GVP Deployment Guide

Genesys Voice Platform 85

4/14/2015

Table of Contents

GVP Deployment Guide

[+] What is GVP?

Genesys Voice Platform (GVP) is a software suite that constitutes 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.

As part of the Voice Platform Solution (VPS), GVP is fully integrated with the Genesys Management Framework. GVP uses Genesys Administrator, the standard Genesys configuration and management graphical user interface (GUI), to configure, tune, activate, and manage GVP components 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).

[+] Intended Audience

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

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

Orientation	Installation and Deployment	Configuration and Maintenance
• Introduction • GVP Architecture • How GVP Works • GVP Call Flows • Specifications and Standards	• Prerequisites and Planning • Preparing the Operating System • Preparing the Hosts for GVP • Installing GVP	• Configuring GVP • Configuring Resource Manager High Availability • Configuring Reporting Server High Availability • Optimizing Performance through HTTP Caching • Maintaining GVP

Introduction to GVP 8.5

Overview

Genesys Voice Platform is a software suite that integrates a combination of call-processing, reporting, management, and application servers with Voice over Internet Protocol (VoIP) networks to deliver Web-driven dialog and call-control services to callers and enables Genesys customers to deliver interactive, media-centric applications to end users.

Integration with Genesys Framework

GVP is a major component of the Voice Platform Solution (VPS), which integrates GVP with the Genesys Framework to deliver next-generation voice processing that meets advanced call routing and voice self-service needs for an enterprise contact center. Although GVP is commonly used in enterprise self-service environments that use voice over telephone, many other applications including assisted service, multimedia interactions, and applications outside the contact center are possible.

GVP Interactive Voice Response

GVP differs from traditional Interactive Voice Response (IVR) solutions, in that it separates the voice and call-control applications from the call-processing environment. GVP does not rely on proprietary hardware, and it executes voice and call-control applications that are created in nonproprietary coding languages Voice Extensible Markup Language (VoiceXML) and Call Control Extensible Markup Language (CCXML). For the coding languages and other open standards that GVP supports, see Specifications and Standards. The GVP software includes a voice and call-control browser that interprets VoiceXML and CCXML documents into call-processing events. The voice and call-control applications are configured as IVR Profile objects that are provisioned through the Genesys Administrator Web-based user interface. The IVR Profiles define how requests received by the VPS are translated into concrete service requests that GVP components in the deployment can execute.

Third-Party Servers

Third-party application servers within a GVP deployment store and deliver the VoiceXML and CCXML applications. VoiceXML and CCXML documents can be generated dynamically by using any number of Web-based technologies, such as Active Server Pages (ASP) or

Java Server Pages (JSP), or by using a complete application development and execution environment, such as Genesys Composer. For more information about Composer, see Composer. GVP supports automatic speech recognition (ASR) and speech synthesis text-to-speech [TTS]) as part of a VoiceXML dialog through supported third party ASR and TTS engines that use the open standards listed in Specifications and Standards.

Features

Genesys Voice Platform provides a variety of features that support call handling for voice and call-control applications through either Time Division Multiplexing (TDM) or VoIP functionality. As a flexible, standards-based voice processing platform, GVP also expands traditional IVR functionality with self-service and assisted-service apabilities that are tightly integrated with the Genesys product suite.

Core Telephony Features

GVP 8.5 provides the following core telephony features:

- Call handling through Session Initiation Protocol (SIP).
- Support for major Private Branch Exchange (PBX) switches through the SIP Server.
- Support for major media gateways.
- Support for blind and consultative IP call transfers triggered by SIP REFER messages. SIP REFER messages also trigger Time Division Multiplexing (TDM)/Public Switched Telephone Network (PSTN) network transfers when the media gateway supports this functionality.
- Call Bridging, in which the inbound and outbound legs are maintained (for the call duration) when GVP sits in front of the switch.
- Media services, including voice prompts, menus, and data collection for example, Dual-Tone Multi-Frequency (DTMF) or speech.
- Acceptance and processing of information delivered with a call from the media gateway, including Automatic Number Identification (ANI), Dialed Number Identification Service (DNIS), and Calling Line Identification (CLID).

Advanced Features

The following advanced features are available:

- Support for voice and call-control applications written in standard VoiceXML and CCXML, respectively. For the coding language standards supported by GVP, see Specifications and Standards. GVP also supports extensions, to assist in the call-control requirements of a voice application.

- Support for automatic speech recognition.
- Support for text-to-speech.
- Conferencing.
- Call parking, providing multi-site contact centers with the ability to enable self-service and call queuing on GVP, before transferring or bridging the call to an agent.
- Intelligent call routing provided by Genesys Enterprise Routing Solution (ERS) and Network Routing Solution (NRS), when GVP is combined with other Genesys products.
- Graphical User Interface (GUI) for the development of VoiceXML applications using Composer. For more information, see Other Genesys VPS components.
- Provisioning, configuration, deployment, and monitoring using Genesys Administrator.

New in 8.5.x Releases

Supplement to Documentation includes:

- Latest Available IPs
- New in 8.5.1
- Documentation Corrections
- Deployment Notes

GVP Architecture

This chapter describes the primary components and basic architecture of Genesys Voice Platform (GVP) 8.5.

GVP and the Voice Platform Solution

GVP, Session Initiation Protocol (SIP) Server, Management Framework, and Genesys Administrator together constitute the Voice Platform Solution (VPS), which integrates voice self-service, agent-assisted service, and application-management functions into a single, IP-based contact-center solution.

GVP 8.5 provides a unified communication layer within the Genesys suite, and offers a robust solution that incorporates all required call control including computer-telephony integration (CTI) and media-related functions.

The figure below depicts the GVP architecture and the communication channels among GVP components in the VPS.

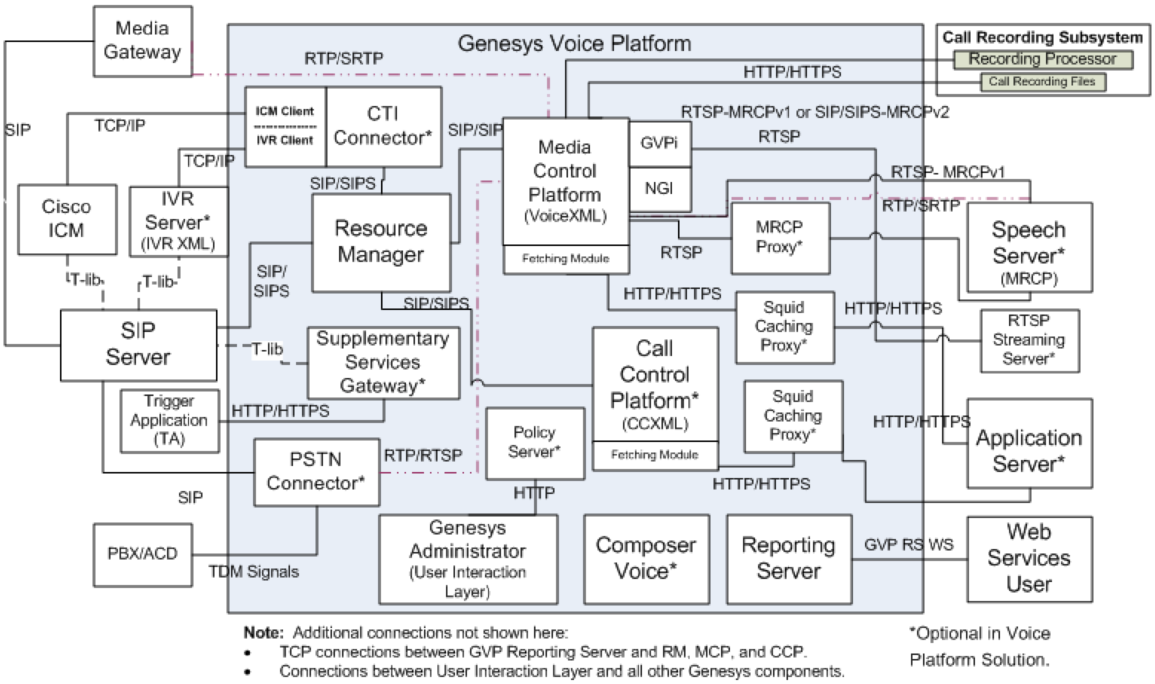


Figure: Genesys Voice Platform Solution Architecture

For more information about the VPS, see the Voice Platform Solution 8.1 Integration Guide.

GVP Components

GVP includes the following components:

- Resource Manager
- Policy Server
- CTI Connector
- PSTN Connector
- Media Control Platform
- Call Control Platform
- MRCP Proxy
- Fetching Module and Squid
- Supplementary Services Gateway
- Reporting Server

There is an installation package (IP) for each of these GVP components. Each component is configured as an Application object in Genesys Management Framework. The Fetching

Module was integrated with the Media Control Platform IP in release 8.1.2, and in all subsequent releases is no longer in a separate IP.

Voice Platform Solution Components

The GVP components integrate with other Genesys Suite components to extend the features and functionality of the voice platform, thereby increasing the flexibility of your deployment.

The Voice Platform Solution is Genesys Voice Platform and three other components:

[+] SIP Server

SIP Server is a T-Server for IP environments in which Genesys T-Lib applications such as Universal Routing Server, Outbound, and Agent Desktop deliver services in SIP environments. SIP Server is a critical integration point for GVP components that interact with network and T-Lib applications.

Interfaces

Unlike other T-Servers, SIP Server operates in environments where there are no switches present. It supports direct interfaces and connectivity to IP agents, voice platforms, gateways, soft switches, and other elements that are used to establish inbound and outbound communication sessions with customers.

SIP Server acts as a SIP B2BUA, and controls the flow of SIP requests and responses between SIP endpoints, performing the switching functions that are normally performed by the PBX or ACD.

Routing

SIP Server can be used in conjunction with an IP PBX or ACD. When used in this way, SIP Server controls the routing and transformation of requests, but does not act as a registrar with which agents communicate. This type of control is normally provided by a CTI link.

For more information about SIP Server and integration with GVP, see the VP Solution 8.1 Integration Guide.

[+] Management Framework

GVP 8.5 is fully integrated into Genesys Management Framework. The GVP component processes are configured as Application objects in the Genesys Configuration Layer, the GVP Application and IVR Profile objects are stored in the Configuration Database, and Configuration Manager is required for bulk provisioning of DNs and Places.

The GVP components interface with Management Framework to obtain the configuration information they need to communicate with other GVP components:

- The Resource Manager obtains configuration information about the SIP resources that it manages for example, the Media Control Platform and the Call Control Platform. The Resource Manager must be aware of these SIP resources and the services they offer.
- The Media Control Platform and Call Control Platform obtain configuration information about SIP proxies in the deployment (the Resource Manager). The call-processing components must be aware of the SIP proxy resources and the services they offer.
- The Supplementary Services Gateway and PSTN Connector obtain configuration information about SIP T-Servers in the deployment (SIP Server). These components rely on SIP Server to obtain access to media processing services offered by the Media Control Platform through the Resource Manager.
- GVP Reporting obtains logging information from the component log sinks, and it is integrated with Management Framework to accumulate summary statistics that are used by Reporting Server.

User Interaction Layer

The User Interaction Layer of Management Framework is the unified Web-based interface that controls applications and solutions. It acts as a manager and administrator for all Genesys components, including GVP components. It provides an interface to the Configuration and Management layers and to otherGenesys solutions.

For more information about the User Interaction Layer, see the Management Framework Architecture Help.

Genesys Administrator

Genesys Administrator is the Web-based GUI that is used to manage all Genesys products, including GVP, with a single user interface. It is part of the Management Framework User Interaction Layer.

Functions

Use Genesys Administrator to access the following functions within the User Interaction Layer:

- Configuration
- Provisioning
- Hierarchical Multi-Tenant configurations and management.
- Management operations (starting or stopping applications)
- Monitoring of current status
- Service Quality (SQ) metrics and latency alarming.
- Installation
- Deployment
- Data collection and logging
- Data management

Genesys Administrator retrieves information about GVP IVR Profiles (voice or call-control applications) and components from the Configuration Database. Therefore, you can use Genesys Administrator as an interface to create, modify, delete, and save GVP information.

It also provides multi-site reporting statistics aggregating reports from multiple independent Reporting Servers, and allowing an administrator to view data that is aggregated across multiple sites.

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

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

For more information about Genesys Administrator, see Genesys Administrator 8.1 Help.

[+] Composer

Genesys Composer is a voice application development tool that is used to develop VoiceXML and CCXML applications. Composer is the preferred tool for customers who write their own applications, but you can use any tool you choose Composer is optional in the VPS.

The Composer GUI enables you to build voice and call-control applications by using drag-and-drop operations.

The integrated Composer development environment simplifies the creation of voice applications. Developers use the Composer authoring tool to build voice applications from a visual call flow editor or a rich XML editor. Applications are compiled and deployed directly to a web application server, and are thenfetched and executed by GVP.

Composer includes a run-time tool that debugs VoiceXML applications in real time, while the developer performs testing by using a SIP phone.

Tip

The runtime debugger, and the code that is created with it, work only with the NGI. If you are using Genesys Studio, you can migrate your CTI applications to GVP 8.5 by using the CTI Connector and the GVPi.

Composer Functions

Composer performs the following functions:

- Creates voice applications by using a visual call-flow editor
- Generates VoiceXML code from the visual call flow
- Provides context-rich Editors for writing and editing VoiceXML, CCXML, and Grammar Extensible Markup Language (GRXML) (speech-recognition grammars) code
- Tests and debugs VoiceXML applications
- Provides project management
- Obtains version history and team support
- Creates CTI applications by using SIP Server (non-CTIC) for NGI
- Creates CTI applications by using CTIC for NGI

Tip

Genesys Composer supports the testing and debugging of VoiceXML applications that are written by using third-party development tools. For applications that are written for the Legacy GVP interpreter (GVPi), use Genesys Studio. For more

> information about Genesys Studio, see the GVP 7.6 Studio Deployment Guide.

SIP Server and Genesys Management Framework, including the User Interaction Layer, are required to create an overall VPS solution.

Third-Party Software

In addition to the Squid Caching Proxy described on page 47, GVP either requires or optionally supports the use of additional third-party software in the VPS.

This section describes the following third-party software that is used in conjunction with GVP:

- Automatic Speech Recognition
- Text-to-Speech
- Reporting Database
- Web Server

For information about other third-party software requirements for GVP, and for details about the supported versions of the third-party software, see Prerequisites.

Automatic Speech Recognition

GVP uses MRCP speech-recognition technology to incorporate automatic speech recognition for use in voice applications. Using ASR in a GVP deployment is optional.

Text-to-Speech

GVP uses MRCP speech synthesis technology to incorporate text-to-speech for use in voice applications.

Using TTS in a GVP deployment is optional.

Reporting Database

VP Reporting Server works with a relational database-management system (RDBMS) and currently works with Microsoft SQL Server and Oracle Server. The RDBMS provides storage and queries the data that is in the relational database. The Reporting Server is responsible for controlling the RDBMS and providing reporting web services on top of the relational database.

To store GVP usage information for later analysis, Reporting Server database in your GVP deployment is mandatory.

Web Server

The GVP voice and call-control applications reside on a web server that the GVP interpreters access on every call; by using either standard HTTP or HTTPS.GVP supports interactions with multiple web servers. If voice or call-control applications reside on separate web servers, these web servers can be located on a web farm architecture in a local or remote network configuration.

Communication between the web server and GVP is analogous to the desktop web browser model. In a standard Web-based application, desktop browsers make requests to an application server to provide HTML so that they can render the Web-based application. The browser renders a web page, and establishes linksto other pages on the Web. When you click a link, the browser issues a request to the designated URL, which results in the retrieval and rendering of another web page. When the page or its contents change, the next request from any browser retrieves the changed page.

Requests and information exchanged on GVP are handled in a similar fashion, but the markup languages are CCXML and VoiceXML instead of HTML. The HTTP Client requests pages from web servers.

Call-control and voice applications can be developed by using Active Server Pages (ASP) or Java Server Pages (JSP), manually by using CCXML and VoiceXML(rather than being generated by the ASP or JSP pages), or by using Genesys Composer.

The Call Control Platform and Media Control Platform interpreters parse the CCXML or VoiceXML to affect:

- Call handling (answering, bridging, and disconnecting calls).
- Media management (playing greetings, prompts, and messages by using cached voice files and TTS).
- Caller input (collecting touch-tone digits and performing speech recognition).

The Media Control Platform and Call Control Platform enable VoiceXML and CCXML applications to drive an interaction with a caller in the same way that the desktop web browser would interact with an application server to render a screen, and to react to keyboard or mouse input. As with the desktop browser, depending on the page s cache control headers, any changes to the call-control or voice application on the application server generally becomes effective the next time a page is requested.

Communication Within GVP

The VPS is a complex solution that requires GVP to handle various types of communications.

Communication Protocols

As Figure: Genesys Voice Platform Solution Architecture shows, GVP uses the following communication protocols:

- **SIP** For call-control messaging between the Resource Manager and SIP Server, and for resource-management messaging between the Resource Manager and GVP resource components.
- **HTTP** For fetch communications among the NGI/CCXMLI, Fetching Module, and Squid Caching Proxy, and between the Fetching Module and web application server. HTTP is also used for communication between the Reporting Clients and Reporting Server, between the Reporting Web Services and Genesys Administrator, and between the Supplementary Services Gateway and third-party Trigger Application.
- **MSML** For media services communications between SIP Server and the Media Server through the Resource Manager.
- **MRCP** For managing speech services between the Media Control Platform and the ASR or TTS speech engines. GVP 8.5 supports MRCPv1 over Real-time Streaming Protocol (RTSP) and MRCPv2 over SIP.
- **RTP** For delivering media (audio and video data) between the Media Control Platform and the external media gateway, and between the Media Control Platform and the speech engines.
- **IVR XML** For accessibility to CTI functionality through the IVR Server to the CTI Connector, and the CTI connector to the GVP components.
- **GED-125** For interacting with Cisco ICM.

For the exact specifications that GVP supports, see Specifications and Standards.

Tip

The GVP components support IPv6 communications with compatible devices and networks. For information about how to enable IPv6 support in the GVP component Applications, see "Enabling IPv6

> Communication" in "Chapter 3: Configuring Common
>
> Features" in the GVP 8.5 User's Guide.

Secure Communications

GVP 8.5 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 Managerand Media Control Platform and Call Control Platform resources.

GVP supports TLSv1.

- Secure HTTP (HTTPS) HTTP over Secure Socket Layer (SSL) and TLS version 1 (TLSv1), for fetch communications among the NGI/GVPi/CCXMLI, Fetching Module, and Squid caching proxy, and between the Fetching Module and web application server. The Reporting Server supports HTTPS for receiving and responding to authenticated reporting requests from Genesys Administrator or an HTTP Client. The Supplementary Services Gateway supports HTTPS for requests from the third-party Trigger Application.

GVP supports SSL version 2 (SSLv2), SSL version 3 (SSLv3), SSL version 23 (SSLv23), and TLSv1.

- 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 Application object for each component. Therefore, basic security is implemented without having to configure it. 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 the section about enabling secure communications in the GVP 8.5 User's Guide.

Tip

Before you use HTTPS to reference grammars, ensure that your ASR engine supports it.

IPv6 Communications

GVP components support IPv6 communications with compatible devices and networks. The dual-stack functionality supports scenarios where one call leg is onIPv4 and the other, IPv6.

Notes: While GVP 8.1.5 is IPv6 ready, other Genesys and third party interfaces are not. Investing in GVP assures that as vendors and other Genesys products adopt IPv6, GVP is ready. In addition, GVP 8.1.5 is dual-mode enabled, so it preserves compatibility with existing supported interfaces that use IPv4 only.

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

Local Port Ranges

Users can configure a TCP or TLS local port range by using the sip.tcp.portrange and sip.tls.portrange configuration parameters, respectively. This parameter can be configured in all four Resource Manager service configuration sections: monitor, proxy, registrar, subscription.

These port range configuration option values are empty, by default. If they are not configured, the operating system selects the local port.

Tip

The CTI Connector supports SIP IPv6. However, Cisco ICM supports IPv4 only. Therefore, the current ICM implementation in CTI Connector supports IPv4 only.

IPv6 in Initial SDP Offer

A default IP version can be defined for SDP offers. The `[mpc] preferredipinterface` configuration option defines this behavior. If the default IP version is not specified, IPv4 is used as the default version. This configuration defines the IP version that will be used when the Media Control Platform generates an SDP offer.

> Tip
>
> The IP version that is used for SDP negotiation (IPv4 or IPv6) is independent of the version that is used for SIP transport.

SSG Connectivity to SIP Server

The Supplementary Services Gateway supports connectivity to SIP Servers that are running on IPv6 networks. The enable-ipv6 configuration option in the Common section of the Supplementary Services Gateway Application enables IPv6 support.

The Supplementary Services Gateway sends the ip-version parameter as a transport parameter to T-Lib and is required to define the preferred protocol version in the DNS of the SIP Server.

Web Services Access

The Reporting Server supports web services access through a network interface that is configured to use an IPv6 address. Genesys Administrator can use this interface to the Reporting Server and detect the database mode to determine which dashboards and reports to display and hide.

If you are planning to use IPv6 in your environment, you must configure the GVP components accordingly. For information about how to enable IPv6 support in the component Applications, see Chapter 3 in the GVP 8.5 User's Guide.

SNMP Monitoring

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

The Master Agent handles all queries from the Management Data GUI and any Network Management Systems (NMS), and sends Agent X queries to the appropriate subagent (in other words, to the appropriate GVP process).

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. For more information about the GVP MIBs, see the GVP 8.5 SNMP MIB Reference.

High Availability and Scalability

GVP 8.5.x supports GVP High Availability in three ways:

- by configuring the Resource Manager for HA.
- by configuring the Reporting Server for HA.
- by combining the call-processing components into resource groups.

Different Resource Manager configurations may require different approaches. See Resource Manager High Availability Solutions.

Resource Manager

The Resource Manager controls access and routing to all resources in a GVP 8.5 deployment.

