326] is present in the `BYE` Request. **Note:** Set `sip.in.bye.headers` to `Reason`.	Supported
	Support for terminating a session with a `BYE` Request when the VoiceXML application encounters a `<disconnect>` or `<exit>`, the VoiceXML application completes, or the VoiceXML application has unhandled errors.	Supported
Media Support		
3.1	Support for the Offer/Answer mechanism of [RFC3960].	Supported
3.2	Support for early media streams as described in [RFC3960].	Supported
3.3	Support for the using a `re-INVITE` to modify a media session.	Supported

Table 75: Burke Draft Support Requirements, by Section (Continued)

Section	Requirement	Current Support
3.4	Support for the following Codecs: • G.711 mu-law and A-law using the RTP payload type 0 and 8—set `mpc.codec` to `pcmu pcma` • H.263 Baseline—set `mpc.codec` to `h263` • AMR-NB audio—set `mpc.codec` to `amr` • Other codecs and payload formats—set `mpc.codec` as specified for the various codecs.	Supported
	Support for the following Codecs: • MPEG-4 video • MPEG-4 AAC audio	Not Supported
3.5	Support for DTMF events [RFC4733].	Supported
Returning Data to the Application Server		
4.1	Support for returning data to the application server with a HTTP Post using the `<submit>`, `<subdialog>`, or `<data>` elements.	Supported
4.2	Support for returning data to the application server with the `SIP expression` or the `namelist` attribute on the `<exit>` element, or the `namelist` attribute on the `<disconnect>` element.	Supported
	Support for encoding the `expr` or the `namelist` data in the `BYE` request body when encountering the `<exit>` or `<disconnect>` element.	Supported
	Support for including the `expr` or the `namelist` data in the `200 OK` response to a `BYE` request.	Not Supported
	Support for sending a `100 Trying` response to a `BYE` request [RFC4320].	Not Supported
	Support for using the `_reason` reserved name to differentiate between a `BYE` resulting from a `<disconnect>` element and a `BYE` resulting from an `<exit>` element. For example, `_reason = exit`.	Supported
	Support for using the `_exit` reserved name if the `expr` attribute is specified on the `<exit>` element instead of the `namelist` attribute. For example, `_exit=<value>`	Supported

Table 75: Burke Draft Support Requirements, by Section (Continued)

Section	Requirement	Current Support
Outbound Calling		
5.0	Support for triggering outbound calls using third party call control [RFC3725].	Supported
Call Transfer		
6.1	Support for blind transfers with a `REFER` message on the original SIP dialog [RFC3515]. **Note:** Set `sip.defaultblindxfer` to `REFER`.	Supported
	Support for terminating the session with a `BYE` message and generating the `connection.disconnect.transfer` event if a `REFER` request is accepted with a 2xx response.	Supported
	Support for the following form item variables and events depending on the SIP response if the `REFER` is accepted with a non-2xx response: • `404 Not Found` = `error.connection.baddestination` • `405 Method Not Allowed` = `error.unsupported.transfer.blind` • `503 Service Unavailable` = `error.connection.noresource` • (No response) = `network_busy` • (Other 3xx/4xx/5xx/6xx) = `unknown`	Supported
	Support for appending the `aai` or `aaiexpr` attribute to the `Refer_To` URI as a parameter named `aai`.	Supported
	Support for URL-encoding reserved characters as required for SIP/SIPS URIs [RFC3261].	Supported
6.2	Support for ejecting the callee if the bridged transfer is terminated.	Supported
	Support for appending the `aai` or `aaiexpr` attribute to the `Refer_To` URI in the `INVITE` as a parameter named `aai`.	Supported
	Support for URL-encoding reserved characters as required for SIP/SIPS URIs [RFC3261].	Supported
	Supporting for playing early media from the callee to the caller if the `transferaudio` attribute is omitted.	Supported
	Support for setting the `<transfer>`'s form attribute to `noanswer` after issuing a `CANCEL` when `connectiontimeout` expires.	Supported

Table 75: Burke Draft Support Requirements, by Section (Continued)