The Resource Manager is the first element to process requests for services, and it interacts with the Configuration Server to determine the Interactive Voice Recognition (IVR) Profile, Voice Extensible Markup Language (VoiceXML), Call Control Extensible Markup Language (CCXML), Announcement, and Conference application, resource, and service profile required to deliver the service. It then pushes the profile to a component that can deliver the service, such as the Media Control Platform or Call Control Platform, or CTI Connector.

Hierarchical Multi-Tenant Configurations

The Resource Manager also supports Hierarchical Multi-Tenant (HMT) configurations for service providers, enabling them to apportion a select number of inbound ports for each customer, which provides greater flexibility when enforcing policies during service selection. For more information about HMT policy enforcement, see Service Parameters.

This section provides an overview of the following topics:

- Resource Manager Roles
- Resource Manager Functions

Resource Manager Roles

The Resource Manager performs the following key roles in a GVP deployment SIP Proxy, SIP Registrar, and SIP Notifier.

SIP Proxy

The Resource Manager resides between all SIP resources within the GVP system architecture. It acts as a proxy for SIP traffic between any two SIP components.

As a SIP proxy, the Resource Manager is the interface to a collection of media-processing resources, such as the Media Control Platform, the Call Control Platform, audio and video conferencing, and other resources. SIP devices and VoiceXML or CCXML applications can then make use of media-centric services through the proxy, without information about the actual location of these resources or how to manage various routing decisions:

- External clients, such as media gateways or soft switches, can access GVP services without knowing the topology or other details of the resource fulfilling the request.
- Internal media resources can access the services offered by other components without knowing the location or status of the resource that are fulfilling the request.

SIP Registrar

The Resource Manager acts as a registrar for GVP resources; however, it accepts registration only from those resources that are added to the Connections section of the Resource Manager Application object. Registration occurs through SIP REGISTER messages; therefore, GVP supports transparent relocation ofcall-processing components.

Currently, the Media Control Platform, Call Control Platform, and CTI Connector do not register with the Resource Manager at startup. The Resource Manager detects instances of these components through configuration information that is retrieved from the Configuration Database.

If the Media Control Platform Resource group has been configured for monitoring, the Resource Manager monitors resource health by using SIP OPTIONS messages. For example, to determine whether the resources in the group are alive, the Resource Manager periodically sends SIP OPTIONS messages to each Media Control Platform resource in the group. If the Resource Manager receives a 200 OK response, the resources are considered alive.

SIP Notifier

The Resource Manager acts as a notifier, accepting SIP SUBSCRIBE requests from SIP Server and maintaining multiple independent subscriptions for the same or different SIP devices. The subscription notices are targeted for the tenants that are managed by the Resource Manager. In this role, the Resource Manager periodically generates SIP NOTIFY requests to subscribers (or tenants) about port usage and the number of available ports.

The Resource Manager supports multi-tenancy by sending notifications that contain the tenant name and the current status (in- or out-of-service) of the Media Control Platform (active or passive) that is associated with the tenant. For information about how the Resource Manager provides resource status information, see Notification of Resource Status.

Resource Manager Functions

The Resource Manager performs the following functions:

- **Resource management**—The Resource Manager allocates and monitors SIP resources to maintain a current status of the resources within a GVP deployment. The Resource Manager provides load balancing and high availability for each resource type, as the workload is evenly distributed among resources of the same type. These processes ensure that new, incoming services are not interrupted when a resource is unavailable. (See also the description of the resource selection, and High Availability and Scalability.)
- **Session management**—The Resource Manager combines two logical functions of session management:
    - **Physical resource management**—The Resource Manager monitors the status of the various GVP resources and, based on request-for-service and

capability mapping, routes to other resources that offer a particular set of capabilities or services.
    - **Logical service management**—The Resource Manager applies high-level application and business logic to select the service that is delivered and the parameters that are applied. This means that the resource to fulfill the service does not need to be specified in advance.

In this way, the Resource Manager provides session management functions to handle logical call sessions, individual calls within a logical session, and the lifetime and coordination of call legs within a call session.

- **Service selection**—When a call session arrives at the Resource Manager, the Resource Manager maps the call to an IVR Profile and, if applicable, to a tenant, and selects a service for the request.

Application Selection

There are various ways in which the Resource Manager determines which IVR Profile to execute. GVP most commonly uses one of the following methods:

- Dialed Number Identification Service (DNIS) mapped to the IVR Profile GVP uses the DNIS to identify which application to run. In this scenario, the incoming call corresponds directly to the DNIS.
- Voice application specified as a treatment within a call Another Genesys component (for example, the CTI Connector) acts as a master and executes a number of slave applications on GVP. When a service is required as part of a call flow, the voice application invokes a treatment on GVP. In this scenario, the voice service is invoked as part of the master call flow that the master application executes.

> Tip
>
> For a description of how the Resource Manager executes IVR Profiles when the CTI Connector is deployed, see How the CTI Connector Works.

Tenant Selection

When the platform administrator segregates services into a multi-tiered hierarchy, the Resource Manager also identifies the tenant for which a request is intended. The IVR Profile, policy enforcement, and service parameters are determined by the tenant that is associated with the request. In an HMT environment, when a tenant is selected, the policies enforced, and application and service parameters associated with that tenant, also affect the child tenants within that tenant object.

Service Selection

After the Resource Manager has determined the IVR Profile for a session, it identifies the service type and the service prerequisites for each call leg.

The Resource Manager supports the Differentiated Services (DS) Field for outbound SIP message packets for UDP, TCP, and TLS transport protocols. The DS Field value, which prioritizes the type-of-service (ToS), is configured in the `sip.transport.[n].tos` parameter. For a complete list of supported TOS standard values, see the Genesys Voice Platform 8.5 User's Guide.

Tip

A separate set of SIP transports are used for processing SUBSCRIBE requests (for which the Resource Manager acts as a SIP User Agent). However, subscribers can also use the Resource Manager proxy transport for subscriptions.

Service Parameters

For each type of service within an IVR Profile, you can configure a set of service parameters that the Resource Manager forwards to the VoiceXML or CCXML application to affect the way that the application is executed. For example, you can configure the default languages for the VoiceXML services for voiceapplications.

- **Policy enforcement:** For each IVR Profile and, if applicable, for each tenant, you can configure policies such as usage limits, dialing rules, and service capabilities. The Resource Manager enforces policies by imposing them on the VoiceXML or CCXML application to determine whether or not to accept a SIPsession. If the session is accepted, the Resource Manager locates a resource to handle it. The Resource Manager also enforces policies related to how a VoiceXML or CCXML application uses a resource.
    - **Multi-tenant policy enforcement:** For multiple tenants, you can configure the Resource Manager to apply and enforce policies in a hierarchical

manner. HMT enables you (a service provider or parent tenant) to allocate portions of its inbound ports to each reseller (or child tenant). The reseller can, in turn allocate ports to a number of child tenants within its tenant object. When tenant policies are enforced at the child tenant level, the policies are propagated to all other child tenants within that child tenant object. For more information about how the Resource Manager enforces tenantpolicies in a multi-tenant environment, see HMT Policy Enforcement.

- **Service request modification:** Before the Resource Manager forwards a request to a resource that can handle the mapped service, it can modify the SIPrequest to add, delete, or modify the SIP parameters. You can configure this user-defined information on a per-service/per-application basis.

Tip

Definitions of the service parameters that are required for a service within a voice or call-control application are specific to the component that is providing the service. The Resource Manager merely provides the framework within which an application defines the parameters that influence the way an application is executed.

- **Resource selection:** After the Resource Manager has identified an IVR Profile and service type, it identifies a Resource Group that can provide the service. Then, on the basis of the load-balancing scheme for the group and the status of individual physical resources in the group, it allocates the request to a particular physical resource.
    - **Resource selection with geo-location information:** When the Resource Manager receives a request with geo-location information from a gateway resource (SIP Server), it checks the Resource Groups to determine if the geo-location parameter that is configured for the group matches the geo-location in the request. If it finds a match, the Resource Manager routes the call to the group based on port availability, preference and other criteria.

      For more information about how the Resource Manager processes geo-location information during resource selection, see Notification of Resource Status.
    - **Resource selection for outbound campaigns:** For outbound-call campaigns, the Resource Manager can predict the ratio of agent calls to customer calls b using a prediction factor (factor-P) parameter and, when

there are multiple Media Control Platforms in a deployment, it can distribute calls based on the maximum number of calls and free ports for a particular campaign.

Requests for conference services are not handled in the same manner as requests for other services, because the Resource Manager must route requests for a particular conference ID to the same conference resource, even if it is from a different Resource Manager session. For more information, see Resource Selection for Conference Services.

For service types other than conferencing, there is no special correlation required for requests from different Resource Manager sessions.

- **Call-data reporting:** When data collection and logging events occur, the Resource Manager sends these log events to the Reporting Server. For more information, see CDR Reporting.

  For the CTI and PSTN Connectors, the Resource Manager submits Component Arrival and Peak data for Historical reporting services.

  For a description of how Resource Manager selects resources when the CTI Connector is deployed, see the section CTI Connector.

  For information about installing the Resource Manager with a basic configuration, see Installing GVP.
- **Detection and monitoring of Recording Servers and Clients:** To provide and facilitate GVP call recording services, when the Media Control Platform is acting as a Recording Client and to support third-party recording devices. See also the section *Recording Servers and Clients*, Chapter 3 in the Genesys Media Server 8.5 Deployment Guide.

  Return to GVP Architecture—Components

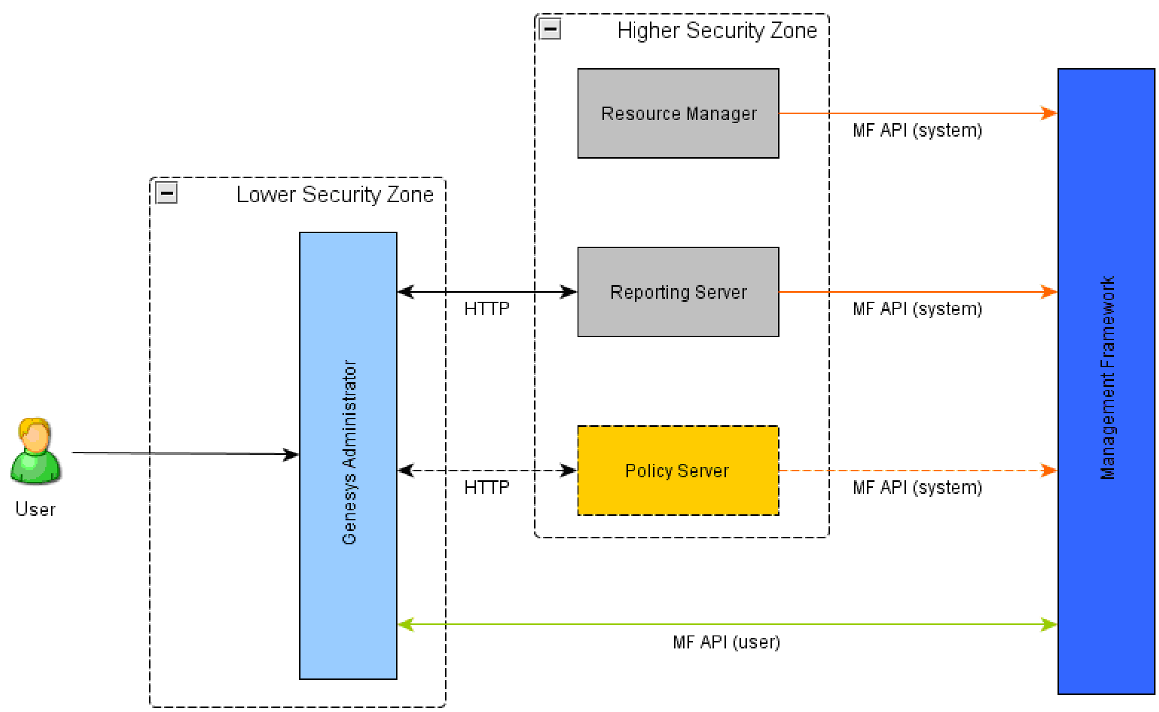
Policy Server

The Policy Server component validates and resolves GVP-specific business rules (the policies that are enforced by the Resource Manager) and provides this information to Genesys Administrator in response to HTTP queries. It is a stand-alone Java process that exposes an HTTP interface through which it connects to Management Framework. The read permissions that are granted when a user logs into Genesys Administrator determine which Management Framework objects are accessible.

Secure Architecture

Within the GVP system architecture, the Policy Server resides behind Genesys Administrator and reads data from Management Framework. Genesys Administrator is considered to be in a lower security zone, because it can be deployed on a WAN or opened up to the Internet. If Genesys Administrator is compromised, the attacker would have access equal to the logged-in user only. In this way, access to a platform-wide view of a the GVP environment can be secured and managed.

Policy Server is designed as a separate component rather than a module within Genesys Administrator, so that it can be placed in a higher security zone, as in the figure below.



Policy Server Architecture

Policy Server runs as a system service and has permission to view all objects in the deployment. The current release of Policy Server is used by Genesys Administrator only, however, it has been designed for use by other components, user interfaces, or third-party tools in the future.

Policy Server Functions

Policy Server performs the following functions:

- Resolves Resource Manager policies within the a multi-tenant hierarchy and IVR Profiles.
- Manages DID ranges (newly created and existing) within the deployment by checking for overlaps.
- Provides a service that returns a Web Application Description Language (WADL) document that describes all of the Policy Server services, including information about the server instance.

For more information about how Policy Server performs its functions, see How the Policy Server Works.

Policy Server Interfaces

Policy Server interacts with components within the architecture by using two interfaces an HTTP interface to receive incoming policy queries and an interface with Management Framework to retrieve configuration and provisioning information.

CTI Connector

The CTI Connector supports two modes of CTI deployments Genesys CTI and Cisco CTI integration. A single instance of CTI Connector can support either Genesys CTI or Cisco CTI integration, which can be selected by the user during installation.

In both the deployment, the CTI Connector offers functions, such as the ability to obtain call related information (for example, ANI, DNIS), read or attach call data, play treatments to the caller, and transfer the call to the agent.

For more information about the role of the CTI Connector in the VPS and functionality offered by Management Framework, see the Genesys Voice Platform Solution 8.1 Integration Guide.

This section provides an overview of:

- CTI Connector Role
- CTI Connector Functions

CTI Connector Role

The CTI Connector acts as a SIP B2B UA to provide a SIP interface to the GVP components and it communicates with CTI by using the following protocols and interfaces:

- ML over TCP/IP with Genesys IVR Server
- GED-125 interface over TCP/IP with Cisco ICM.

The CTI Connector acts as a border element within GVP, interfacing with the CTI network on one side, and through the Resource Manager, interacts with theMedia Control Platform on the other side.

Third-party components do not use the CTI Connector directly. The CTI Connector supports both NGI and GVPi when it is integrated with IVR Server, however, when CTI Connector is integrated with Cisco ICM, only NGI is supported.

CTI Connector Functions

This section describes the various CTI Connector functions when integrated with IVR Server and Cisco ICM.

Resource Selection

CTI sessions receive special handling from the Resource Manager. For example, requests from the Media Control Platform are sent to the same CTI Connector that was used to establish the call. In addition, when the CTI Connector attempts a transfer, the Resource Manager sends the request to the same gatewayfrom which the call originated. The behavior is described as follows:

- If an incoming call is received from a gateway, and the use-cti configuration option in the Gateway group is set to `1`, the Resource Manager identifies the CTI Connector resource group (not an IVR Profile) to provide the service.
- If an incoming call is from a gateway and if `use-cti = 2`, then Resource Manager maps an IVR Profile, extracts the CTI service parameters that are configured in the profile and forwards these parameters in the INVITE message that is sent to the CTI Connector.
- If `use-cti = 0`, the Resource Manager does not treat the incoming call as a CTI call and proceeds with the DNIS-to-IVR Profile mapping.

SIP Back-to-Back User Agent

The CTI Connector acts as a SIP B2BUA. It remains in the call signaling path, and receives inbound calls through the Resource Manager or receive soutbound-call requests (transfers) from the Media Control Platform (through the Resource Manager). The CTI Connector intercepts SIP messages that are intended for itself, and acts as a pass-through for SIP messages that are intended for other SIP endpoints in the call.

Call Treatments

Call handling is determined by the interaction between the CTI Connector and the ICM framework. Treatments are used to start and control external applications. These applications then process calls that return the data that is used to route the call. For example, if the ICM framework specifies a particular treatment for a call, the CTI Connector can translate that treatment into a request for a specific Media Control Platform service.

IVR Server Integration

- The CTI Connector can also route calls to, and receive instructions from, Universal Routing Server (URS). The CTI Connector supports the following URS treatments for both the NGI and the GVPi:
    - Play Application
        - For NGI, the specific VXML application to be played can be mentioned in the main VXML application; it is then passed in the INVITE to RM/MCP as the APP_URL.
        - For GVPi, used to invoke specific branching from the IVR script.
    - Play Announcement Used to play an announcement for the caller.
    - Play Announce and Collect Digits Used to play an announcement for and collect digits from the caller.
    - Music Used to play a `.vox` or `.wav` file.

Cisco ICM Integration

- The CTI Connector supports the following treatment for NGI:
    - Script Execution Used to invoke specific branching from the IVR application, based on the script ID that is received.

Transfers for CTI Connector with IVR Server

The CTI Connector supports three types of transfers:

- Blind transfers through the GVP platform.
- Blind transfers through the CTI framework (using OneStepXfer)
- Bridge transfers through the GVP platform

Tip

CTI transfers are supported when the CTI Connector is deployed in behind-the-switch mode only.

Blind transfers can occur in one of two ways:

- **Through the CTI framework**—Used in VoIP and TDM environments, but blind transfers only and supported when IVR Server is in behind mode (behind-the-switch) only.
- **Through GVP**—The CTI Connector is acting as a SIP B2BUA. This transfer is supported when IVR Server is in any mode (in-front, behind, or network).

For information about how the CTI Connector can be configured to interact with IVR Server, see the Voice Platform Solution 8.1 Integration Guide.

Transfers for CTI Connector with Cisco ICM

GVP performs blind and bridge transfers, based on the mode of operation, as follows:

- In SCI mode, it performs a blind or bridge transfer, based on the IVR Profile parameter. The CTI Connector checks the IVR Profile scti.icm.enableBridgeXfer configuration option value, to identify the type of transfer. If this option is enabled, the CTI Connector uses the BRIDGE transfer; if not enabled the CTI Connector uses the BLIND transfer.
- In CRI mode, it performs a blind or bridge transfer, based on the VoiceXML application

Tip

GVP supports three types of transfers: blind, bridge, and consultation. However, CTI Connector supports the two: blind and bridge transfers only.

For more information about blind and bridge transfers, see Transfers.

PSTN Connector

The PSTN Connector is a stand-alone component that provides connectivity to traditional telephony networks and equipment, such as a private branch exchange (PBX) or automatic call distribution (ACD). For existing deployments that use Dialogic TDM cards, the PSTN Connector provides seamless integration and migration to the IP-based GVP 8.5 architecture.

The PSTN Connector supports inbound and outbound calling by acting as a border element, interfacing with a PSTN cloud or PBX or ACD on one side, and through SIP Server, interacting with the Media Control Platform through the Resource Manager on the other.

This section provides an overview of the following topics:

- PSTN Connector Roles
- PSTN Connector Functions
- PSTN Connector Interfaces
- PSTN Connector Supported Transfers

PSTN Connector Roles

The PSTN Connector acts as a media gateway by using Dialogic boards to interface with the TDM side of the network and translate TDM calls to SIP calls so they can be handled by SIP Server and the GVP components.

PSTN Connector Functions

PSTN Connector functions are primarily based on GVP 7.x Voice Communication Server (VCS), but with a few enhancements. The VoIP interface is compliant with RFC 3261 for SIP session control and with RFC 3550 for packetization and control of RTP packets.

The PSTN Connector performs the following functions:

- Captures and transmits DTMF tones (using Dialogic technology) to and from the TDM networks.
- Receives and controls ISDN and non-ISDN calls from TDM networks.
- Detects and emits DTMF tones to and from the Media Server over VoIP networks
- Converts TDM signals and media to SIP messages and RTP over VoIP networks.
- Works with the Media Server to provide media playback and buffer management.
- Provides Dialogic port management and the ability to re initialize ports that are stuck (by using Genesys Administrator).
- Captures dynamic call, port, and Dialogic board statistics in SNMP MIB tables, and generates traps for critical information or failures.

- Passes Dialogic port numbers on to the CTI Connector in SIP custom headers to facilitate integration with IVR Server.
- Supports User-to-User Information (UUI) messages by using the codeset that is sent from a user to the network to transfer information to another remote user.
- Provides bidirectional port functionality and a strategy for managing glare.
- Supports Call Progress Analysis for outbound calls using PSTN Connector.
- Provides media prefilling.
- Supports features for inbound-call support, such as:
    - Disable ISDN Alerting and Overlap Receive DNIS/ANI.
    - Extracting data such as, Redirecting Number (RN), Presentation and Screening Indicators, Numbering Plan, Numbering Type, Billing Number, and Information Indicator Digits from ISDN Information Elements (IE).
- Supports features for inbound and outbound call support, such as:
    - Disconnect Cause Propagation

For more information about how the PSTN Connector performs its functions, see How the PSTN Connector Works.

PSTN Connector Interfaces

The PSTN Connector provides interfaces to support the following three signaling protocols:

1. Integrated Services Digital Network (ISDN) Comprised of digital telephony and data-transport services offered by regional telephone carriers. ISDN uses digitization to transmit voice, data, text, graphics, music, video, and other source material over existing telephone wires.
2. Robbed-bit signaling (RBS) A specific type of Channel Associated Signaling (CAS), which robs the least significant bit of information in a T1 signal from the channels that carry voice and uses it to transmit framing and clocking information.
3. CAS for channelized E1 lines Commonly used in Latin America, Asia, and Europe and is configured to support channel banks in the network that convert various battery and ground operations on analog lines into signaling bits, which are forwarded over digital lines.

To find out how these signaling protocols interacts with the PSTN Connector, see How the PSTN Connector Works.

PSTN Connector Supported Transfers

The PSTN Connector supports many types of call transfers for both inbound and outbound calls, including the following types of transfers:

- **Dialogic Transfers:** Dialogic blind and bridge transfers are treated the same as any other blind and bridge transfers within GVP. See Transfer Types.

Subscribed Transfer Services

- **Transfer Connect** is an AT&T service which enables subscribers to redirect or transfer a call to another location or target party (TP). The toll-free subscriber that receives and transfers the incoming call is referred to as the redirecting party (RP)--in this case, GVP. This service supports data forwarding for inbound and outbound calls, and in-band and out-of-band transfers.
- **Two B-Channel Transfer** (TBCT) service enables a controller (or subscriber) on a PRI to request the Stored Program Control Switch (SPCS) to connect two independent calls on the controller s interface--in this case, the PSTN Connector. When the SPCS accepts the request, the controller is released and the two calls are connected directly.
- **Release Link Trunk Transfer** (RLT) call transfer accepts calls on two different B-channels, pulls the calls back from the GVP, and bridges them at the switch. It then releases both B-channels for further inbound or outbound calls. RLT works with Nortel DMS-100 and DMS-250 switches on ISDN PRI T1 trunk groups.
- **Explicit Call Transfer** (ECT) enables an ISDN PRI user (in this case, GVP) to send requests to the switch to connect two independent calls on the users interface. The two calls can use the same PRI trunk or different PRI trunks. GVP implements a supplemental ECT service, as defined in EN 300 367 and EN 300 369-1. ECT, and supports the ECT_AUS, ECT_UK, and ECT_NZ variants.
- **SIG Call Transfer** Q Signaling (Q.SIG) is a signaling protocol that uses Remote Operation Support Element (ROSE) encoding and object identifiers to provide various supplementary services, including transfers, call control, and path-replacement. The GVP implementation conforms to the method recommended by the European Telecommunications Standard Institute (ETSI) and is based on the ITU-T Q.931 standard. Although Q.SIG is not technically a subscribed service, GVP must be on a network that supports Q.SIG to access its call control and transfer features. In addition, the following conditions must be met:
    - The two connected calls must have compatible B-channels.
    - Both incoming and outgoing calls from the PSTN Connector must be answered.

To use TBCT, RLT, or ECT you must subscribe to the service and the following conditions must be met:

- The two connected calls must have compatible B-channels.
- One of the two calls must be answered.

- If the other call is outgoing (from PSTN Connector), it can be answered or alerting.
- If the other call is incoming (to PSTN Connector), it must be answered.

For more information about how the PSTN Connector integrates with the PSTN network to utilize these transfer services and signaling protocols, see How the PSTN Connector Works.

Tip

The PSTNC is only available on the GVP 8.1.4 CD, but it functions properly with GVP 8.1.6 and above.

Media Control Platform

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

This section provides an overview of the following topics:

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

Media Control Platform Components

The Media Control Platform is composed of:

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

Tip

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

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

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

- The Media Resource Control Protocol (MRCP) Client, which is a group of libraries that runs in-process in the Media Control Platform, to handle MRCPv1 or MRCPv2 communication with automatic speech recognition (ASR) and text-to-speech (TTS) speech engines.
- The Fetching Module, previously a separate component, is now integrated with the 8.x Media Control Platform and communicates directly with the NGI.

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

Media Control Platform Services

VoiceXML

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

NETANN

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

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

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

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

MSML

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

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

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

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

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

Service Delivery

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

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

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

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

Media Control Platform Functions

The Media Control Platform performs these functions:

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

- Performs ASR and dual tone multi-frequency (DTMF) input handling (barge-in or non-barge-in).
- Streams audio data to an ASR server for speech recognition.
- Reserves ASR and TTS resources at call initiation.
- Transfers calls.
- Sends active speaker notifications to the conference creator.
- Conference calls that use audio and video, and support an unlimited number of participants.
- Performs transcoding from one media codec to another when required for example, by bridging media sessions.
- Logs data and produces metrics.
- Performs Call Progress Detection (CPD) and analysis.
- Provides dual-stack functionality, where one call leg uses IPv4 communication and the other IPv6.

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

Interpreters

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

VoiceXML Interpreters

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

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

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

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

The following subsections briefly describe the GVP 8.5 interpreters, to provide a context for the syntactic and semantic differences between the applications that they support.

NGI

The NGI is the default VoiceXML interpreter for voice applications that are running on GVP 8.x. It was introduced with GVP 8.0 and is built on scalable architecture that leverages multi core and multiprocessor environments. This architecture provides higher levels of performance and positions the NGI for support of evolving standards, such as VoiceXML 3.

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

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

- Entity declaration elements (`<!ENTITY>` elements) in the XML document type declaration are not handled and an `error.semantic` is generated when XML documents that contain these elements are retrieved.
- Namespace declaration attributes (`xmlns` attributes) within an element are not exposed as normal attributes in the exposed DOM object.
- If there is no namespace declaration with a local name in the XML document, the prefix property of the Node object in the exposed DOM.
- If the same local name is redefined with another namespace URI, the document does not treat the re-definition correctly.
- The Attr object's child Nodes property is evaluated to null, instead of the value of the attribute.
- The evaluation of the <data> name variable returns the XML document. In the old implementation, it returned the [object VG_DOM_CLASS] string.

> Tip
>
> You can revert the NGI back to the old behavior by setting the value of the

`data.use_xerces_dom_parser` configuration object in the [vxmli] section of the Media Control Platform Application to true.

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

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

GVPi

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

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

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

CTI Interface

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

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

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

Call Control Platform

The Call Control Platform is an optional GVP component that is required only for configurations that use CCXML.

The Call Control Platform supports the execution of CCXML 1.0 applications in SIP-based environments. Therefore the Call Control Platform services are defined by applications written in CCXML.

CCXML is not a media/dialog language like VoiceXML, so it does not provide any dialog resources on its own. Instead, it supports moving calls around and connecting them to dialog resources. Therefore, the Call Control Platform frequently makes use of Media Control Platform services.

This section provides an overview of the following topics:

- Call Control Platform Components
- Call Control Platform Functions
- Session Data

Call Control Platform Components

The Call Control Platform is composed of a core executable file and the CCXML Interpreter (CCXMLI), which runs in-process in the Call Control Platform.

The Fetching Module, previously a separate component, is integrated with the Call Control Platform and communicates directly with the CCXMLI.

For information about installing the Call Control Platform, with a basic configuration, see Specifications and Standards.

Call Control Platform Functions

The Call Control Platform conforms to the W3C CCXML standard for call control to accept, reject, join, or redirect calls.

The Call Control Platform performs the following functions:

- Initiates conferencing.
- Initiates VoiceXML dialogs on the Media Control Platform through the Resource Manager.
- Creates outbound calls through an IP/Public Switched Telephone Network (PSTN) gateway.
- Applies implicit transcoding through the Media Control Platform Media Server.
- Starts new CCXML sessions when:
  - It receives requests from an incoming call.
  - It receives requests from an HTTP client on its HTTP interface.
  - An existing CCXML session creates a new CCXML session by using `createccxml`.
- Logs data and produces metrics.
- Supports MSML dialog requests.
- Supports IPv6 in SDP.

To receive requests from, and send requests to, the GVP components, the Call Control Platform acts as a SIP back-to-back user agent (B2BUA). The Call Control Platform achieves call control by sending a request to the Resource Manager to acquire access to the resource. When the request is received, the Resource Manager finds the appropriate resource for example, the Media Control Platform or a bridging server and forwards the request.

The Call Control Platform uses the NETANN and Burke Draft SIP messaging standards for requests for service.

An HTTP interface enables external entities to use HTTP POST to initiate a new CCXML session through the create session event I/O processor.

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

Session Data

The Call Control Platform maintains the current, peak, and total count data and exposes it to a GUI for monitoring. Data is captured for the following objects:

- CCXML connections
- CCXML dialogs
- CCXML conferences

- Conference participants
- Bridging server participants
- CCXML sessions

Fetching Module and Squid

Fetching Module Now Integrated

The Fetching Module (the executable file `pwproxy.exe` in earlier versions of GVP) is integrated with the Media and Call Control Platform Installation Packages[IP]). The 8.1.2 and later releases of the Fetching Module support HTTP/1.1-compliant caching and is responsible for fetching VoiceXML and CCXML files, as well as HTTP/HTTPS resources.

The Fetching Module efficiently passes fetch results and other information back to the NGI (on the Media Control Platform host) or CCXMLI (on the Call Control Platform host).

Squid Caching Proxy

The third-party Squid software acts as a caching proxy for the Fetching Module. Like any other caching mechanism, the Squid Caching Proxy caches frequently used files so that fetching a copy of a file needs to occur only once; after that, the file is retrieved from the cache.

In GVP 8.1.2 and above, the Squid proxy is optional to provide more flexibility in the deployment. For example, when it is deployed with a web server, multiple Media Control Platform instances can share the same Squid proxy to optimize caching.

For information about how to install Squid Caching Proxy and the Fetching Module, see Specifications and Standards.

GVP Caching

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

GVP can perform the caching function itself (through the Fetching Module and Squid), or you can add another server a caching appliance, or a web proxy server.

External Caching

External cache servers can be beneficial. For example, if you have a site with 10 GVP servers, and an audio file expires, each server must fetch a new copy of the audio file. If there is an external cache server, fetching a new copy of the audio file occurs only once. Also, the external cache servers typically have very robust cache management tools to purge and refresh content.

Fetching Module Caching

The Fetching Module performs caching, as follows:

1. The 8.1.2 and above Fetching Module itself performs in-memory caching, which is HTTP/1.1-compliant. (GVP 8.1.1 and earlier versions of the Fetching Module are not HTTP/1.1-compliant.)
2. 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.
3. If Squid determines that it cannot serve the content from its cache, it goes to the web server to try to fetch the content.

> **Tip**
>
> It is important that the clocking between the HTTP server and client be synchronized, so that the caching policies such as max-age and max-stale work properly.

The Media and Call Control Platforms support clearing of the Fetching Module in-memory cache at runtime. In Genesys Administrator a custom command (CLEARFMCACHE) can be triggered and sent to the Media or Call Control Platform through Management Framework's CCLib.

For more information about how the Fetching Module performs caching, see Caching.

Page Collector Caching

Page Collector is the caching mechanism within GVPi. It fetches XML pages, determines the document format, and passes the pages on to the appropriate parser (VoiceXML or Transportation Extensible Markup Language (TXML)). The cache consists of two types of files an index file and the actual cached files.

Index File

The index file stores thumbnail data for each cached file in binary format, which facilitates a fast search. The thumbnail data consists of the URL, the local file location, response headers, and the last access time. At process start up, the index file is read into memory. After that, the in-memory image is persisted to the index file at periodic intervals and during shutdown.

At process start up, the index file is read using best effort. If a file is missing or corrupt, the process starts up with whatever data it can retrieve. During an HTTP operation, the index file is searched for the entry. If the entry is not found, the request is sent to the web server. If the entry is found, it is validated (using the RFC 2616 standard) to determine whether the cached entry should be returned to the client or whether a request should be sent to the web server.

Caching Files

Each response that can be cached is stored in a local file. The file location is `<MCP Install_Dir>/data/CnCache/<server>/CACnnnn.tmp`, where `<server>` is the name of the web server (as it appears in the URL) from which the response was received, and *nnnn* is a random number. Each time a response is cached, a corresponding thumbnail entry is created in the index file.

For information about how to use HTTP caching to improve performance, see Optimizing GVP Performance Through HTTP Caching.

MRCP Proxy

The MRCP Proxy can be placed between the Media Control Platforms and the MRCPv1 resources within a GVP deployment. Deploying the MRCP Proxy enables ASR/TTS usage reporting data to be sent to the Reporting Server.

MRCP Proxy Functions

MRCP Proxy manages access and routing to the MRCPv1 resources and performs the following additional functions:

- Manages MRCPv1 resources in the following ways:
    - Routes requests to the supported resources.
    - Provides round-robin load balancing between resources.
    - Monitors the health status of resources.
- Sends ASR and TTS peak usage data to the Reporting Server.
- Provides highly available MRCPv1 services to the Media Control Platform through a warm active standby High Availability (HA) configuration.

MRCP Proxy Interfaces

MRCP Proxy supports three component interfaces:

- MRCP interface To manage speech resource requests. MRCPv1 Requests from clients (Media Control Platform) and requests sent to speech servers (ASR/TTS)are supported through MRCPv1.
- Management Framework interface To integrate with CCIlib and the EMS Logger library to receive configuration information, send logging data, and send and receive status information.
- Operational Reporting interface To integrate with the Operational Reporting API to send peak ASR and TTS usage data for IVR Profiles, tenants, other resources, and the overall deployment to Reporting Server.

In addition, MRCP Proxy supports a User Interface (UI) to integrate with the Genesys Administrator web-based UI.

For more information about how the MRCP Proxy performs its functions, see How the MRCP Proxy Works.

Supplementary Services Gateway

The Supplementary Services Gateway manages the initiation of outbound sessions in GVP 8.5. It provides services for customer applications through the SIP Server to the Resource Manager to establish outbound calls between the caller and the Media Control Platform. This allows the Resource Manager to enforce policies, such as usage limits and dialing or translation rules, or to prevent certain customers from placing outbound calls.

For information about how the Resource Manager enforces policies, see Policy Enforcement.

This section provides an overview of the following topics:

- Trigger Applications
- Supplementary Services Gateway Interfaces
- Supplementary Services Gateway Services
- Supplementary Services Gateway Functions

Trigger Applications

The Supplementary Services Gateway uses the standard HTTP request/response method when communicating with trigger applications (or customer applications), which generally resides at the customer premises. Trigger applications send requests to the Supplementary Services Gateway server to initiate outbound-call sessions. The requests are validated and stored before processing and the Supplementary Services Gateway sends final call results (success or failure) to the trigger applications through notification URLs (which are part of the call initiation requests sent by the trigger application).

Supplementary Services Gateway Interfaces

The Supplementary Services Gateway performs outbound-call initiation through the following interfaces:

- Customer Application interface Enables the Supplementary Services Gateway to receive customer requests and manage the initiation of outbound calls with the help of SIP Server. SIP Server initiates two call legs and bridges them to establish a media path between the Media Control Platform (through the Resource Manager) and the external party.

Through this interface the Supplementary Services Gateway acts as an HTTP Server, collecting HTTP call requests from trigger applications, providing authentication by using HTTPS, validating the input data, and storing it in its external database. If the customer input is not sufficient to make a call or if data is missing, the Supplementary Services Gateway returns an error to the trigger application.

- **T-Lib Client interface**—Enables the Supplementary Services Gateway to batch and initiate requests to SIP Server to establish third-party call control from the Media Control Platform to external parties or destinations.
- **HTTP Client interface**—Enables the Supplementary Services Gateway to post results to a notification URL at the logical conclusion of a call.

Supplementary Services Gateway Services

The Supplementary Services Gateway can establish instances of outbound-call sessions across multiple instances of the Media Control Platform. The interfaces that are used by the Supplementary Services Gateway rely on the Resource Manager to distribute the outbound-call-processing load across multiple Media Control Platforms.

Gateway Services

- In a hosted environment, third-party clients accessing the Supplementary Services Gateway do not require direct access to the Media Control Platform or other GVP components because the Supplementary Services Gateway provides an external interface for outbound-call-processing requests. Multiple Supplementary Services Gateways can reside between private customer networks and GVP.

Tenant Services

- The Supplementary Services Gateway supports outbound-call requests from tenants in the following scenarios:
    - Multiple requests for a VoiceXML application from multiple trigger applications
    - Requests for multiple VoiceXML applications from a trigger application
    - Requests for multiple VoiceXML applications from multiple trigger applications

Supplementary Services Gateway Functions

The Supplementary Services Gateway performs the following functions:

- Outbound-call initiation Outbound calls are initiated and VoiceXML applications provide IVR functions for end users.

- Outbound-call triggers Trigger applications use HTTP POST requests to initiate call triggers for single and bulk requests to generate outbound calls. Initial and reattempt outbound-call triggers are queued and prioritized.
- Batched and queued requests Batches of outbound session creation requests are accepted and executed by using application-specified limits on concurrent port usage and launch rates.
- Persistent call trigger data Call trigger data is stored persistently in the Supplementary Services Gateways external database until ports are available. Storing the data persistently prevents data loss where multiple restarts may occur. Any outbound call that is attempted but not completed when the Supplementary Services Gateway restarts is reinstated as a new call, or removed from database, based on the configuration, when the server is fully operational again.
- **Result notification for requests.** The trigger application is notified by using HTTP URLs when an outbound call succeeds or fails. Notifications are sent after the call has been successfully established, the TTL has expired, or after a call fails the specified number of attempts.
- **Cancellation of outbound requests.** Trigger applications can use HTTP POST or DELETE to cancel requests for calls that are not yet initiated. The Request ID that is returned to the trigger application for the create request and tenant ID is specified in the cancel request.
- **Status of outbound requests.** Trigger applications use HTTP GET or POST requests to obtain the status of a request stored in the Supplementary Services Gateways external database. The Request ID that is returned to the trigger application for the create request and tenant ID is specified in the query request.
- **Call Progress Detection (CPD) results.** The Supplementary Services Gateway can use either a media gateway or the Media Server as a CPD provider. The trigger application can also specify that CPD is not required for a call. The Supplementary Services Gateway, based on the parameters provided in the TA,controls whether CPD is started either with the first media packet received or after the call is connected and can specify whether the IVR should be started for specific CPD results.
- **HTTP access from IPv6 networks.** The Supplementary Services Gateway supports HTTP access from IPv6 networks for all requests.
- **Connectivity to SIP Server on IPv6 networks.** The Supplementary Services Gateway supports connectivity to SIP Server on IPv6 networks for T-Lib activities.
- SNMP MIBs and trap generation are supported in the same way as all other (monitored) GVP components.

For more information about how the Supplementary Services Gateway performs its functions, see How the Supplementary Services Gateway Works.

Reporting Server

The Reporting Server component of GVP provides a comprehensive view of the calls serviced by a GVP deployment. The Reporting Server receives data from the Media Control Platform for VoiceXML applications, from the Call Control Platform for CCXML applications, and from other components involved in servicing a call, such as the Resource Manager.

The Reporting Server is one of the key components of the GVP logging and reporting feature, which is referred to as GVP Reporting.

> In GVP 8.1.2, the Reporting Server is an optional component. See Options to Deploying VP Reporting Server.
>
> Reporting Server 8.1.3 and above are backwards compatible with 8.1.1 and 8.1.2 reporting clients.

This section provides an overview of the following topics:

- GVP Reporting Architecture
- Reporting Server Functions

GVP Reporting Architecture

GVP Reporting uses a client/server architecture. The figure below illustrates the GVP Reporting architecture.

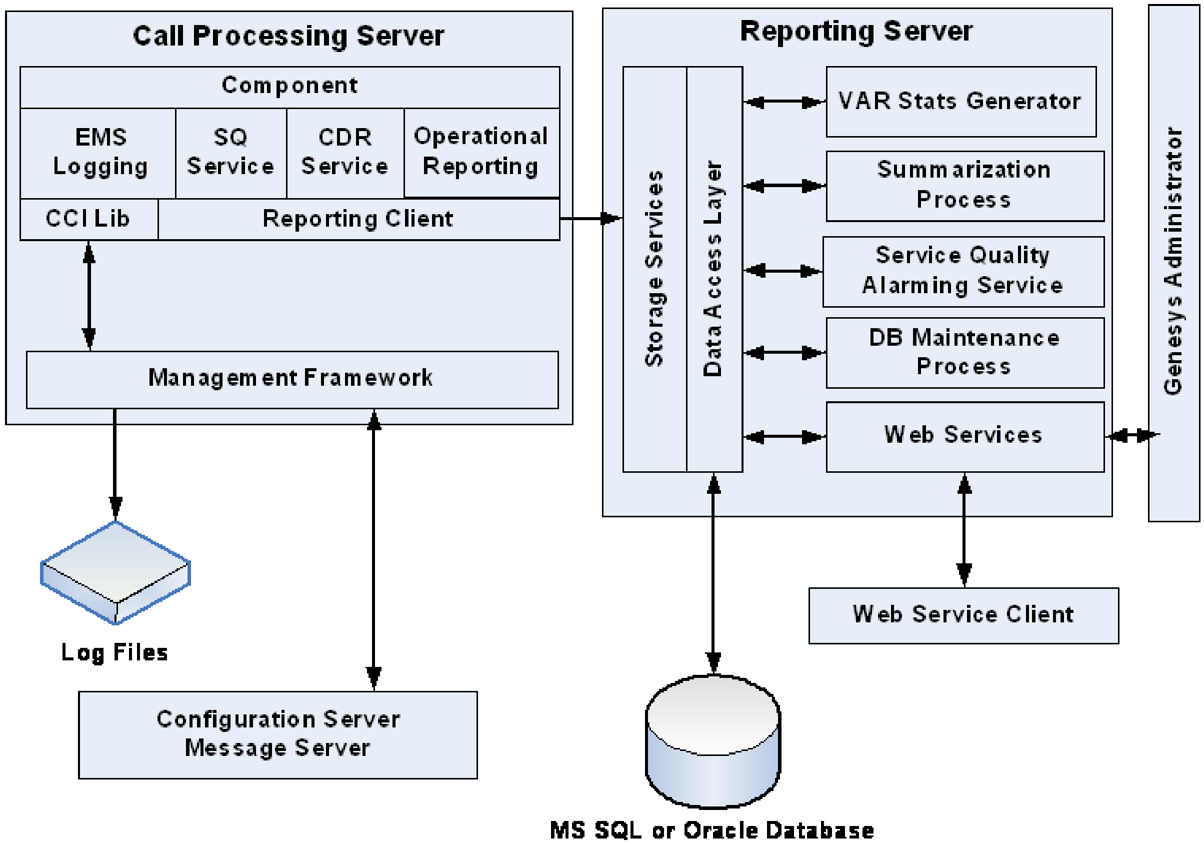


Figure: GVP Reporting Architecture

GVP Logging

- Each component, or GVP Application object, uses a GVP Logging interface to emit call and logging events that are related to component activity and to route the logs to one or more log sinks. For more information, see Logging and Reporting. An API that is used by each call-processing component (Media Control Platform, Call Control Platform, or Resource Manager) submits logging, reporting, service-quality, latency, and Operational Reporting data, such as:

  ### CDR Service

  - Call Detail Record (CDR) Service, through which the components submit CDRs that contain specific call information such as start time, end time, and the IVR Profile and tenant that are associated with the call to the Reporting Server. For more information about CDRs, see CDR Reporting.