Section	Requirement	Current Support
6.2 (continued)	Support for the following form item variables and events depending on the SIP response if the `INVITE` is accepted with a non-2xx response: • `404 Not Found` = `error.connection.baddestination` • `405 Method Not Allowed` = `error.unsupported.transfer.blind` • `408 Request Timeout` = `noanswer` • `486 Busy Here` = `busy` • `503 Service Unavailable` = `error.connection.noresource` • (No response) = `network_busy` • (Other 3xx/4xx/5xx/6xx) = `unknown`	Supported
	Support for listening for speech or DTMF hotword results in a near end disconnect.	Supported
	Support for issuing a `BYE` if the call duration exceeds the maximum duration specified in the `maxtime` attribute.	Supported
6.3	Support for the following form item variables and events depending on the SIP response if the `INVITE` is accepted with a non-2xx response: • `404 Not Found` = `error.connection.baddestination` • `405 Method Not Allowed` = `error.unsupported.transfer.consultation` • `408 Request Timeout` = `noanswer` • `486 Busy Here` = `busy` • `503 Service Unavailable` = `error.connection.noresource` • (No response) = `network_busy` • (Other 3xx/4xx/5xx/6xx) = `unknown`	Supported
	Support for generating the `connection.disconnect.transfer` event when receiving a `200 OK` to a `NOTIFY` request.	Supported
	Support for setting the VoiceXML `input item` variable to `unknown` with the non-2xx response to the `NOTIFY` request.	Not Supported

Appendix

F Caching Reference Information

This appendix provides information about the Fetching Module and Squid caching algorithms, as well as information about the Squid access log.

It contains the following sections:

Caching Algorithms

This section provides the algorithms that the Fetching Module and Squid use to determine when they will use cached documents:

Fetching Module Caching Algorithm

When the VoiceXML interpreter requests the Fetching Module to perform a fetch to URI, the Fetching Module uses the following algorithm to determine whether it will use the cached version within its own memory:

```
if ( URI contains one of the items in the iproxy.no_cache_url_substr
list )
  re-fetch the item from the squid proxy
else
  if (the URI matches exactly [including all parameters] with a URI
in Fetching Module cache)
  if (the previous was a fetch error)
```

```
  if (the previous result was fetched within
iproxy.cache_error_max_age seconds)
  return result from cache;
  end if
  else if (the previous was a successful fetch)
  if (the previous result was fetched within iproxy.cache_max_age
seconds)
  return result from cache;
  end if
  end if
  end if
  re-fetch the item from the squid proxy
  end if
```

Squid Caching Algorithm

The following algorithm summarizes the caching policy rules, and represents the Squid caching proxy behavior when a resource is requested by the VoiceXML interpreter.

```
If the resource is not present in the cache, fetch it from the
server using get.

If the resource is in the cache,
  If a maxage value is provided,
    If age of the cached resource <= maxage,
     If the resource has expired,
     Perform maxstale check.
     Otherwise, use the cached copy.
    Otherwise, fetch it from the server using get.
  Otherwise,
    If the resource has expired,
    Perform maxstale check.
    Otherwise, use the cached copy.
```

The algorithm for `maxstale` check is:

```
If maxstale is provided,
  If cached copy has exceeded its expiration time by no more than
maxstale seconds,
  then use the cached copy.
  Otherwise, fetch it from the server using get.
Otherwise, fetch it from the server using get.
```

Squid Expiry Time Algorithm

Determining the freshness or staleness of an object is a multi-step process:

1. Squid calculates the following values:
   - `AGE`—How much the object has aged since it was retrieved. (`AGE = NOW - OBJECT_DATE`)

- LM_AGE—How old the object was when it was retrieved. (LM_AGE = OBJECT_DATE - LAST_MODIFIED_TIME)
- LM_FACTOR—The ratio of AGE to LM_AGE. (LM_FACTOR = AGE / LM_AGE)
- CLIENT_MAX_AGE—(Optional) The maximum object age the client will accept, as taken from the Cache-Control request header.
- EXPIRES—(Optional) The expiry time from the server reply headers.

2. Squid compares the calculated values against the parameters of the refresh_pattern rules (see "Refresh-Pattern Rules" on page 201). Squid checks the URL regular expressions in the order in which the refresh-pattern rules are listed, until a match is found. Squid uses the first entry that matches. If no match is found, the following default values are used:
   - MAX_AGE = 4320 minutes
   - MIN_AGE = 0
   - PERCENT = 20%
3. When a match is found, Squid applies the following algorithm to determine if the object is fresh or stale:

```
if (CLIENT_MAX_AGE)
 if (AGE > CLIENT_MAX_AGE)
  return STALE
if (AGE <= MIN_AGE)
 return FRESH
if (EXPIRES) {
 if (EXPIRES <= NOW)
  return STALE
 else
  return FRESH
}
if (AGE > MAX_AGE)
 return STALE
if (LM_FACTOR < PERCENT)
 return FRESH
return STALE
```

Note: The MAX-AGE in a client request takes the highest precedence. Genesys recommends that you usually set the MIN_AGE value to zero, because it has higher precedence than the Expires: value from the server. However, use the MIN_AGE value if you wish to override the Expires: headers.