OR Service

- Operational Reporting Service, which accumulates call arrival and call peak statistics. (These statistics are applicable for the Resource Manager, Media Control Platform, and Call Control Platform.) For more information about Operational Reporting, see OR Service.

SQ Reporting Service

- Service Quality (SQ) Reporting Service, in which the Media Control Platform generates INFO-level logs that are used by the Reporting Server to generate SQ and latency calculations and call-tracking information. For more information about Service Quality Reporting, see SQA Service.

  Service Quality reports apply to NGi VoiceXML applications, and are found in Genesys Administrator. GVP 8.1.5 and above are NGi-only platforms unless you run MCP 8.1.4 to incorporate support for GVPi 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 to assign success or failure either to individual transactions or to a call as a whole. The `<log>` tag 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 Genesys Administrator.

For more information, see VAR Metrics. For more information about how developers can use Composer to customize the VAR and SQA reporting features, see Customizing SQA and VAR by Using Composer.

Reporting Server

- The Reporting Server stores and summarizes data and statistics that are submitted from the Reporting Clients on the call-processing servers, to provide5-minute, 30-minute, hourly, daily, weekly, and monthly reports.

The Reporting Server manages the following types of data:

- Call-detail records
- Call events
- Call arrival and peak data
- Usage-per-IVR Profile data
- Usage-per-tenant data
- Service-quality (SQ) summary statistics
- SQ latency histograms
- VAR summary statistics

The Reporting Server receives data from the Reporting Clients on a TLS socket, which can be configured by using the options in the messaging section of the Reporting Server Application. See the section "Important Reporting Server Configuration Options" in the GVP 8.1 User's Guide.

For more information about the Reporting Server, see Reporting Server.

One Reporting Server in a GVP deployment can provide adequate call-data management; however, high availability configuration for Reporting Server, which is supported in GVP 8.5 and later releases, requires two Reporting Servers.

For more information about HA for Reporting Server, see High Availability and Scalability.

Customizing SQA and VAR by Using Composer

The SQA and VAR features represent two separate reporting features; the first forces an SQA failure, the second defines a VAR call result. If you are customizing these features by using Composer, keep the following information in mind:

SQA

- If a VoiceXML `<log>` tag is logged with the label `com.genesyslab.quality.failure` or `com.genesyslab.quality.failure`, the session is considered a failed call for SQA.

VAR

- The platform provides an extension to the `<<log>>` tag, using the label= com.genesyslab.var. CallResult label, that allows application developers to specify a VAR Result for a call in the following syntax:

```
<log> label="com.genesyslab.var.CallResult">result[|reason]</log>
```

- `result` must be SUCCESS, FAILED, or UNKNOWN (default). The call result is not case-sensitive and any preceding or trailing white space is ignored.
- `reason` is an optional, textual reason for the call result. The maximum length of the reason is 256 characters and it is not case sensitive. Any text beyond the character limit is truncated.

Reporting Server Functions

The Reporting Server retrieves accurate call related data, VAR reports (call peak and arrival summary data which are not call-specific), OR summaries, and call events. The OR summaries and VAR reports are rolled into higher level summary reports.

The Reporting Server provides:

- Near-real-time call event and CDR data collection and processing information is submitted as soon as it is available or committed.
- Data reliability through guaranteed data delivery policies. These policies ensure that accumulated data is not lost if the Reporting Server or the Reporting Server database are offline for a period of time.
- Efficient organization of the database, by partitioning CDR and Call Event tables so that each partition represents a predetermined period of time (1 hour to 1 day), allowing database operations to occur at times when data accumulation is heavy, such as, when the call rate is high or the data retentionperiod is long.
- Multi-site reporting, by aggregating reports from multiple independent Reporting Servers, allowing administrators to view data that is aggregated across multiple sites.
- Open data access interfaces (through Reporting Web Services) that allow multiple user interfaces, including third-party interfaces, to access call-related data.
- An interface that provides relevant data in various reporting formats to suit many business and operational needs.
- XML reports that correspond to report requests made through Reporting Web Services. The Web Services are exposed over HTTP, and returned in XML format.
- Provides support for queries and traps by communicating with the Genesys SNMP Master Agent through its SNMP interface.

- Retrieves application data from the Supplementary Services Gateway components.

For more information about how GVP Reporting works, see Logging and Reporting.

For information about high availability for Reporting Server, see High Availability and Scalability.

For information about installing the Reporting Server, with a basic configuration, see Specifications and Standards.

Tip

GVP Reporting is unable to track Media Server services use at the tenant level (by tenant or by application). Applications that use URS centric routing have the following reporting issue:

During an MSML call into GVP, if SIP Server changes the `X-Genesys-gsw-ivr-profile-name` or the `X-Genesys-gvp-tenant-id` parameters in the middle of the call (e.g. applying different treatments that use different IVR profiles), the change is not reflected by Resource Manager, Media Control Platform or Reporting Server. All reporting for the call will be against the original IVR profile.

Resource Manager High Availability Solutions

SIP Server with Resource Manager (RM) and GVP with RM offer several options for HA:

- External Load Balancer
- Virtual IP Takeover Solution Windows
- Virtual IP Takeover Solution Linux
- Microsoft NLB Windows

Each option has conditions and limitations.

External Load Balancer

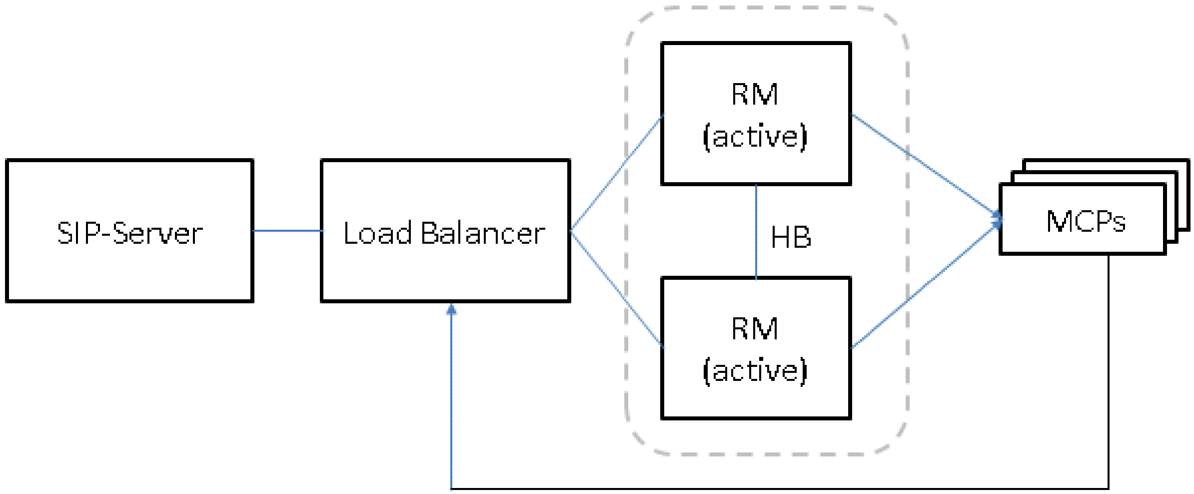


Figure: External Load Balancer (F5/Radware) for RM active-cluster

- This configuration is applicable for RM active-cluster.
- GVP deployment can be Windows or Linux.
- There are separate hosts for SIP-Server, RM and MCPs (preferred).
- SIP-Server goes through the load balancer to RMs.
- RM inserts the Load-Balancer IP address in the Record-Route header so that messages sent within the dialog traverse through the Load-Balancer.
- It is possible, but not preferred, to have RM and MCP on the same host.
- The load balancer must reside on its own host.

Virtual IP Takeover Solution Windows

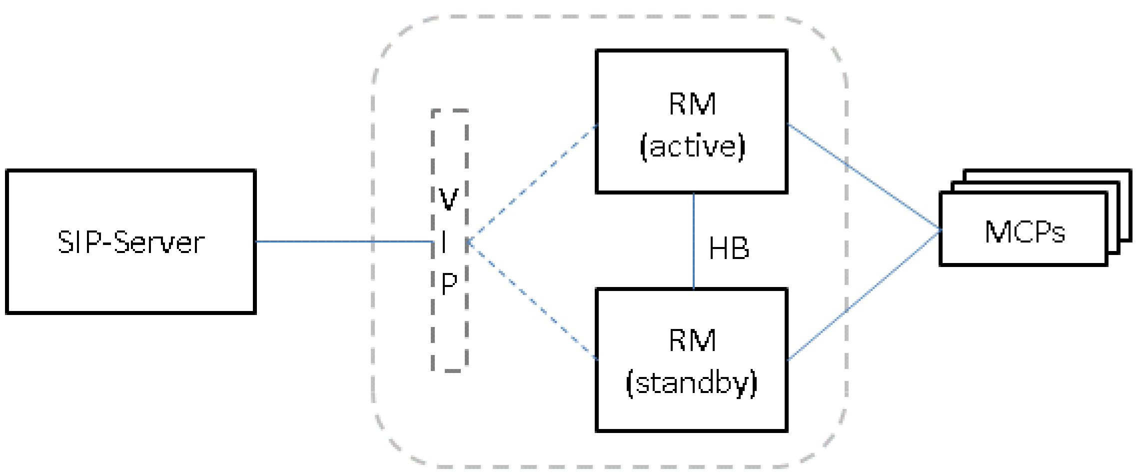


Figure: Virtual IP Takeover Solution (Windows) RM active-standby

- This configuration is applicable for RM active-standby.
- GVP deployment is for Windows.
- Separate hosts are required for SIP-Server and RMs.
- MCPs can be in the same host or in a separate host (preferred) from Resource Manager.
- Virtual IP Takeover is tied to just one NIC.
- Genesys recommends you configure alarm conditions and reaction scripts for handling the failover/switchover condition in this case.
- Due to ARP cache update issues in Windows 2008, this solution uses the third-party utility arping.

Virtual IP Takeover Solution Linux

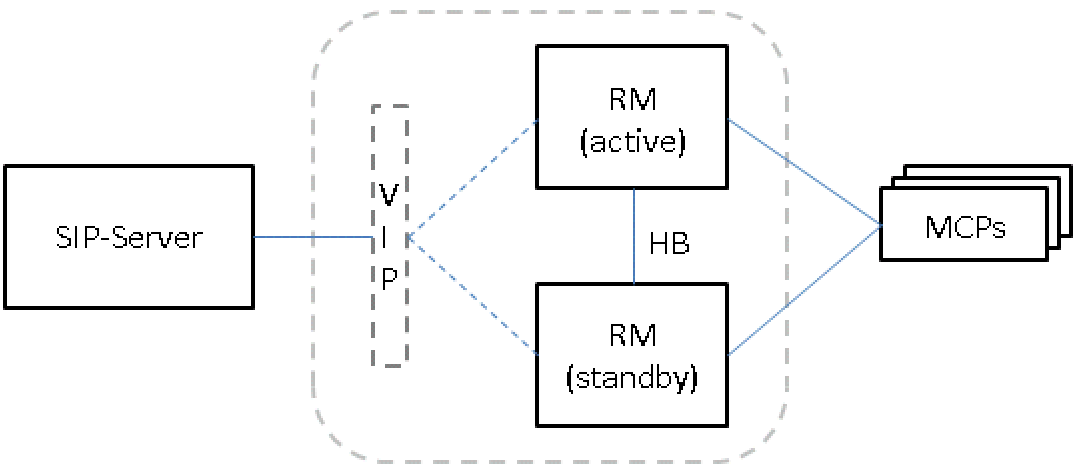


Figure: Virtual IP Takeover Solution (Linux) RM active-standby

- This configuration is applicable for RM active-standby.
- GVP deployment is for Linux.
- Simple IP Takeover for single NIC is supported; if multiple NICs are present in the system, then a Linux bonding driver can be used.
- Separate hosts are required for SIP-Server, RMs and MCPs.
- Genesys recommends that you configure alarm conditions and reaction scripts for handling the failover/switchover condition in this case.

Microsoft NLB Windows

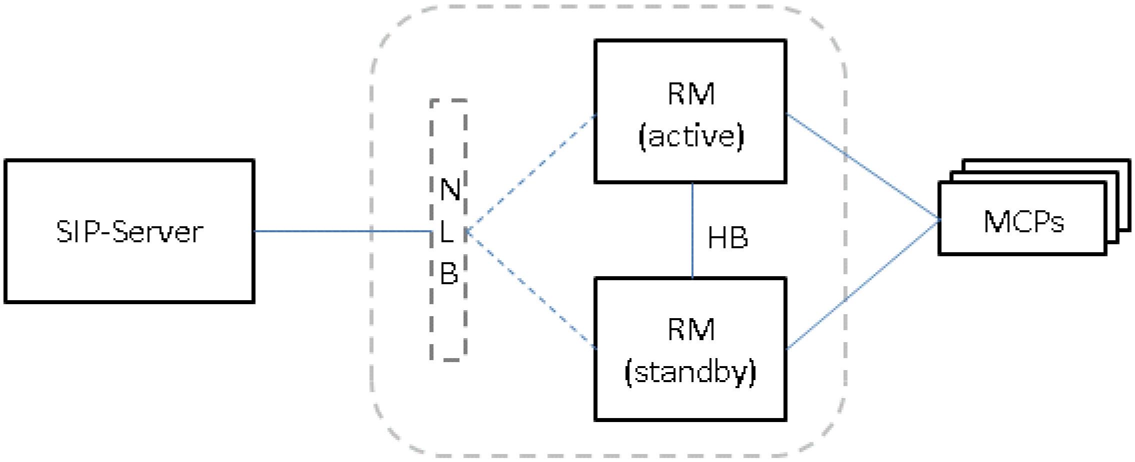


Figure: Microsoft NLB (Windows) RM active-standby

- This configuration is applicable for RM active-standby. GVP deployment is for Windows.
- NLB is used in unicast mode. NLB ensures that other elements can communicate with RMs from outside local network, or within the same subnet.
- NLB configuration requires that RMs be in separate hosts from SIPS or MCPs.
- Each RM host must have multiple NICs; one dedicated to this NLB cluster communications, and the other NIC for non-NLB communications.

HA Using Virtual IP

With this HA solution, multiple hosts share the same virtual IP address, with only one instance actively receiving network traffic. A switchover from active to backup instances in the HA pair can occur with no apparent change in IP address, as far as the SIP dialog is concerned.

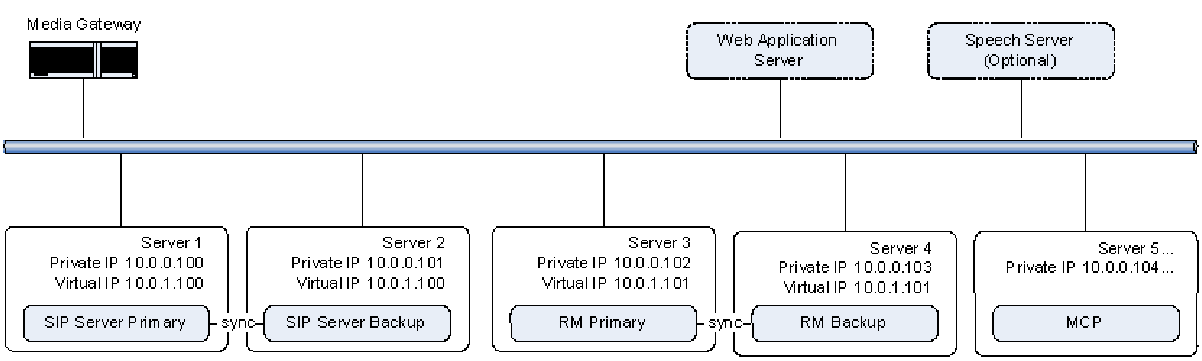


Figure: High Availability Using a Virtual IP Configuration

Feature Limitation

When using Windows Network Load Balancing (NLB) for virtual IP-based HA, only processes running outside the Windows NLB cluster can address that cluster. If SIP Server uses Windows NLB and Resource Manager/Media Control Platform are running on the same machine as SIP Server, then RM/MCP cannot address SIP Server using the cluster address.

With Windows NLB, a local process will always resolve a virtual IP address to the local host. This means that if an MCP process on a particular server tries to contact a failed Resource Manager, Windows NLB will resolve the virtual IP address in the configuration to the local host, and the same local Resource Manager will be contacted, instead of the backup Resource Manager on the backup host.

Sample Configuration

The following task table outlines the basic steps required for an HA deployment on Windows, with the following assumptions:

- This is a two-machine deployment, with one SIP Server and Resource Manager instance co-deployed on each machine.

Task Summary: Configuring HA through Virtual IP for Windows

Objective	Key Procedures and Actions
1. Configure SIP Server instances using Windows NLB.	See the SIP Server Deployment Guide for more information.
2. Configure RM applications	Go to: `Provisioning > Environment > Applications` • Set the Resource Manager option `cluster.ha-mode` to `active-active`.
3. Configure GVP components.	See the GVP 8.5 Deployment Guide for more information.
4. Configure GVP DNs.	Go to: `Provisioning > Switching > Switches` • In the GVP Trunk DN, set the contact and contact-backup options to the two Resource Manager IP addresses.

- For the Voice over service IP Service DN, create two separate DNs with each contact option set to one of the Resource Manager IP addresses.

HA Using an External Load Balancer

This HA method uses an external hardware load balancer to manage clusters of active nodes. The load balancer owns a virtual IP address that is used to forward requests to the active cluster. The load balancer can apply its own load-balancing rules when forwarding the requests.

Use of Active SIP Server Pairs

In some deployments, the customer network does not allow the use of virtual IP addresses. In this case, both SIP Server and Resource Manager pairs can be deployed as active instances. The SIP Server instances are deployed as separate active instances without synchronization. The load balancer will load-balance across the set of active SIP Server hosts. Without the active-backup relationship between the SIP Server instances, SIP Server will lose the state of mid-dialogs if it fails, even though the call will not be immediately dropped.

> **Important**
>
> Genesys does not recommend this HA solution for deployments that require the use of URS routing strategies. If routing is required, Genesys recommends using the load balancer with active-backup detection, so that SIP Server can be deployed as an active-backup HA pair.

Use of Active-Active Clusters for Resource Manager

Resource Manager can be deployed as an active-active cluster, where both instances run together as the active instance, each with a unique IP address. The active pair then synchronizes active session information, so that both instances can correctly route incoming requests.

The next figure shows a sample deployment with an external load balancer, using the following assumptions:

- SIP Server instances are configured as two separate active instances with no synchronization.
- Resource Manager instances are configured as an active-active cluster with synchronization.

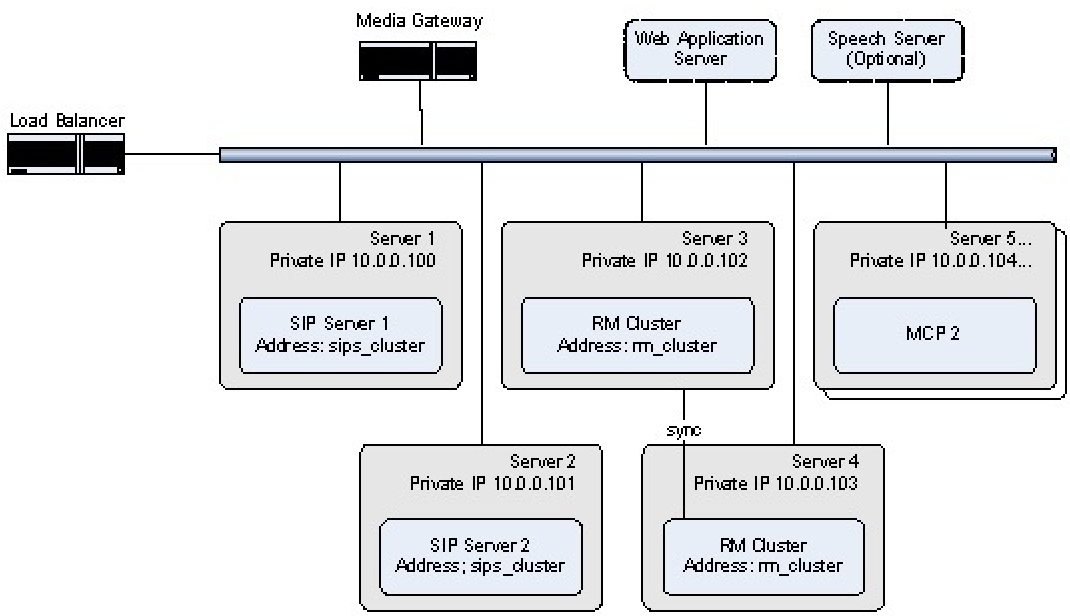


Figure: High Availability Using An External Load Balancer

Resource Manager

GVP 8.5.x supports HA for Resource Manager on both Windows and Linux operating systems.

HA (Windows)

Windows Network Load Balancing (NLB) provides HA for the Resource Manager. You can configure two Resource Managers to run as hot standby, or warm active standby pairs that have a common virtual IP.

Incoming IP traffic is load-balanced by using NLB, in which two Resource Manager servers use a virtual IP number to switch the load to the appropriate server during failover. The network interface cards (NICs) in each Resource Manager host in a NLB cluster are monitored to determine when network errors occur. If any of the NICs encounter an error, the Resource Manager considers the network down, and the load balancing of the incoming IP traffic is adjusted accordingly.

To determine the current status of the Resource Manager at any time, check the traps in the SNMP Manager Trap Console to which the traps are being sent. In the Console, check the most recent trap from each Resource Manager in the HA-pair. If the specific trap ID is 1121, the Resource Manager is active. If the specific trap ID is 1122 the Resource Manager is in standby mode.

In the following example, 5898: Specific trap #1121 trap(v1) received from: 170.56.129.31 at 12/8/2008 6:34:31 PM, the IP address 170.56.129.31 represents the Resource Manager that is active, therefore, the other Resource Manager in the HA-pair is in standby mode.

Scalability

NLB 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.

Multiple clusters of Resource Manager instances that operate largely independently of one another can be deployed to support large-scale deployments, such as those that involve multiple sites. Each Resource Manager cluster manages its own pool of resources.

> Tip
>
> GVP 8.5.x does not support more than two Resource Managers in a cluster.

HA (Linux)

There are two ways to achieve HA for Resource Manager on Linux: by using Simple Virtual IP failover or Bonding Driver failover.

In each of these options, each host in the cluster maintains a static IP address, but all of the hosts share a virtual public IP address that external SIP endpoints use to interact with the Resource Manager hosts in the cluster. If an instance of the Resource Manager fails on any host, the virtual IP address remains valid and provides failover.

When the Bonding Driver failover option is used, two or more network cards are required for the same server and the bonding driver controls the active standby capabilities for the network interfaces.

Resource Groups in HA Environments

When the call-processing components are provisioned in Resource groups, the Resource Manager provides HA for GVP resources in the same way that it normally manages, monitors, and load-balances the resource groups. For example, provided that more than one instance of the Media Control Platform has been provisioned in a VoiceXML resource group, the Media Control Platform service is still available to other VPS components, even if one of the HA provisioned instances is not available.

You can set up your HA Resource Manager environment in one of two ways, depending on the Windows or Linux OS version:

- The **active standby** configuration, in which the active Resource Manager instance only processes SIP requests. Windows 2003, Windows 2008, or RHE Linux 4 or 5 is required.
- The **active active** configuration, in which an external load balancer is used and either of the active nodes can process SIP requests. Windows 2003 or Windows 2008 is required.

For information about configuring the Resource Manager for HA, see Resource Manager High Availability.

MRCP Proxy

GVP 8.5 supports the MRCP Proxy in HA mode to provide highly available MRCPv1 services to the Media Control Platform through a warm active standby HA configuration.

To support the MRCP Proxy in HA mode, the latest versions of Management Framework and LCA must be installed and the Solution Control Server (SCS)Application configured to support HA licenses. For more information about HA licenses for the SCS, see the Framework 8.5 Deployment Guide and the Framework 8.5 Management Layer User's Guide.

Policy Server

GVP 8.5 supports the Policy Server in warm active standby HA mode. The active standby status is determined by the Solution Control Server (SCS), which must be configured to support HA licenses. (See MRCP Proxy, above.) Also, the Policy Server is stateless, therefore, data does not require synchronization.

Reporting Server

GVP 8.5 supports HA for Reporting Server by using a primary/backup paradigm and an Active MQ message store in one of two solutions Segregated Storage or Shared Storage.

Active MQ

The Reporting Server has JMS queues to which data from reporting clients is submitted. The JMS queues implementation used in GVP 8.5 is Active MQ. If the Oracle or Microsoft SQL database is unavailable, but the Reporting Server is still operational, the Active MQ must persist and store the submitted data to the hard disk drive (HDD), thereby ensuring that the data that was submitted to the Reporting Server is not lost if the Reporting Server fails before the database server is restored.

Segregated Solution

In the Segregated Solution, each Reporting Server instance in the cluster uses its own independent Active MQ message store. However, only the server that is designated primary activates its message store. The other independent message store is activated only if the primary Reporting Server fails.

When the Segregated Solution is used, the backup and primary Reporting Servers are configured in Genesys Administrator.

Shared Storage Solution

In the Shared Storage Solution, the Reporting Servers in a cluster share a connection to one Active MQ message store that receives, queues, and dequeues data from Reporting Clients. Only one Reporting Server instance obtains and holds the Active MQ message-store lock.

When the Microsoft Cluster Service (MSCS) is used, two distinct Reporting Servers access a single shared drive. Switch-over to the backup server occurs when the primary server goes down. After the MSCS is configured, the administrative user interface can be used to manage the primary and backup Reporting Servers.

For more information about configuring the Reporting Server for HA, see Reporting Server High Availability.

How GVP Works

- Resource Manager
- Policy Server
- CTI Connector
- PSTN Connector
- Media Control Platform
- Call Control Platform
- Fetching Module
- MRCP Proxy
- Supplementary Services Gateway
- Logging and Reporting

How the Resource Manager Works

The Resource Manager interacts with other components by using three main interfaces: a SIP interface for call handling, the Genesys Management Framework interface for retrieving configuration and provisioning information, and a Reporting Server interface for reporting events related to call processing.

This section describes in more detail what the Resource Manager does with service requests, such as the calls described in GVP Call Flows.

The Resource Manager performs the following functions:

- Session Management
- Service Selection
- Policy Enforcement
- Service-Request Modification
- Managing Resources
- Full Call Recording Requests (from Cisco UCM)

For information about how the Resource Manager reports its activities to the Reporting Server, see Logging and Reporting.

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. This 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 the section about GVP identifiers in the GVP 8.5 User's Guide.
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.
4. In multi-tenant environments, before mapping the session to an IVR Profile, the Resource Manager first checks the resource from which the request originated. If the resource is one that is managed by this specific Resource Manager, and it is a gateway resource, the Resource Manager determines which tenant owns the gateway resource and proceeds to map the IVR Profile. For more information, see Service Selection.
5. The Resource Manager adds the following parameters to the `X-Genesys-GVP-Session-ID` header:
    - `gvp.rm.datanodes` To identify itself as the Resource Manager for the session. In GVP 8.5, it is used mainly to configure cluster information. When Resource Manager is in stand-alone mode, the value is `node-id=1`. When Resource Manager is clustered, either member of the cluster can have a value of `node-id=1|2` (primary) or `node-id=2|1` (secondary), depending on its current status.
    - `gvp.rm.cti-call` To identify a call that is routed through the Computer Telephony Integration Connector (CTIC). In GVP 8.5, the value of this parameter is set to `1` to invoke a CTI service.
    - `gvp.rm.tenant-id` To identify the voice or call-control application under which the session is executed. 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 the section about GVP identifiers in the GVP 8.5 User's Guide.
6. 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.

7. 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 is executed. The value of this parameter is the Database Identifier (DBID) that Configuration Server assigned for the IVR Profile object.
   The GVP components need this information to log the IVR Profile DBID in the call-detail records (CDR) that they send to the Reporting Server.
8. 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), it consults the policies of the IVR Profile and tenant related to the existing session, to determine whether the call leg can be created. For more information, see Policy Enforcement.
   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), it accepts and processes the incoming request without checking policies.
9. The Resource Manager maintains the session in accordance with configurable session inactivity and session expiration 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, it internally cleans up the call leg data (application, tenant, and resource usage) as if a BYE were 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, see the section about configuring session timers and timeouts in the GVP 8.5 User's Guide.

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 (in the `proxy.sip.min_se` parameter), the Resource Manager rejects the request.
- If the incoming request contains a Session-Expires header with a valid value that is 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 that it receives a SIP request or a 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.

See also CTI Call Mapping Genesys CTI for a description of service selection when the CTI Connector is deployed.

Mapping the Call to an IVR Profile

This section describes how the Resource Manager maps SIP requests to IVR profiles for both single (GVP 8.1.1 and earlier 8.x releases) and multi-tenant (GVP 8.1.2) platforms.

Single-Tenant GVP

In a single tenant environment, 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 that has a name that matches the value of the `gvp-tenant-id` parameter. Alternatively, if either the `X-Genesys-gsw-ivr-profile-id` or `X-Genesys-gsw-ivr-profile-name` header is present, the Resource Manager checks it to determine which IVR Profile to use. If one of the two headers is present and the `Request-URI` parameter `gvp-tenant-id` is also present, the gvp-tenant-id is treated as the tenant.

  - If the Resource Manager finds an IVR Profile that matches, it routes the SIP session to that application and 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 (in the `default-application` parameter, in the gvp.general section).

2. If the SIP Request-URI does not include a `gvp-tenant-id` parameter, but the Voice Platform Solution (VPS) has been configured so that SIP Server provides Dialed Number Identification Service (DNIS) information in the SIP header, the Resource Manager uses the DNIS that it extracts from the SIP message to map the SIP request 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 the section about mapping IVR Profiles to Dialed Numbers in the GVP 8.5 User's Guide.

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

   For more information, see the description of the `rm.sip-header-for-dnis` configuration option in the GVP 8.5 User's Guide.

   Tip

   GVP 8.5 supports all-numeric DNIS or numeric DNIS with asterisk (*) suffix for example, 8005556699 or 80055544*. All other characters such as, #, a, or b are stripped from the incoming request.

3. If the Resource Manager cannot map the SIP request to an IVR Profile, it executes the new SIP session in the context of the default VoiceXML or CCXML application

that has been configured for the Environment tenant (in the `default-application` parameter, in the `gvp.general` section).

4. If the Resource Manager cannot map the SIP request to an IVR Profile, and if no default application has been configured for the Environment tenant, the incoming SIP request fails with a 404 Not Found response.

Multi-Tenant GVP

In a multi-tenant environment, the Resource Manager maps the SIP request to a tenant and an IVR Profile, as follows:

1. If the SIP Request-URI includes a `gvp-tenant-id` parameter and a `X-Genesys-gsw-ivr-profile-id` (or `X-Genesys-gsw-ivr-profile-name`) custom header, the Resource Manager looks for a tenant that has a name that matches the value of the `gvp-tenant-id` parameter and an IVR Profile that has a name that matches the `X-Genesys-gsw-ivr-profile-id` custom header.
    - **If the Resource Manager cannot find a matching tenant, the session fails, and a 404 Not Found SIP response is generated.
    - **If the Resource Manager cannot find a matching IVR Profile, but the tenant is selected, the Resource Manager executes the session by using the default application that is defined for the tenant.
    - **If no default application is defined for the tenant, the session fails and a 404 Not Found SIP response is generated.
    - **If the `gvp-tenant-id` parameter is included in the Request-URI, but the `X-Genesys-gsw-ivr-profile-id` custom header is missing, the Resource Manager first selects the tenant that owns the gateway resource, and then looks for an IVR Profile that has a name that matches the value of the `gvp-tenant-id` parameter.
    - **If there is no matching IVR Profile, the Resource Manager executes the session by using the default application that is defined for the tenant.
    - **As a special case for SSG call flow, if the `gvp-tenant-id` parameter in the Request-URI is missing but the `X-Genesys-gsw-ivr-profile-id` or `X-Genesys-gsw-ivr-profile-name` custom header is present, then the Resource Manager:
    - **Selects the tenant that owns the gateway resource.
    - **Looks for an IVR Profile that has an application-id/name that matches the value of the `X-Genesys-gsw-ivr-profile-id` or `X-Genesys-gsw-ivr-profile-name` custom header.
2. In a Supplementary Services Gateway outbound-call flow, if the SIP Request-URI is not present, but the `X-Genesys-gsw-ivr-profile-id` custom header is present,

the Resource Manager first selects the tenant that owns the gateway resource, and then looks for an IVR Profile that has a name that matches the value of the `X-Genesys-gsw-ivr-profile-id` custom header.

   - If there is no matching IVR Profile, the Resource Manager executes the session by using the default application that is defined for the tenant.

   If any one of the two entries are found, the Resource Manager removes the gvp-tenant-id SIP `Request-URI` parameter from the outbound request.

3. Alternatively, if neither the `gvp-tenant-id` parameter nor the `X-Genesys-gsw-ivr-profile-id` header is included in the request, the Resource Manager uses the DNIS that is extracted from the SIP message to select the tenant and IVR Profile, as follows:
   - The Resource Manager starts with the tenant that owns the gateway resource, using the DID Groups that are associated with the tenant to match the DNIS. It searches the entire tenant hierarchy by using a breadth first order.
   - When a match is found in a DID Group, the Resource Manager designates the tenant that is associated with the DID Group for the call, and then selects the IVR Profile that is associated with that tenant.
   - With a successful match for the DNIS, the Resource Manager attaches it to the `Request-URI` as a parameter that is named trunkport.
   - If the application mapping fails, the Resource Manager executes the session by using the default application that is defined for the tenant.

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.

   The table below describes the parameters that the Resource Manager looks for in SIP messages, to specify the required service:

Table: Service Specified in SIP Request

SIP Request-URI	Service	Comment
User part is dialog, and it contains a parameter with the name voicexml	voicexml	Service prerequisites are included.
User part starts with dialog.vxml	voicexml	Service prerequisites are included.
User part is ccxml, and it contains a parameter with the name ccxml	ccxml	Service prerequisites are included.
User part is `msml[=conf-id]` (`conf-id` is optional)	msml	No service prerequisites are required.
User part starts with conf=	conference	Service prerequisites are included. The remainder of the user part is the conference ID for this request.
Contains the parameter user= phone	gateway	Request originates from a resource that supports the voicexml or ccxml service.
User part is composed of characters 0 9 and - (hyphen)	gateway	Request originates from a resource that supports the voicexml or ccxml service.

2. If the SIP Request-URI does not include parameters to identify the required service, and if the request originates from a resource that supports the voicexml or ccxml service, the Resource Manager handles the incoming SIP request as a request for the external-sip service.
3. If the SIP Request-URI does not include parameters to identify the required service, and if the request does not originate from a resource that supports the voicexml or ccxml service, the Resource Manager uses the service type and service prerequisites that are configured for the IVR Profile in the gvp.general and gvp.service-prerequisite sections.
4. If the Resource Manager cannot map the request to a service in accordance with the preceding rules, it rejects the request with a 404 Not Found SIP response.

NETANN Dialog Requests

The Resource Manager processes NETANN dialog requests from the CTI Connector in the following way:

- The CTI Connector sends the NETANN dialog request with VoiceXML= <APP_URL> and `script-id` parameter.
    - If an INVITE message is received with VoiceXML=SCRIPTURL instead of the APP_URL, the Resource Manager substitutes this for the `SCRIPTURL` that is defined in the IVR profile.
    - If an INVITE message is received with `VoiceXML=SCRIPTURL`, and if the `SCRIPTURL` is not configured in the IVR Profile, the Resource Manager rejects the call leg with a 4xx response.
    - The Resource Manager reads the `SCRIPT-URL` parameter from the `<tt>gvp.service-prerequisites`</tt> section of the IVR Profile object.
    - The `gvp.service-prerequisites` parameter is populated before the Resource Manager sends the request to the Media Control Platform.
    - If the `SCRIPT-URL` parameter is not configured or populated, the Resource Manager uses the INITIAL-PAGE-URL parameter instead.

Policy Enforcement

The Resource Manager tracks sessions, IVR Profiles, and service usage, to enforce policies that are imposed on a per-application and per-tenant 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.

The configuration options specifying the policies that you can configure for GVP are in the gvp.policy, gvp.policy.dialing-rules, and gvp.policy.call-info configuration sections of the IVR Profile and Tenant objects. Configuration options enable you to customize the SIP responses that are sent when the Resource Manager rejects a call because of policy criteria. They also enable you to specify whether certain policy violations will trigger an alarm.

HMT Policy Enforcement

Starting in GVP 8.1.2, the Resource Manager can manage parent-child relationships between tenants in a Hierarchical Multi-Tenancy (HMT) to enforce policies, selectively. HMT tenants are organized in a tree hierarchy, which means that the parent tenant can have many child tenants, and the child tenants can have many child tenants within its own tenant object. There is no limitation on the depth of the tree. Only a single Resource Manager or a Resource Manager High Availability (HA) pair can manage a complete tenant hierarchy.

The Resource Manager manages the policies for the entire tree hierarchy by using a top-down method of enforcement for all parameter categories except category 3 on page 84. Each tenant in the hierarchy, except for the root tenant (Environment), uses the parent Tenant DBID to reference its parent tenant and inherits the policies of the parent tenant.

In general, when enforcing policies the Resource Manager checks the parent tenant to see if there are any overriding policies for the child tenant. If the `gvp.policy.<child-tenant-dbid>` section is defined in a tenant, where `<child-tenant-dbid>` is the DBID of the first child tenant in the hierarchy, the Resource Manager enforces configuration parameters of the parent tenant in the Annex section. The child tenant can have the same parameters defined in its own gvp, policy section of the Annex, but the policies of the parent tenants have priority over the policies of the child tenant.

Policy Enforcement Types

Four types of policy enforcement parameters exist in HMT:

- **Usage-limit parameters** The Resource Manager checks the hierarchy from the top-down to determine whether tenant-level usage is violated. It also checks the parent-to-child enforcement sections to determine if the usage-limit that is configured in the child tenant can be overridden. The last child in the hierarchy is checked, provided that no usage violation is detected at a higher level.
- **Permission-policy parameters** These parameters are treated in the same way as the `usage-limit` parameters in that it searches from the top down until it finds a not-allowed (false) value for one of the options. However, it continues to check for a value of false in any other tenant, including the IVR Profile. If a not-allowed value is encountered at any level, the Resource Manager enforces the policy based on that value. The following parameters are included in this category:
    - `announcement-allowed`
    - `out-bound call-allowed`
    - `transfer-allowed`
    - `voicexml-dialog-allowed`
    - `cti-allowed`
    - `msml-allowed`
    - `recordingclient-allowed`
    - `recordingserver-allowed`
- **Non-enforcement/Unrestricted parameters** The Resource Manager does not use parent-to-child enforcement for these parameters. Instead, it checks the

`IVR-Profile` parameter first. If that parameter is not configured, it searches the hierarchy for a match by using the bottom-up method.
The following parameters are included in this category:
  - `codec/media` and `disable/enable` parameters that are passed to the Media Control Platform
  - speech and language parameters (passed to the Media Control Platform)
  - `mcp-sendrecv-enabled` parameter (passed to the Media Control Platform)
  - `metricsfilter` parameter (passed to other GVP components)
  - `max-subdialog-depth` parameter (passed to the Media Control Platform)
  - `prediction-factor` parameter (related to OCS call distribution related)
- **Ordered list of rules**--The Resource Manager uses the top-down method to check the parameters in this category. The first rule in the ordered list that matches the regular expression determines the outcome of the action. For simplicity, these policies are enforced on all of the child tenants. The parent tenant cannot define rules for one specific child tenant. Tenant and child objects can be managed, created, modified, and deleted by using the Genesys Administrator. The Environment tenant (and any other tenant at the same level) is considered the root tenant. For information about how HMT is displayed and managed in Genesys Administrator, see the Genesys Administrator 8.1 Help.

Speech Resource Reservation

Users can configure the `gvp.policy.asr-reserve` and `gvp.policy.tts-reserve` configuration options to reserve ASR and TTS resources on a per-application or per-tenant basis. When the Resource Manager receives and incoming call and maps it to the voicexml service, it passes those options to Media Control Platform mapped as the `gvp.config.asr.reserve` and gvp.config.tts.reserve SIP `Request-URI` parameters. The Resource Manager uses the bottom-up methodology to enforce the policy by checking the IVR-Profile first, and then the tenant hierarchy.

Speech Resource Limit Policy

You can configure Resource Manager (RM) to enable or disable access to ASR and TTS speech resources, by language and by tenant IVR profile. To complete the configuration, users must partition ASR and TTS so that each engine handles a specific language.

While handling a VoiceXML call, the RM checks the chosen profile to see if it specifies the authorized ASR/TTS engines. If it does, then the RM searches in the tenant hierarchy (bottom-up) for these configuration parameters:

- List of Authorized ASR Engines
  `[gvp.policy.speech-resources]authorizedasrengines`
- List of Authorized TTS Engines
  `[gvp.policy.speech-resources]authorizedttsengines`

If these parameters are configured, the RM passes that information to the Media Control Platform (MCP) handling the request (the VoiceXML call). RM itself will not enforce the policy; that is done by the MCP (or MRCP Proxy, if present).

Context-Services Authentication

On a per-application basis, the Resource Manager supports context-services authentication. The [gvp.context-services-authentication].username and `[gvp.context-services-authentication].password` configuration options can be used to configure a user name and password, respectively for context-services authentication. When the Resource Manager receives and incoming call and maps it to the voicexml service, it passes those option to Media Control Platform mapped as the `X-Genesys-GVP-CS-Username` and `X-Genesys-GVP-CS-Password` custom headers with the url-encoded value.

Transaction List in JSON Format

On a per-application basis, the user can configure a Transaction List by using the `gvp.OPM.Transaction_dbid` configuration option. When the Resource Manager parses the IVR Profile configuration, it checks for a List object. It reads the key-value pairs in the List object s OPM section and transforms them into a JSON string. The Resource Manager maps this string as parameters in the Request URI to the Media Control Platform. If the list changes during runtime, the Resource Manager processes the change and forms a new JSON string, based on the updated `OPM` parameters in the List.

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 that the VoiceXML or CCXML applications can extract. You can configure GVP so that different service parameters are used for each service of each application.

The configuration options specifying the service parameters and service prerequisites that you can configure for IVR Profiles and the Environment tenant, and also the SIP parameters in which the Resource Manager captures this information, are in the `gvp.service-parameters` and `gvp.service-prerequisite` sections of the IVR Profile object.

Managing Resources

All requests for GVP services go through the Resource Manager, which identifies a SIP resource that is capable of serving the request, and forwards the request to it. The Resource Manager monitors GVP resources to maintain an up-to-date status of the resources used in the GVP deployment. The Resource Manager also manages GVP resources to provide load balancing and, if applicable, high availability for each resource type.

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

- Resource Groups
- Monitoring Status
- Selecting a Resource
- Failed Requests
- Recording Server and Client Resources
- Multi-Site Resources

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:

- Recording Server
- voicexml
- ccxml
- msml
- conference
- gateway
- cti
- recordingclient
- recordingserver

Resource service types in the Media Control Platform group are:

- announcement
- conference
- CPD
- Media
- msml
- recordingclient
- voicexml

For detailed information about all the resource group properties, see the descriptions of the logical group configuration options in the section about configuring Resource groups in the GVP 8.5 User's Guide.

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.

Resource Groups:

- Media Control Platform (MCP)
- Call Control Platform (CCP)
- Gateway
- CTI Connector (CTIC)

Resource Groups in HMT Environments

Starting in GVP 8.1.2, Resource Groups are part of a Configuration Unit (CU), with each CU created under a specific tenant. CUs contain a gvp.resources section that includes a `rm_dbid` configuration option that point to DBIDs for specific Resource Manager instances. If this option is not defined, the CU is considered unassigned and not in use.