Squid Access Logs

The Squid access.log is stored in the following location:

```
C:\squid\var\logs\
```

Table 76 describes the fields in each `access.log` entry.

Table 76: Squid Access Log Fields

Field	Description
Timestamp	The time when the client socket is closed. The format is "Unix time" (seconds since Jan 1, 1970), with millisecond resolution. Use the following command to modify this to visible format: `cat access.log | perl -nwe 's/^(\d+)/localtime($1)/e; print'`
Elapsed Time	The elapsed time of the request, in milliseconds. This is time between the `accept()` and `close()` of the client socket.
Client Address	The IP address of the connecting client, or the fully qualified domain name (FQDN) if the `log_fqdn` option is enabled in the configuration file. This parameter is normally turned off for performance reasons.
Log Tag/HTTP Code	The Log Tag describes how the request was treated locally (hit, miss, and so on). The HTTP code is the reply code taken from the first line of the HTTP reply header. Non-HTTP requests may have zero reply codes. For descriptions of the Log Tags, see Table 77 on page 318.
Size	The number of bytes written to the client.
Request Method	The HTTP request method, or `ICP_QUERY` for ICP requests.
URL	The requested URL.
Ident	If `ident_lookup` is on, this field may contain the username associated with the client connection as derived from the `ident` service. This lookup is typically turned off for performance reasons.
Hierarchy Data/ Hostname	A description of how and where the requested object was fetched.
Content Type	The `Content-type` field from the HTTP reply.

Table 77 describes the log tags for the following types of requests:

- TCP requests, on the HTTP port
- UDP requests, on the ICP port

Table 77: Log Tags in the Squid Access Logs

Access Log Tag	Description
TCP_ Requests	
TCP_HIT	A valid copy of the requested object was in the cache.

Table 77: Log Tags in the Squid Access Logs (Continued)

Access Log Tag	Description
TCP_MISS	The requested object was not in the cache.
TCP_REFRESH_HIT	The object was in the cache, but `STALE`. An `If-Modified-Since` request was made, and a `304 Not Modified` reply was received.
TCP_REF_FAIL_HIT	The object was in the cache, but `STALE`. The request to validate the object failed, so the old (stale) object was returned.
TCP_REFRESH_MISS	The object was in the cache, but `STALE`. An `If-Modified-Since` request was made and the reply contained new content.
TCP_CLIENT_REFRESH	The client issued a request with the `no-cache pragma`.
TCP_CLIENT_REFRESH_MISS	The client issued a `no-cache pragma`, or some analogous cache control command along with the request. Thus, the cache has to refetch the object from origin server. In short, the browser forced the proxy to check for a new version (usually caused by users clicking `Reload`, but can also be triggered in some browsers by selecting a bookmark).
TCP_IMS_HIT	The client issued an `If-Modified-Since` request and the object was in the cache and still fresh. `TCP_HIT` and `TCP_IMS_HIT` are hits, the only difference is that in the `TCP_IMS_HIT` case the browser already had an up-to-date version so there was no need to send the Squid cached copy to the requestor.
TCP_IMS_MISS	The client issued an `If-Modified-Since` request for a stale object.
TCP_NEGATIVE_HIT	A previously failed request is satisfied from the cache, as the proxy believes it still be a problem.
TCP_SWAPFAIL	The object was believed to be in the cache, but could not be accessed.
TCP_DENIED	Access was denied for this request.
UDP_ Requests	
UDP_HIT	A valid copy of the requested object was in the cache.
UDP_HIT_OBJ	Same as `UDP_HIT`, but the object data was small enough to be sent in the UDP reply packet. Saves the next TCP request.
UDP_MISS	The requested object was not in the cache.
UDP_DENIED	Access was denied for this request.

Table 77: Log Tags in the Squid Access Logs (Continued)

Access Log Tag	Description
UDP_INVALID	An invalid request was received.
UDP_RELOADING	The ICP request was “refused” because the cache is busy reloading its metadata.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X