The following sections describe the way in which the Resource Manager handles CUs and manages the Resource and DID Groups within them:

Handling of CUs

- When the Resource Manager (RM) searches the tenant hierarchy, it parses only those GVP CUs which have the same DBID that the RM has, configured in the `rm_dbid` option. The resources that are contained in the CU service both the parent tenant and all of the child tenants in the hierarchy. Child tenants are identified by the value of the `tenant.N` configuration option, where N=<number> (for example, tenant.1, tenant.2, and so on) is the value of the DBID of the child tenant. If there are no configured `tenant.N options` in the CU, the Resource Manager assigns the resources that are contained in the CU to all of the child tenants in the entire sub tree for the hierarchy. The Resource Manager manages CUs in the following additional ways:

  - When it checks the `tenant.N` option, if the Resource Manager finds that the DBID does not match any of the child tenants in the hierarchy, it ignores that configuration.
  - A single Resource Manager instance can handle the assignment of more than one CU under a specific tenant. In this case, the Resource Manager handles all of the CUs for that tenant and all of the resources that are contained in the CUs.
  - When the Resource Manager is configured for HA, a single CU can be assigned to more than one Resource Manager instance, because, when Resource Manager is clustered, the two instances must manage the same set of resources.
  - The Resource Manager uses a top-down method to assign resources; therefore, although child tenants are serviced by resources that are assigned to the tenant, resources that are assigned to the child cannot service requests for the parent tenant of that child.

Resource Group Management

- The Resource Manager determines if Resource Groups are contained in a CU by checking the `gvp.lrg` option under the Annex section. In HMT environments support these types of Resource Groups:
  - Media Control Platform groups
  - Call Control Platform groups
  - CTI Connector groups
  - Gateway groups
  - Recording Server groups

Gateway Resource Groups continue to conform to the Resource Group architecture in GVP 8.1.1 and earlier 8.x releases, and the address of record (AOR) and port count are configured in the same way.

Resource Configuration

- The Media Control Platform, Call Control Platform, and CTI Connector Applications have a gvp.rm section, which includes the following configuration options:
  - `aor`--Address of record
  - `port-capacity`--Number of ports that are allocated for this resource

  - `redundancy-type`--Monitoring status of the resource (active or passive)

DID Group Validation

- The Resource Manager validates the DID Group and DN assignments for all tenants.

Monitoring Status

The Resource Manager acts as a SIP registrar (Request for Comments [RFC] 3261) for resources about which it receives information from Management Framework. The Resource Manager maintains information about the registration and usage status of each resource. If the logical resource group has been configured for monitoring (in the `monitor-method` configuration option), 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).

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

Notification of Resource Status

When Resource Manager is deployed as part of the VPS solution in a multi-tenant environment, it provides port-availability notifications that include data which is specific to each tenant.

SIP Server generates a SUBSCRIBE request that includes an `X-genesys-mediaserver-status` event and the Resource Manager immediately sends a NOTIFY response, which includes the current status of all Media Control Platforms in the deployment, whether in-service or out-of-service, and it continues to send notification if the status of any one of the Media Control Platform changes.

Each entry in the body of the NOTIFY message is in the following format, `<Tenant-Name>/<MCP-IP:Port>/<MCP-Status>`, where `<Tenant-Name>` is the name of the tenant that was specified in the SUBCSCRIBE request and `<MCP-Status>` is either in-service or out-of-service. See the following example of a NOTIFY message:

```
NOTIFY sip:Environment@10.10.30.46:5060 SIP/2.0 Via: SIP/2.0/UDP
10.10.30.212:5098;branch=z9hG4bK0a5b6af05e2ed0
```

```
From: sip:Environment@10.10.30.212:5066;tag=0F2DC9A6-2F17-486C-58AD-
B84E3F788C86 To: sip:Environment@dev-
photon:5060;tag=61B43FC6-F841-4326-A4F8-A8DA34DE627B-1 Max-Forwards:
70 CSeq: 1 NOTIFY Call-ID:
09142015-1524-4EF8-8FE6-F3AA8898F0BD-1@10.10.30.46 Contact:
<sip:GVP@10.10.30.212:5098> Content-Length: 77 Content-Type:
application/ x-genesys-mediaserver-status Supported: timer
Environment/mcp1.com:5070/in-service Environment/mcp2.com:5070/out-
of-service
```

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 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, which are configured in the gvp.policy configuration section in the IVR Profile object.
- Information that is parsed from SIP `Request-URI` parameters that have the `gvp.rm.resource-req` prefix. The Resource Manager does not forward these `Request-URI` parameters unless a CTIC resource is selected and the parameter `rm.pass-capability-params-to-ctic` is set to `true`.

Locating Resources Using Geo-Location

In multi-tenant environments, requests from gateway resources are parsed for the `X-Genesys-geo-location` custom header.

> You can configure geo-location for an LRG through the Resource Group Wizard within Genesys Administrator. See "Configuring Logical Resource Groups" beginning on page 89 in the GVP 8.5 User's Guide.

- If the custom header is included in the request, the RM checks the configured `geo-location` parameter for each resource group to determine if the `geo-location` parameter matches the location in the request.

- If the RM finds more than one group that matches the geo-location information, it routes the call to the group that best meets criteria such as preference and capability.
- If no groups match the geo-location, or the groups that match do not have available ports:
    - The RM routes the call to any group that has available ports, even if the location information does not match. Unless...
    - If the Service Type of the call request is configured in option `reject-on-geo-location-nomatch`, or if the request INVITE contains the `X-Genesys-strict-location:enforce` header, the Resource Manager will not route the call to any Logical Resource Group (LRG) whose geo-location does not match the geo-location specified in the INVITE. Instead, the Resource Manager will reject the call.
      You can use the option `reject-on-geo-location-nomatch` to configure Service Types that require strict geo-location matching.
      For example, `reject-on-geo-location-nomatch = conference, recordingclient`
      If the Service Type of the incoming INVITE is `conference` or `recordingclient`, RM will perform strict geo-location matching and reject the request in case no matching resources are found.
      With release 8.5.1, the option `reject-on-geo-location-nomatch` replaces the former option `reject-recording-request-on-geo-location-nomatch`, which specified this same behavior but only for calls being recorded. `reject-on-geo-location-nomatch` applies to all calls.
      In response to a call request from a gateway resource (SIP Server), the RM routes only to Logical Resource Groups (LRGs) with a geo-location that matches the geo-location specified in the INVITE message. If the RM cannot identify a matching LRG, it rejects the call.

> **Tip**
>
> These LRGs specify MCP (recording-client) or Recording Server (recording-server) resources. An LRG with no available MCPs and/or Recording Servers is considered non-matching, even if its geo-location is correct.

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 Resource Selection for Conference Services), the Resource Manager selects the physical resource based on the load balancing scheme for the group. The load balancing options are:

- `Round robin`--From a circular list, the Resource Manager selects the next resource whose usage has not exceeded configured limits.
- `Least used`--The Resource Manager selects the resource with the lowest usage that has not exceeded configured limits.
- `Least percentage used`--The Resource Manager selects the resource from the resource group with the least percentage of resource usage.

Usage is calculated in the manner specified by the `port-usage-type` parameter.

For more information, see the description of this parameter in the GVP 8.5 User's Guide.

Load Balancing in Resource Manager vs. in Media Control Platform

- 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 third-party speech engines, on a round-robin basis.

Load-Balancing for Gateway Services

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

If ESS is deployed, the Resource Manager uses a third source to identify the capability requirements (see Resource Selection for Conference Services), the `X-Genesys-geo-location` custom header.

Outbound-Call Distribution

The Resource Manager can use the prediction factor (`factor-p`) parameter to predict the ratio of agent calls to customer calls in a campaign, based on the assumption that factor-p can vary between 0.5 and 1.0.

A customer outbound call in a campaign is identified by the value of the `X-Genesys-gsw-predictive-call` SIP custom header. If the value is on, it is a customer call, if the value is off or the custom header is absent, it is identified as an agent call.

When the Resource Manager receives subsequent calls on an existing campaign, it finds the resource (Media Control Platform) with the maximum number of calls for that campaign that also has free ports, and uses the following process:

1. The Resource Manager examines the current number of agent calls (A), outbound calls (O), and free ports (F) on the Media Control Platform
2. If the new request is an agent call, and `A - O < F x P`, the Resource Manager places this call on the Media Control Platform.
3. If the new request is an outbound call, and `O - A < F x P`, the Resource Manager places this call on the Media Control Platform.
4. If the Resource Manager is unable to route the call to the Media Control Platform with the maximum number of call for that campaign, it finds the resource with the next highest number of calls for this campaign (with free ports) and it repeats Steps a, b, and c.
5. If the Resource Manager is unable to find a Media Control Platform by using these steps, it sends the request to a new Media Control Platform (if there is one available).

You can also set a factor-P value in the `gvp.policy.prediction-factor` parameter for IVR Profiles. If the value is not changed, 0.5 is the default value.

No Resource Selected

If the Resource Manager is cannot select a resource to meet the request, it responds to the SIP request with a configurable error message. For more information, see the section about customizing SIP responses in the GVP 8.5 User's Guide.

Resource Selection for Conference Services

Conference services have the special requirement in 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 if the specified `confmaxsize` value is less than the `confreserve` value, 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 the confmaxsize and confreserve requirements that are specified in the SIP `Request-URI` parameters with the conference maximums that have been configured for the IVR Profile and the resource group, taking into account the current status and usage of conference resources. For more information about the IVR Profile and resource group parameters that are considered, see the section about enabling conference services in the GVP 8.5 User's Guide.
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).

   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 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.

   > **Tip**
   >
   > When the conference session is established, the maximum number of participants is the smallest among the conference size maximums(`confimaxsize` parameters) specified in the SIP request, the IVR Profile, and the resource group.
   >
   > As a result, the Resource Manager might 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 in this procedure).

5. When the Resource Manager receives subsequent requests with the same conference ID, it forwards each request to the same conference resource, provided that 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, leaving it to the conference resource to reject the request if necessary.

   If the maximum size of the conference or the usage limit configured for the resource is exceeded when the new call leg is added, the Resource Manager rejects the request.

   If a request is received for a new participant to join an existing conference, the Resource Manager detects that the conference resource has gone offline, and releases all the calls associated with the conference. When the new participants join the conference, the Resource Manager routes the requests to a new online conference resource with available ports.
6. If the Resource Manager cannot select a resource to meet the request, it responds to the SIP request with a configurable error message. For more information, see the section about customizing SIP responses in the GVP 8.5 User's Guide.

Resource Selection in HMT Environments

In HMT environments, the Resource Manager checks the Management Framework objects that it manages to determine if they are in an enabled or disabled state and processes requests, based on the following rules:

- If a resource is disabled, new calls are not forwarded to that resource for processing.
- If a resource group is disabled, it is excluded during resource selection for new calls.
- If an IVR Profile is disabled, new calls for this profile are handled by the parent tenant s default profile. If a default profile is not defined, the call is rejected.
- If a Tenant is disabled, new calls for this tenant are rejected.

The Resource Manager reads the state information for Management Framework objects during initialization. It also detects dynamic state changes during runtime.

> **Warning**
>
> Take great care when you disable tenant objects. Disabling a tenant does *not* disable its child tenants. But it does have a cascading effect on all of the objects that are owned by the tenant, including Resource Groups, resources, and IVR Profiles.

Exclusive Resources for Tenants

An enhancement to resource selection for HMT tenants, enables access to exclusive resources to specific tenants.

Previously, when resources were configured for a tenant, they were, by default, also available to its child tenants. Now when a tenant's DBID is configured in the parent tenant's `tenant.N` option, the child tenant cannot use the resources. In addition, the `tenant.N` option could be configured to allocate resources to a specific subset of child tenants.

Now, you can add the exclusive configuration option to the gvp.resources section in the GVP Configuration Unit (CU). When this option value is set to `true`, the resources under the CU are dedicated to the tenants that are specified in the `tenant.N` option only. Thus:

- Set the parent tenant s DBID in `tenant.N` to exclude the children from using those resources.
- Set the child tenant s DBIDs in `tenant.N` to exclude the parent from using those resources.

Therefore, if the parent tenant's DBID is configured in the `tenant.N` option, the child tenants cannot use the specified resources. Conversely, if the child tenant's DBID is configured in the `tenant.N` option, the parent tenant cannot use the specified resources.

If the exclusive configuration option value is set to false, or if it is not configured in the CU, resource allocation occurs as described Resource Selection in HMT Environments.

Failed Requests

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

- For voicexml, ccxml, conference service and CTI Connector 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 it has tried all resources in the group.
- For a gateway services 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 receives no successful 2xx responses, but it does receive at least one final response from one of the resources, it forwards one of the received responses to the UAC, in the order of selection shown in this table:

Table: Order of Selection for Responses to the UAC

	Response Received	Response Returned to the UAC
1.	6xx	6xx
2.	401, 407, 415, 420, 484	401
3.	Any other 4xx response	The first 4xx response received
4.	Any 5xx other than 503	The first 5xx response received
5.	500 Server Internal Error	500 Server Internal Error

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

For more information about the SIP response codes that GVP components generate, see the appendix about SIP response codes in the GVP 8.5 User's Guide.

Failed ASR/TTS Reservation Requests

If the Media Control Platform rejects a SIP INVITE request and the service-type is voicexml, the Resource Manager processes a Warning header, if configured, to check the specified warning code. A 390 warning code indicates an ASR reserve failure and a 391 warning code indicates a TTS reserve failure.

When Resource Manager receives these types of warning codes, it checks the `gvp.policy.speech-reserve-failure-retry` parameter in the profile:

- If it is not configured in the profile or is set to `true`, the Resource Manager routes the call to another Media Control Platform in the same resource group.
- If it is set to `false`, the Resource Manager checks the value of the `gvp.policy.speech-reserve-failure-response` configuration option to see if it is set to a valid SIP error response code (by default, set to 0). If it is, this response code is returned to the UA. If it is not, the response sent from the Media Control Platform is returned.

Recording Server and Client Resources

Resource Manager manages recording servers and recording clients by detecting and monitoring them to provide and facilitate GVP Call Recording services. In the Call Recording solution, the Resource Manager functions include:

- Provisions third-party recording servers.
- Provisions Media Server resources.
- Handles load balancing and failover of Media Servers.

For more information about how the Resource Manager manages Reporting Servers and Clients, see the section Recording Servers and Clients in Chapter 2 of the Genesys Media Server 8.5 Deployment Guide.

Multi-Site Resources

The Resource Manager supports GVP multi-site configurations, which consist of multiple single-site deployments, with each site consisting of a Resource Manager instance (or an HA pair), a Reporting Server instance (or an HA pair), and multiple Media Control Platform instances. The Resource Manager shares resources and enforces policies consistently across all sites. In addition, administrators can generate real-time and historical reports with or without site identification filters, which means they can also generate system-wide reporting data.

Site Identification

Resource Manager obtains information about GVP sites by checking the site folder object, which is configured in Management Framework and can be viewed in Genesys Administrator on the Provisioning tab, under **Environment > Applications**. The folder

Annex contains a `gvp.site` configuration section, in which site configuration options are kept. They include (but are not limited to) the following parameters:

- Weight Relative weight for this site
- Geo-Location Comma-separated list of geo-locations associated with this site
- Resource Sharing Whether resource sharing is enabled or disabled for this site
- Contact The SIP route address for this site

The Resource Manager uses the name of the site folder in Management Framework for the site name. It must be unique within the Applications folder. The site ID is taken from the site folder s DBID and must also be unique. The site folder contains only Resource Manager and Reporting Server components.

Parsing Site Configuration

The Resource Manager in each site reads the local site configuration as well as the configurations of the remote site in the deployment and subscribes to notifications for the site object configuration changes. If the weight factor, or any other site properties change, the Resources adjusts its site information accordingly.

Monitoring Other Sites

The Resource Manager in each local site monitors the remote sites, by using SIP OPTIONS messages to ping the Resource Managers in those sites. To do this, it uses the `contact` parameter that is provided in the gvp.site configuration section of the remote sites folder. In this way, each Resource Manager can determine when the remote sites are online or offline.

Multi-Site Policy Enforcement

The Resource Manager enforces policies dynamically across the multiple-site deployments, by using usage-based counters only (Type I policy enforcement). All other policy enforcement types (Type II, III, and IV) are enforced locally on a per-site basis. (See Policy Enforcement Types.)

For detailed information about the usage-based counters, see Usage Limit Counters.

Multi-Site Resource Sharing

The Resource Manager can manage resource sharing across multiple sites. Any site can be selected to enable resource sharing by setting the `gvp.site.resource-sharing` configuration option value to `true` (default) in the site folder. When resource sharing is enabled, and there are no Media Control Platform resources available, the local-site Resource Manager can insert a `routeset` parameter to forward requests to this site. The Resource Manager only forwards requests to sites that have resource sharing enabled. In this case, policy enforcement is done at the local site.

The Resource Manager adds a `X-Genesys-GVP-Site-ID` custom header to the request its own site ID set as the value. This enables other sites to determine the originating site for the request and whether or not further policy checking is required.

Resource sharing applies to the Media Control Platforms only. If a CTI-Connector or Call Control Platform resource is required for a call and there are no ports available, the Resource Manager does not forward the request to a remote site and the existing logic for handling these kinds of requests is used. See How the Resource Manager Works.

Multi-Site Reporting

In GVP multi-site environments, Reporting Server collects data from all GVP segments to generate historical and real-time reports on a per-site or system-wide basis. The Resource Manager logs the site ID into the CDR to identify the site, to which the data applies.

For a detailed description of how multi-site reporting works, see GVP Multi-Site Reporting.

Full Call Recording Requests (from Cisco UCM)

Through the Resource Manager, Cisco's Unified Communications Manager (UCM) can initiate full call recording sessions directly with the Media Server. In other words, it can initiate sessions without the assistance of Cisco T-Server or the UCM Connector.

A recording session is made up of two separate SIP dialogs, where each SIP dialog carries the media stream for one of two parties on the call. The Resource Manager handles these requests by ensuring that the SIP Request URI is transformed into a NETANN format. The recording can then be initiated for two separate dialogs with recording filename formats.

The Resource Manager processes the requests from Cisco T-Server in the following manner:

1. A request from Cisco UCM lands on a DN that is mapped to an IVR Profile that contains the announcement service-type.
2. Resource Manager checks for the `near/farend` parameter in the From header. If it exists, it treats this call as a Unified Communications Manager (UCM) call.
3. The Resource Manager uses the `near/farend` parameter, the DN, and the call-reference to generate the recording file format.

4. The Resource Manager ignores the Date header in the request and chooses to provide the date information for the file format on its own.
    - If `x-nearend` is specified, the recording filename is in the following format:
      `CUCM/call-$RefCI$-at-$AgentDN$-on-$YYYY$-$MM$-$DD$`
      The Resource Manager will then generate a NETANN request to the Media Control Platform in the following format:
      `INVITE sip:annc@mcp:5060;record=CUCM/`
      `call-24432480-at-7013-on-2009-01-05 SIP/2.0`
    - If `x-farend` is specified, the recording filename is in the following format:
      `CUCM/call-$RefCI$-at-$AgentDN$-on-$YYYY$-$MM$-$DD$ (2)`
      The Resource Manager will then generate a NETANN request to the Media Control Platform in the following format:
      `INVITE sip:annc@mcp:5060;record=CUCM/`
      `call-24432480-at-7013-on-2009-01-05(2) SIP/2.0`

Tip

For full call recording requests from Cisco T-Server, the Resource Manager does not require the service-prerequisite requirements for the announcement service.

For more information about the UCM Connector and Cisco T-Server, see the Genesys Media Server 8.5 Deployment Guide.

How the Policy Server Works

Policy Server exposes a read-only HTTP/HTTPS interface that can be used by other components to perform GVP policy queries for the entire deployment.

HTTP Response Format

The HTTP interface returns data (responds to queries) in JSON format with the content-type `application/json` and supports JSONP when a query parameter named callback is supplied. The JSON callback response is returned with the content-type `text/javascript`.

If the target service does not recognize the query parameter, the query is silently ignored. A request within a partial URL that does not directly match an existing service receives a 404 response.

Further information about how Policy Server performs its role in a GVP deployment is provided in the topics:

- DID Management
- Policy Management
- Service Description
- High Availability

DID Management

Policy Server recognizes Direct Inward Dialing (DID) number range specifiers and named DID groups.

DID Range Specifiers

Policy Server recognizes two or more DID numbers expressed as DID range specifiers in one of the following three forms:

1. A single DID, for example, 300.
2. A range of DIDs, for example, 300-400 means all numbers between 300 and 400, inclusive. The lower number must be first in the range and the higher number second. For example, 400-300 is an invalid specifier.
3. A DID prefix, for example, 45* means all numbers that start with the digits 45 (for example: 45, 450, 4500, 4501, 4599 and up to the maximum allowable that match the prefix).

A string that does not match any of these three forms is considered an invalid specifier, and two DID range specifiers are considered overlapping when at least one DID in each span is the same.

Named DID Groups

Named DID groups can contain zero or more DID range specifiers and a tenant can have zero or more DID group assignments. Policy Server maintains DID groups, DID range specifiers for the entire deployment in its in-memory store. Updates to DID Groups or their DID range-specifiers are immediately reflected in-memory.

DID Overlap Queries

Genesys Administrator can query Policy Server for any overlaps in the DID range specifiers in the deployment by using an HTTP GET request in the following format:

`GET /dids/overlaps/?spec=<specifier>[&spec=<specifier>]`
...where `<specifier>` is a DID range specifier, for example: `GET /dids/overlaps/?spec=300-400&spec=550&spec=551`

A request can have multiple spec parameters in a GET request to query overlaps for multiple DID range specifiers or it can have none. If there are no spec parameters in the request, no overlap results are returned.

The maximum number of spec parameters that can be simultaneously specified depends on the client s URL length limits and the server implementation. If there are any invalid specifiers in the request, a 400 response is returned. If no overlaps are found or no spec parameters are specified, a 200 response with an empty array is returned, for example:

```
No overlap response [ ]
```

If overlaps are found, a 200 response is returned with an array of overlap details. Each array item is an object with the following properties:

- **specifier** the DID range specifier in the request.
- **overlaps** An array of objects that provide details about the overlap. The objects have the following properties:
    - **tenant** The tenant with the id property (the DBID of the tenant).
    - **group** The DID group with the name property.
    - **specifier** The DID range specifier that contains the overlap.

See the following example of an overlap response:

```
[
 {
   "specifier": "55*",
   "overlaps": [
     {
       "tenant": { "id": 101 },
       "group": { "name": "Group1" },
       "specifier": "500-600"
     },
     {
```

```
        "tenant": { "id": 101 },
        "group": { "name": "Group1" },
        "specifier": "5567"
      }
     ]
    },
      {
        "specifier": "6700",
        "overlaps": [
          {
            "tenant": { "id": 101 },
            "group": { "name": "Group1" },
            "specifier": "6000-8500"
      }
    ]
  }
 ]
```


Configuration of Maximum Overlaps

You can use the did.max_overlaps option to configure the maximum number of overlaps that can be returned. The default value is 10.

Policy Management

The Resource Manager makes call processing and resource allocation decisions, based on the policies that are defined within the tenant hierarchy and IVR profiles. Some policies are inherited and thereby, enforced by the parent tenant.

The moment the Resource Manages serves up a resource (for example, call or speech processing), the policies that are in effect are resolved by the Resource Manager on-the-fly, based on the current resource utilization.

Static Analysis of Policies

Policy Server performs static analysis on some policies to validate and determine the inherited values or enforcements that are currently in effect for a tenant or IVR profile. This information can be used by other components to determine the policy values that are currently in effect.

Tenant Policy Queries

You can query the Resource Manager tenant policies by using an HTTP GET request in the following format:

`GET /tenants/<tenant id>/policies/[<policy>]` where:

- `<tenant id>` is the DBID of the tenant
- `<policy>` is the name of the policy (optional), for example:

```
GET /tenants/101/policies
```

```
GET /tenants/101/policies/max-ports
```

> **Tip**
>
> The <policy> parameter is optional. If it is omitted, Policy Server returns all known policies (not just the currently defined tenant policies).

Response Rules For Tenants

If a specified tenant does not exist, a 404 response is returned. If a specified policy does not match a known policy, Policy Server performs the following steps in this order:

1. Checks for parent enforcement of this policy. If yes, the enforcement value is used as the effective value.
2. Checks for an existing value for this policy. If yes, this value is used as the effective value.
3. Checks the query to see if there is a new assigned value for this policy. If yes, use the new value as the effective value.
4. If there is no parent enforcement, existing value, or newly assigned value, there is no effective value for this policy.

When an existing tenant and a known policy are specified (or no policy is specified), a 200 response is returned with an array of policy details. Each array item is an object with the following properties:

- **name** The name of the policy.
- **value** The value of the policy that is defined for the queried tenant.

- **enforcement** The value that is enforced by the immediate parent of the queried tenant.
- **effective** The effective value that is resolved, based on the inheritance and enforcement rules for this policy.

See the following example of an effective policy response:

```
[
 {
   "name": "max-ports",
   "value": 300,
   "enforcement": 250,
   "effective": 250
 },
 {
   "name": "conference-enabled",
   "value": false,
   "effective": false
 }
]
```

IVR Profile Policy Queries

You can query the Resource Manager IVR Profile policies by using an HTTP GET request in the following format:

`GET /tenants/<tenant id>/ivrprofiles/<profile id>/policies/[<policy>]` where:

- `<tenant id>` Is the DBID of the tenant.
- `<profile id>` Is the DBID of the IVR profile.
- `<policy>` Is the name of the policy (optional), for example:
  `GET /tenants/101/ivrprofiles/42/policies`
  `GET /tenants/101/ivrprofiles/42/policies/max-ports`

> **Tip**
>
> The <policy> parameter is optional. If it is omitted, Policy Server returns all known policies (not just the currently defined IVR Profile policies).

Response Rules For IVR Profiles

The response rules are the same for the for IVR Profile policy queries as they are for the tenant policy queries. See Response Rules For Tenants.

Service Level Policies

Managed Service Providers in general and Enterprise departments in particular may use GVP to provide self-service to many different, independent entities, using the same shared GVP system of resources. Some entities may also do outbound calling in bulk mode at various times of the day, and this load could impact inbound self-service call handling, due to the potentially excessive load.

To manage these different interests and loads, and also to accomplish billing solutions for MSPs, GVP offers limit settings that are based on the number of simultaneous calls allowed for a given application, which is effectively equivalent to the number of ports.

Level 1 and 2 thresholds are non-blocking, and when exceeded by traffic demands, result in this fact being recorded in all CDR records for that application during the excess traffic. Level 3 can be set to blocking, which means that if the number of simultaneous calls equals the level 3 setting, then any subsequent calls arriving for that application will be rejected until traffic volumes decline below the threshold.

By writing the levels into the CDR record, the potential exists to bill the application owner (or tenant owning the application) for different levels of call volumes without blocking calls.

Procedure: Setting Service Levels

Use Genesys Administrator (GA) to set the parameters that specify these service levels, for both tenant and profile.

1. Set the parameter `[gvp.policy] burst-allowed` (which determines whether bursting is allowed or not) to `true`.
2. Enter the number of ports assigned to the customer in the parameter `[gvp.policy]usage-limits`.
   These are level 1 ports the initial ports that a customer purchases for which they will be billed at normal rates. Rule 1 refers to the default usage policy value that is present when a GVP installation is done for a customer and specific number of ports are allotted. If its value=0 (not empty), all calls fail. You can set Rule 1 at the Tenant level or at the specific IVR application level.

3. Enter the number of ports assigned to the customer in the parameter `[gvp.policy]level2-burst-limit`.
   These are level 2 ports used when all of the Level 1 Ports are exhausted. For these ports, the customer is billed at a different rate. `Level 2 Ports = Level 1 Ports + Level 2 bursting` For example, if the customer has 1000 ports at level 1 and they want 100 more ports in the next level, set level 2 ports at 1100. Therefore, if the customer IVR profile receives 1100 simultaneous calls, 1000 ports are billed at normal rates, and 100 ports are billed at the level 2 rate.
4. Enter the number of ports assigned to the customer in the parameter `[gvp.policy]level3-burst-limit`.
   These are level 3 ports the next level of bursting, billed at a higher rate than level 2. For example, if the customer wants 100 ports in level 3, then set the level 3 Ports field to `1200 (1000 + 100 + 100).`
5. Enter the number of billable ports assigned to the customer in the parameter `[gvp.policy].`
   Billable ports are used by some customers where the number of ports to be billed is different from the number of ports provisioned (port levels 1, 2, and 3). This information is transferred to Reporting Server and stored as part of the CDR records. The parameter `[gvp.policy]` is available only if Reporting Server is installed.

Policy Resolution For Uncommitted Values

Policy Server can stage a policy resolution of uncommitted values by providing the value and enforcement query parameters when effective policies are queried for either a tenant or an IVR profile. This can be useful when you want to know what the effective policies will be before you save any changes that were made to the policies.

Policy Server supports a limited set of policy resolution types, which is a subset of the types that the Resource Manager can achieve during real-time call processing.

- Policy Server supports two top-down policy resolution types limit and feature-allowed. For example, if the tenant s usage-limits option has a value of 23 and its parent tenant usage-limits option has a value of 8, then the tenant's effective value for usage-limits will be 8.
- When there is no defined resolution type, Policy Server supports bottom-up (or pass-through) resolution of policy values in the following ways:
  - If the tenant's policy value is available, it is used as the effective value.
  - If the tenant's policy value is not available, the first available value of a parent tenant's policy is used from the bottom-up.
- Policy Server supports one-level enforcement policy resolution. The tenant's policy value is enforced by its immediate parent tenant. When the policy value is enforced,

the enforced value is used, regardless of any other policy values. Enforcement policy resolution is applied for tenants only, (not for IVR Profile policies).

Service Description

Policy Server returns a Web Application Description Language document that describes the services it provides whenever a user visits the root URL ("/"). The WADL document is styled with an embedded Extensible Stylesheet Language (XSL) reference to render an HTML page that acts as documentation for the services. Policy Server serves the static files that are required for this transformation (XSL, Cascading Stylesheet (CSS), and image files) from the /static URL.

The WADL document includes the following information:

- The name of the product (VP Policy Server).
- The name of the Management Framework Application object that is represented by this instance.
- The product version.

High Availability

You can deploy Policy Server in warm standby mode for High Availability by using the Backup Server option in the Server Info section of the Policy Server Application. The active standby status is determined by the Solution Control Server (SCS), and because the Policy Server is stateless, data does not require synchronization.

Data Storage

Policy Server uses in-memory to store the data that it obtains from Management Framework. It does not require external databases or files for data storage.

How the CTI Connector Works

Like other GVP components, the CTI Connector (CTIC) relies on the Resource Manager for session management, service selection, policy enforcement, and resource management.

The Resource Manager processes CTI calls for each physical resource that represents a CTI Connector. A call is identified as a CTI call:

- If it arrives from a gateway resource.

- The `use-cti` parameter is set to a value other than zero in the Gateway resource group from which the call arrives.

This section describes how the CTI Connector performs its functions, in the following topics:

- Inbound Call Mapping
- Outbound Calls
- Genesys CTI Deployment Modes
- Integration with Cisco ICM
- Cisco CTI Deployment Modes

Inbound Call Mapping

When the CTI Connector is deployed, the Resource Manager manages call sessions in the following way:

- A CTI call arrives from gateway resource, and the Resource Manager routes the call to a CTI Connector resource. The Resource Manager marks the session as a CTI session.
- The Resource Manager checks the use-cti parameter in the Gateway group and, based on the parameter value, determines how the call is mapped for example:
    - `use-cti = 0`—The call is not treated as a CTI call. The DNIS is provided and mapped to an IVR Profile.
    - `use-cti = 1`—Initially, the DNIS is not provided and the call is not mapped to an IVR Profile. This is done later, as described in SIP Back-to-Back User Agent.
    - `use-cti = 2`—The DNIS is provided and the call is mapped to an IVR Profile; however, the call may be treated as a CTI call, depending on how the gvp.policy section of the IVR Profile is configured:
        - If `cti-allow = false`, the call is treated as non-CTI call.
        - If `cti-allow = true`, the call is treated as a CTI call.
        - If `cti-allow` is not configured, the call is treated as a CTI call.
        - If `use-cti = 2`, then for CTI calls, the Resource Manager extracts the CTI service parameters configured in the IVR Profile and sends them to the CTI resource as Request-URI parameters. It also sends these parameters in mid call requests, such as SIP REFER.

Tip

When you create a Gateway resource group by using the Resource Group wizard, the value that you enter in the CTI Usage field, configures the use-cti parameter. See CTI Connector Functions.

CTI Connector Resource Selection

- The Resource Manager selects a CTI Connector resource group to service the call based on its preference and capability. (See Resource Groups.)

Tip

In multi-tenant environments, only one CTI Connector service can be included in a Resource Group, and there can be only one CTI Connector Resource Group per tenant.

- A request is sent to a resource in the CTI Connector Resource Group based on the load balancing scheme. (See Load Balancing.) The Resource Manager modifies the `X-Genesys-GVP-Session-ID` header in the request, as follows:
    - It adds the parameter `gvp.rm.cti-call = 1`.
    - It adds the parameter `gvp.rm.tenant-id` (but only if `use-cti = 2`).

CTI Call Mapping Genesys CTI

- If the Resource Manager receives an INVITE from CTIC, and the call is not mapped to an IVR Profile (`use-cti=1`), the Resource Manager maps the call and selects a service for the request. The Resource Manager adds the `gvp.rm.tenant-id` parameter to the `X-Genesys-GVP-Session-ID` header as described in Connector Resource Selection.

- The CTI Connector, acting as a B2BUA, fetches the Automatic Number Identification (ANI), DNIS, Connection Identifier (CONNID), and Universal Unique Identifier (UUID) from the IVR Server when a request is received from the Resource Manager. The CTI Connector then sends a new SIP INVITE to the Resource Manager with the following information:
    - The user part of Request-URI set to DNIS
    - The user part of FROM header set to ANI
    - The user part of TO header set to DNIS

When the Resource Manager receives a request from the CTI Connector, it searches for DNIS based on the `rm.sip-header-for-cti-dnis` parameter.

CTI Call-Detail Records

- The Resource Manager processes the `X-Genesys-GVP-CDR` header for the following information in the BYE request from the CTI Connector, or in the final response for a BYE message to the CTI Connector:
    - The call-disposition information
    - The call wait time in the queue

The Resource Manager passes this information to Reporting Server in the final CDR for the call.

Outbound Calls

The Resource Manager supports outbound calls from the Media Control Platform to the CTI Connector and outbound calls from the CTI Connector to the gateway from which the call originated.

- The Resource Manager routes the outbound calls that are initiated by the Media Control Platform to the gateway with which the CTI Connector instance is associated.
    - The Resource Manager attaches the CTI-related service parameters as Request-URI parameters. They are configured as cti service-types The format is the same as that described in Selecting the Service on page 81.
    - The following CTI-related service parameters are attached for transfer requests:

- `DefaultAgent`
- `TransferOnCTI`
  For behind mode (`use-cti=1`) and during blind transfer cases, `TransferOnCTI` is applicable only when CTI Connector is deployed with Genesys CTI. The valid values for `TransferOnCTI` are `Yes` and `No`, and the value type is `fixed`.

> Tip
>
> The Resource Manager passes on the `DefaultAgentNumber` service parameter for both outbound SIP INVITE and REFER messages. The CTI Connector uses the `TransferOnCTI` parameter only in SIP REFER messages.

- For an existing CTI Connector session, the Resource Manager forwards outbound calls from the CTI Connector to the gateway resource from which the inbound call arrived. To ensure this occurs, retain the default value, Always, for the `gvp.policy.use-same-gateway` parameter in the application profile.
- When the Resource Manager receives SIP INVITE messages from the CTI Connector, it uses the following logic to determine whether the message is for an outbound or inbound call:
  - If the Request-URI from the CTI Connector contains the `gvp.ctic.outbound` parameter that is set to a non-zero value, it is an outbound call and must go through the gateway.
  - If `gvp.ctic.outbound = 0`, it is an inbound call.
  - If `gvp.ctic.outbound` is not set in the Request-URI, it is an inbound call.

Failed Requests

If the CTI Connector sends a specific 4xx or 5xx SIP response code in the initial INVITE message, the Resource Manager assumes that connectivity to the CTI server is broken.

- The Resource Manager checks the `rm.cti-unavailable-action` parameter. If it is set, the Resource Manager performs the action specified in the parameter. If it is not set, the Resource Manager check the next resource in the CTI group. Possible action values are:
    - `answer`--This call is considered a non-CTI call. The DNIS is provided and the call is mapped to an IVR Profile based on the initial SIP INVITE message from the gateway (if the IVR Profile is not already mapped).
    - `reject`--The Resource Manager does not retry any further CTI resources in the CTI group, and it rejects the call with the response code from the CTI Connector.
    - `script;<service-type>;<url>`--The Resource Manager sends a NETANN request based on the service-type and Universal Resource Locator (URL). The request is sent in the context of the mapped IVR Profile or the default IVR Profile (if mapping fails).
        - When the CTI Connector sends the first SIP INVITE message to the Resource Manager, and if the call is not mapped to an IVR Profile, the Resource Manager checks that the IVR Profile is not already mapped. If it is not, the Resource Manager maps the call and passes the CTI service parameters to the CTI Connector in a 200 OK response in the `X-Genesys-GVP-CTI-Params` header.

Genesys CTI Deployment Modes

The CTI Connector interacts with other components in the Genesys suite by using the IVR Server XML interface. IVR Server can be deployed in front of the switch, behind the switch, or in Network mode:

- If IVR Server is in front of the T-server (or TDM) switch Inbound calls that are routed through the Resource Manager to the Media Control Platform contain call-related information, such as, the ANI, DNIS, DN, and IVR port number in the SIP INVITE message.
- If IVR Server is in Network mode Inbound calls that are routed through the Resource Manager to the Media Control Platform contain call details such as, the ANI, DNIS, Toll Free Number (TFN), and IVR port number in the SIP INVITE message.
- If IVR Server is behind the switch Inbound calls that are routed through the Resource Manager to the Media Control Platform do not have the ANI or DNIS in the SIP INVITE message. Only the channel identifier is presented to GVP. In this case, GVP retrieves the ANI and DNIS from the IVR Server through the CTI Connector, based on the channel identifier.

In all three IVR deployment modes, the Resource Manager and Media Control Platform communicate with IVR Server through the CTI Connector.

For more information about setting up and configuring IVR Server in the various deployment scenarios, see the VP Solution 8.1 Integration Guide.

Integration with Cisco ICM

GVP obtains call-related information, such as, the ANI and DNIS, from the initial call-setup message and uses it to fetch IVR Profiles and identify a tenant with which to associate the call. The Cisco Intelligent Contact Management (ICM) framework provides the call handling instruction, exchanges call-related data, and fetches the number of an available agent to which the call can be transferred.

The CTI Connector interacts with ICM through the Voice Resource Unit-Peripheral Gateway (VRU-PG). The PG serves as an intermediary between the proprietary interfaces that are provided by the switch and GVP (or IVR vendor), and the routing logic of the Intelligent Call Router (ICR). For ACD or PBX devices, the PG monitors real-time agent status, calculates call handling performance statistics, and forwards the appropriate event and statistical information to the Database Server.

The PG monitors and responds to routing requests from the switch and/or IVR and enables the intelligent post-routing of calls. Post-routing functions include call transfers between agents and call inter-flows between ACDs or PBXs. Figure: CTI Connector Interaction with Cisco ICM shows a simple VRU-PG configuration.

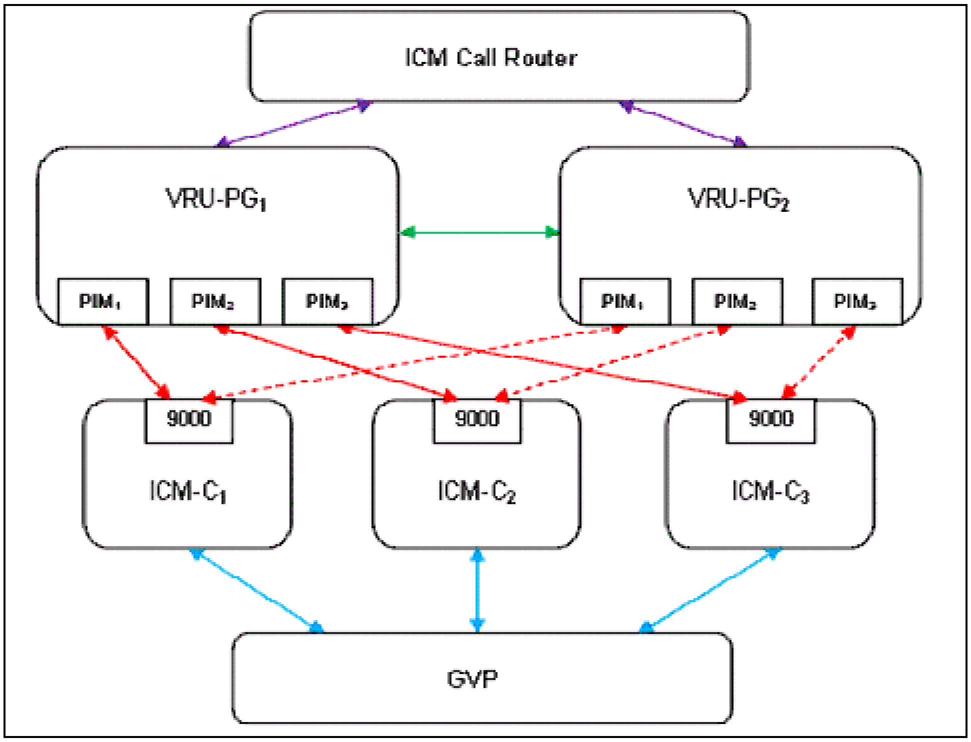


Figure: CTI Connector Interaction with Cisco ICM

CTI Connector (ICM) in Type 8 Network VRU Deployment

The Type 8 Network Voice Resource Unit (VRU) call flow that is documented here enables Cisco ICM to divert incoming calls to a VR**GVP in this case for the purpose of voice treatment such as prompting and collecting data, and for providing complete self-help voice response under ICM control before connecting the call to an agent. For GVP, this is referred to as a *pre-routed call*.

The CTI Connector (ICM) not only supports Type 8 deployment, but also allows the ScriptID or a particular Call Variable from the first script request execution to be treated as a DNIS, for invoking a VXML application. The unique ScriptID must be configured in the DID profiles to fetch the appropriate IVR profile, as a DNIS is configured. The following section describes configuration and the flow of messages for such a deployment.

CTI Connector (ICM) Configuration

Configure these parameters as follows:

- **use-cti**--Set to 1, for the Gateway resource of the Resource Manager (RM). This setting enables the RM to forward the call to CTIC without fetching the IVR Profile.
- **[ICMC] enablePreRouting**--Set to true (default is false). True specifies that the DNIS is not passed to the GVP when the call is presented. Instead, the DNIS information is retrieved through Call Variables or the ScriptID in the initial RUN_SCRIPT_REQ message.
- **[ICMC] DNISIndicator**--Set to the appropriate value either Call Variable or ScriptID, based on how DNIS information is sent.

Cisco CTI Deployment Modes

The CTI Connector and ICM can be deployed in one of two deployment modes, which are described below:

Multiple Connections

- CTI Connector supports multiple VRU-PG connection, however, separate listener ports must be specified (separated by a comma) for each VRU-PG. For example, [Tenant1] Ports=9001,9002. CTI Connector also supports multiple Peripheral Interface Managers (PIM), which are associated with a Peripheral Gateway (PG) to provide services for one tenant or multiple tenants.

Duplex Mode

- CTI Connector supports ICM in Duplex mode in which one of the VRU-PGs establishes connection to CTI Connector, while the other connection is quiescent. Once established, a connection remains active until a failure (on either side of the connection) occurs. When ICM is in * Duplex mode, it might try to open a PG when the same PG already has a session established. If this occurs, the CTI Connector terminates the existing session and processes the new request.

For information about the various deployment options that are supported by CTI Connector and ICM, see the VP Solution 8.1 Integration Guide.

For a description of how basic call flows work when CTI Connector is integrated with Cisco ICM, see GVP Call Flows.

How the PSTN Connector Works

Read here about how the Public Switched Telephone Network (PSTN) Connector performs its role in a GVP deployment.

- Operational Overview
- Signaling Protocols
- Transfer Services and Features
- Selected Features

Operational Overview

The GVP PSTN Connector is a network layer element which provides access to the core presentation layer services by using SIP. External TDM networks access the PSTN Connector through Dialogic Application Programming Interfaces (API) by using E1 Channel Associated Signaling (CAS), T1 CAS, and T1/E1 ISDN signaling protocols.

The PSTN Connector leverages subscribed transfer services, and provides many advanced inbound and outbound-calling features to support integration with TDM networks.

For a description of how the PSTN Connector processes inbound and outbound call triggers, see Basic PSTN Call Flow (Inbound) and Basic PSTN Call Flows (Outbound).

Signaling Protocols

The PSTN Connector provides interfaces for three signaling protocols Integrated Services Digital Network (ISDN), Robbed-bit signaling (RBS), and Channel Associated Signaling (CAS).

ISDN PRI

- ISDN Primary Rate Interface (PRI) provides different service offerings depending on the geographic region, for example:
    - In North America and Japan 23 B channels and 1 D channel, yield a total bit rate of 1.544 Mbps (the PRI D channel runs at 64 kbps).
    - In Europe, Australia, and other parts of the world 30 B channels and 2 64-kbps D channels yield a total bit rate of 2.048 Mbps.

The PSTN connector supports these ISDN PRI T1 and E1 protocol variants:

T1-ISDN PRI	E1-ISDN PRI
• 4ESS ISDN (tested with AT&T 4ESS switch) • NI-2 ISDN • NT-1 ISDN • 5ESS ISDN (tested with Lucent 5ESS switch) • Nortel Custom ISDN (test with Nortel DMS-100 switch)	• NE1 ISDN • NET5 ISDN • CTR4 ISDN • QSIG ISDN

For a complete list of supported specifications and standards, including ISDN PRI physical layer, see Specifications and Standards.

Robbed-Bit Signaling

- Robbed-bit Signaling (RBS) is a type of CAS that is sometimes referred to as in-band signaling. CAS signals each traffic channel instead of a single dedicated channel (like ISDN). The signaling associated with a traffic circuit is permanently associated with it. The most common types of CAS are loop start, ground start, Equal Access North American (EANA), and E&M (Ear & Mouth).

CAS also processes the receipt of DNIS and ANI information, which is used to support authentication and other functions. The PSTN Connector can be configured to restrict or pre-define the length of the DNIS and ANI on an incoming call.

The PSTN Connector supports these types of RBS:

• Line-side T1 • Group A T1 • Group B T1 • Group D T1	• wink start T1 • ground start T1 • loop start T1 • immediate start T1

Channel Associated Signaling

- The PSTN Connector supports standard E1 CAS. CAS transmits signaling information within the voice channel. CAS is configured on an E1 controller and enables the access server to send or receive analog calls. It is categorized as an out-of-band signaling method because it uses the 16th channel (or time slot).

Transfer Services and Features

The PSTN Connector supports transfer services and features in various ways. This section describes how the PSTN functions with each of these transfer types.

Dialogic Transfers

- Blind transfer The PSTN Connector performs a Dialogic Blind or Hook Flash transfer when it receives a REFER message from the Media Control Platform (through the Resource Manager) and releases all ports associated with the call. If the call is not answered or busy, it is usually routed back to GVP. By releasing the call to the PBX, the PSTN Connector is free to handle new calls without handling the transferred call progression.
- Bridge transfer The PSTN Connector supports Dialogic time slot bridging. In this case, when the two separate call legs are established between the PSTN Connector and the Media Control Platform, the media flow bridging occurs at the PSTN Connector. This prevents the latency that is generated when the media is passed to and from the Media Control Platform and allows the Full Call Recording (FCR) of media when it is passed to the Media Control Platform.

For more information about FCR, see Media Services.

AT&T Transfer Connect

- GVP acts as the redirecting party (RP), which sends calls to other locations or target parties (TP). AT&T service supports in-band (IB) and out-of-band (OOB) invocation or triggers. For OOB triggers and data forwarding, the PSTN Connector and TP must have ISDN PRI, as OOB signaling occurs on the D-channel. Transfers or redirection can be provisioned for OOB triggers only, or for OOB triggers and OOB data forwarding.
    - In-band GVP requests the redirection of an answered call by out-pulsing an in-band DTMF touch-tone command (or trigger). The call is then placed on hold by the network. The PSTN Connector role is limited to passing the in-band trigger and dial string received from Media Control Platform to the

network. The in-band transfer processing logic lies with the Media Control Platform component.
    - Out-of-band When an inbound or outbound call is established between the PSTN Connector and the AT&T network, the PSTN Connector initiates a redirection FACILITY message on the signaling channels which then disconnects the call from GVP and transfers it to the TP (regardless of the outcome of the call). The FACILITY message contains User-to-User Information (UUI) from GVP, which is passed to the TP through the network. The network send notification back to GVP in another FACILITY message.

For more information about UUI, see Inbound-/Outbound-Calling Features on page 116. To find out how Media Control Platform works with AT&T Transfer Connect, see Transfer Methods for AT&T Transfer Connect.

Two B-Channel Transfer

When a controller (or subscriber) uses more than one PRI, and transfers two calls that are using different PRIs (each controlled by its own D-channel), the controller must obtain a PRI identifier for the PRI of one of the two calls before it can request the transfer. TBCT can also send transfer notifications to the two callers, but this is optional.

The TBCT implementation is defined in Telcordia Technologies Generic Requirements GR-2865-CORE. For a complete list of supported specifications and standards, see Specifications and Standards.

Release Link Trunk Transfer

- The initial calls can be on the same or different PRI trunk groups, but if they are on different ones, they must connect to the same DMS switch.

Both the primary and secondary trunks must be configured with the RLT feature enabled. If an outbound call from the GVP is redirected to a third-party number, then that number must also be configured for RLT. If the third-party number is not configured for RLT, the switch cannot return a call ID to GVP, and the calls cannot be transferred.

Explicit Call Transfer

- ECT uses two lines to transfer a call, a primary line and an outbound line. When the outbound call reaches the alerting (ringing) state, GVP sends a request to transfer the call to the switch. When the switch accepts the request, the user is released from the calls and they are connected directly. The transfer can be configured to wait until the outbound call is answered before initiating.

Q.SIG Call Transfer

- Q.SIG uses a method of call control in which the switch-type is defined in the configuration file, because each switch vendor uses a different method of implementing the service. This method of call control is defined in EN 300 171 and EN 300 172.

Path-Replacement is a supplemental service that uses two lines, a primary line and an outbound line. Once a switch has accepted the transfer request, both calls are connected at the switch and the GVP releases both B channels. The Path-Replacement method of call transfer is defined in the ETS 300 258 and ETS 300 259 ETSI Specification.

Selected Features

This section describes some of the advanced features and functionality that are supported by the PSTN Connector for inbound and outbound calling:

Inbound-/Outbound-Calling Features

CTI Connector Integration

- The PSTN Connector supports CTI integration with IVR Server in front or behind the switch, and with GVPi and NGI. It passes Dialogic port information to the CTI Connector in SIP custom headers.

Port Management

- PSTN Connector ports can be configured to accept inbound calls, outbound calls, or both. When the Type parameter is set to In/Out and glare occurs, priority is given to inbound calls and outbound calls are given a predefined number of retries.

User-to-User Information

- ISDN PRI User-to-User information enables a user to send information to the network which can then be transferred to a remote user. GVP can send or receive this UUI in a single call leg or transfer it end-to-end from the * incoming call to the outbound call during a transfer. The PSTN Connector transmits UUI from incoming and outbound calls by using the SIP `X-Genesys-GVP-UUI` custom header in the INVITE or REFER (if transferred) messages to and from the Media Control Platform.

The PSTN Connector supports ISDN codeset when UUI is propagated during a transfer. The code set mechanism enables different geographic areas to use their own nation-specific information elements within the data frames.

Presentation and Screening Indicators

- When the call is inbound, the PSTN Connector extracts the Presentation and Screening Indicators, Numbering Plan, and Number Type from the Calling Party Number IE of the ISDN call set-up message if it is supported by the network. The PSTN Connector then propagates this information to the Media Control Platform in the Remote-Party-ID header of the INVITE message. When the call is outbound, this information is extracted from the outbound SIP INVITE message that is sent by the Media Control Platform and the PSTN Connector updates the IE appropriately.

AT&T Coda Extensions

- When a call is inbound from an AT&T network, the PSTN Connector extracts Billing Number and Information Indicator Digits from the ISDN call set-up message and propagates this information to the Media Control Platform in the `X-Genesys-ATT-CODA` custom header of the SIP INVITE message. When the call is outbound, the PSTN Connector extracts this information from the outbound INVITE custom header that is sent by the Media Control Platform and propagates it to the TDM network.

Inbound-Calling Features

ISDN Alerting

- This calling feature is enabled (by default) during inbound call setup to avoid delays in answering calls. The PSTN Connector can be configured to enable or disable this functionality by setting the DisableISDNAlerting parameter to True or False.

Overlap Receive DNIS/ANI

- Some PSTN network switches send DNIS and ANI in overlap-in-band mode. After the call is established, the PSTN Connector waits for this information for a predefined period of time and sends it on to the Media Control Platform (through Resource Manager) in a SIP INVITE message to initiate a dialog. This functionality can be configured for T1 RBS, T1-ISDN, E1 CAS, or E1-ISDN by setting the OverlapReceivedEnable to True or False. It is disabled by default.

Redirecting Number from IE

- The PSTN Connector extracts the Redirecting Number (RN) from the RN Information Element (IE) of an inbound ISDN call set-up message if it is supported by the network. The RN IE identifies the number from which a call is diverted or transferred. This feature is optional and controlled by the network. The PSTN Connector extracts the RN, reason, and original called * number (OCN) and propagates this information to the Media Control Platform by using the History-Info header of the SIP INVITE message. If the RN IE is not available, the PSTN Connector returns an empty string.

Outbound-Calling Features

Disconnect Cause Propagation

- When the PSTN Connector disconnects a call from GVP, it propagates the cause to the PSTN network by using a SIP error response code For example, ISDN=111 (Protocol Error) is propagated with a SIP 400 (Bad Request) response code. For a list of ISDN DISCONNECT messages that map to SIP error response codes, see the Genesys Voice Platform User s Guide.

Call-Progress Analysis

- During outbound-call initiation, the PSTN Connector might receive a request from the remote party to detect call-progress events on the media stream. If this occurs, the Media Control Platform enables Call Progress Analysis (CPA) and responds with a SIP 200 OK message which contains a list of supported events (in the X-Detect header) that can be detected. The PSTN Connector then sends notification of detected events to the Media Control Platform in SIP INFO messages.

The newly defined X-Detect SIP header in the INVITE (200 OK) message indicates a request or response. The request includes a list of event types that the remote party wants notification of, for example, X-Detect:Request=CPT,FAX. The response includes a list of event types that the PSTN Connector is able to detect, for example, X-Detect:Response=CPT,FAX. If the X-Detect header is not in the request the PSTN Connector proceeds as though CPA is not required.

The X-Detect header can only be used while the SIP dialog is being established. After that, detection capabilities are determined and cannot be changed.

Table: CPA Categories and Sub-types as Supported with Dialogic

Type	Subtype	Previous Subtype Name *(for backward compatibility)*	Supported with Dialogic	Supported with CPD Library
AMD	AUTOMATA	Automatic	Yes	Yes
CPT	NoRingBack	No Ring Back	Yes	Yes
	BUSY	Busy	Yes	Yes
	SIT-RO	ReOrder	Yes	Yes
	Not-In-Service	Not-in-Service	No	Yes
	SIT-IC	Operator Intercept	Yes	Yes
	NoDialTone	No Dial Tone	Yes	Yes
	UnAllocatedNumber	UnAllocated Number	No	Yes
	SIT-VC	Vacant Circuit	Yes	Yes
	SIT	Unknown SIT	No	Yes
	NoAnswer	NoAnswer	Yes	Yes
FAX	CED (FAX1)	CED (FAX1)	Yes	Yes
	CNG (FAX2)	CNG (FAX2)	Yes	Yes
PVD	VOICE	Voice	Yes	Yes
	Cadence	Cadence	Yes	Yes
	LoopCurrent	LoopCurrent	Yes	Yes

How the Media Control Platform Works

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

- Operational Overview
- Media Services
- Speech Services
- Transfers

- Conferencing
- Debugging VoiceXML Applications
- HTTP Basic Authentication

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

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

Operational Overview

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

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

Key and Certificate Authentication

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

Network Traffic Partitioning

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

Inbound Calls

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

1. The Media Control Platform, acting as a SIP User Agent Server (UAS), receives a SIP INVITE from the Resource Manager. Because the Resource Manager modifies the SIP request by inserting service prerequisites for the 1. IVR Profile, the SIP Request-URI includes a voicexml parameter that specifies the URL of the initial page of the required VoiceXML application. Alternatively, the Media Control Platform can be configured (in the sip.vxmlinvite configuration option) to accept calls in which the originator specifies the initial VoiceXML URL in the Request-URI of the SIP INVITE. In these cases, provided that the syntax and format of the Request-URI are correct, the normal Resource Manager method of mapping calls to IVR Profiles will be bypassed.
    - The Media Control Platform recognizes the following Request-URI formats:
        - `sip:dialog.vxml.<URL>@host.com`, where the URL portion must be properly encoded (`draft-rosenberg-sip-vxml` format)
        - `sip:<user>@host.com;voicexml=<URL>` (NETANN dialog service format)
        - `sip:conf=<ID>@host.com` (NETANN format, for calls to join a specified conference without going through a VoiceXML application)
        - `sip:msml[=conf_id]@ms.example.com[;uri-parameters]`, where the request is for MSML service when a conference call is created by MSML. The join request is transmitted to the Media Control Platform in a separate SIP INFO message. The conference ID in the SIP Request URI indicates to the Resource Manager that the callers for this conference must be routed to the same Media Control Platform.
    - The Media Control Platform supports the following service parameters in the Request-URI:
        - **voicexml** The value must conform to the URI syntax that is defined in RFC 3986.
        - **maxage** The value must be all digits.
        - **maxstale** The value must be all digits.
        - **method** The value must be either get or post.
        - **postbody** The HTTP body for POST requests.
        - **timeout** The value must be numeric.

- **gvp.alternatevoicexml** Specifies an alternative VoiceXML page if the VoiceXML interpreter fails to fetch the primary page.
- **gvp.config.<parameter name>** Sets the values of certain platform configuration options for the duration of the media session. This mechanism enables an IVR Profile to override certain Media Control Platform configuration parameters, for the session that is being executed in the context of this VoiceXML application. For the configuration parameters whose values can be set dynamically, see the Media Control Platform reference information appendix in the Genesys Voice Platform 8.5 User's Guide.

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

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

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

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

Tip

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

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

3. The Media Control Platform passes all the generic SIP Request-URI parameters to the VoiceXML interpreter.
4. The VoiceXML interpreter sends an HTTP/HTTPS or file retrieval request to the Fetching Module to fetch the initial page. The request includes the timeout, maxage, and maxstale values, if present, to determine whether the fetch can be satisfied from the cache store. For more information about how caching is used to improve Media Control Platform performance, see Caching.
5. The VoiceXML interpreter compiles and interprets the initial page, and all subsequent pages, so that the Media Control Platform can execute the application. The VoiceXML application is ready to proceed when the VoiceXML document is fetched, parsed, and compiled. The NGI supports the following encodings for VoiceXML pages and external ECMAScripts objects:
    - UTF-8
    - UTF-16
    - ISO-8859-x
    - Far-East encoding for Japanese, Chinese, and Korean The VoiceXML interpreter retrieves the encoding information for a document from the encoding attribute of the XML header, or from the charset attribute of the <script> tag.
6. At the same time that it passes SIP INVITE information to the VoiceXML interpreter (see Step 3), the Media Control Platform passes the SDP to the Media Server, so that it can negotiate media capabilities. For information about capability negotiation, and the codecs that the Media Control Platform supports, see Codec Negotiation on page 129. For information about the file formats that are supported for playing and recording audio and video for various codecs, see the Media Control Platform reference information appendix in the Genesys Voice Platform 8.5 User's Guide.
7. When the VoiceXML application is ready to be executed, the Media Control Platform sends a 200 OK response to the initial INVITE request. The response includes the Media Server SDP answer, if applicable. If the initial INVITE and the ACK that is returned do not contain the required SDP information, a media-less dialog is established. In general, the VoiceXML application starts when the 200 OK response is acknowledged (that is, when the platform receives an ACK). However, it is the VoiceXML application itself that determines whether it is ready to start. In particular, if a media-less dialog has been established, the VoiceXML application will not start until the platform receives a re-INVITE that includes the SDP information for the caller.
8. When the VoiceXML application starts, it controls the session. The VoiceXML interpreter is responsible for driving the Media Control Platform to execute the

VoiceXML application appropriately. The NGI performs speech and DTMF recognition, and it issues commands to the platform to execute call and media operations.

- The platform sends and receives SIP INFO messages for the following application events:
    - To make a request The application can specify the content type and content body of the SIP INFO message.
    - To send or receive data The application can send data in custom SIP headers. The platform can send information that it receives in SIP INFO headers to the NGI, and this information is provided to the application in shadow variables. Note that when the SIP INFO content type is application/dtmf-relay, it is treated as DTMF input instead of an application event. The content format is `Signal = <digit>`.
    - The application uses dialogs to initiate transfers as required. For more information about how the Media Control Platform performs transfers, see Transfers. The NGI supports use of the userdata attribute in the <transfer> tag, to abstract CTI data. The NGI exposes CTI userdata to the application in a session variable, session.com.genesyslab.userdata.
    - The platform provides media services through the Media Server, for operations such as playing prompts and recording audio and video. For more information, see Media Services .
    - For ASR or TTS, the Media Control Platform controls speech resources through the MRCP Client. For more information, see Speech Services.

9. The VoiceXML application can invoke other VoiceXML applications. The VoiceXML interpreter is responsible for issuing commands to the Fetching Module to fetch VoiceXML pages and other applications.
10. When a caller disconnects (that is, when a BYE is received), the platform notifies the Voice XML application (through the connection.disconnect.hangup event). If the BYE includes a Reason header, the value of the Reason header is passed verbatim to the application. If the application disconnects, the platform generates a BYE request.
11. For each VoiceXML session, the Media Control Platform generates call-detail records, which it sends to the Reporting Server. For more information, see CDR Reporting.
12. For each VoiceXML session, the Media Control Platform sends logs and metrics (VoiceXML application event logs) to the log sinks and, from here, to the Reporting Server. For more information about metrics, see Metrics. For descriptions of the Media Call Control Platform metrics, see the Genesys Voice Platform 8.5 Metrics Reference.

Media Services

The Media Server provides the following services:

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

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

Selected Features

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

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

Tip

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

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

Dual-Channel Call Recording

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

Dual RTP Streams

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

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

HA for Clients

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

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

Call-Progress Detection and Analysis

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

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

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

Detection Methods

The Media Control Platform supports two methods of CPD:

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

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

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

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

Analysis Logging

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

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

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

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

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

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

- `cpa.enable_log_param`
  - Global Call ID
  - Start time of detection
  - Configuration information that was used in detection.
- `cpa.enable_log_result`
  - Global Call ID
  - Time that the detected event was reported.
  - Detected CPD results

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

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

Log Format

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

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

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

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

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

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

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

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

Example 1:
```
cpa_tone_setting busy=na_busy
```

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

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

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

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

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

For example: `cpa_result Answering machine detected`

Overwriting CPA Configuration Options

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

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

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

Value: `fixed,30000`

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

Codec Negotiation

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

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

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

Tip

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

The Media Control Platform can support the following codecs:

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

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

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

Tip

In certain media operations, such as Conference, CPD/CPA require internal transcoding to a native primitive codec L16. Therefore, the codec being used must be

> added to the `mpc.transcoders` configuration option in the Media Control Platform Application. If the default IVR Profile has specific transcoding disabled, operations that require that it will fail.

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

SDP Negotiation for Telephony Events

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

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

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

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

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

MSML-Based Media Services

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

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

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

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

Table: Media-service Values and MSML Service Types

media-service value	MSML service type
treatment	Used for any of the following treatment types, as asked for in the initiating RequestApplyTreamtent: ◦ Music ◦ Silence ◦ CollectDigits ◦ PlayAnnouncement ◦ PlayAnnouncementAndDigits ◦ PlayApplication (including VoiceXML) ◦ RecordUserAnnouncement
cpd	Used for Call Progress Detection (CPD) for outbound calls.
record	Used for both static (DN-level configured) and dynamic (T-Library initiated) call recording.

conference	Used for conferences or for supervisor monitoring.
media	Used for the following services: • greetings (static or dynamic) • music-on-hold, music-in-queue • ringback or busy tone • nailed-up connections • third party in push video-scenarios • RingBack, Busy, FastBusy, as initiated by RequestApplyTreatment

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

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

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

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

Speech Services

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

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

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

Grammars

The Media Control Platform generates the following grammars:

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

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

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

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

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

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

Transfers

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

Transfer Types

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

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

Whisper Transfer

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

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

AT&T Transfer Types

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

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

Tip

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

Transfer Methods

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

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

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

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

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

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

This is a one-leg transfer.

BRIDGE The Media Control Platform bridges the media path:

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

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

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

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

This is a two-leg transfer.

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

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

6. The transfer fails if a non-2xx final response is received for the initial INVITE request.

This is a two-leg transfer.

Transfer Method for NEC NEAX 61 Switch

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

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

Transfer Methods for AT&T Transfer Connect

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

- ATTCOURTESY, ATTCONSULT, ATTCONFERENCE These inbound-call transfers are treated like any other inbound transfer. The Media Control Platform sends a request to the gateway to trigger the DTMF transfer, and the PSTN Connector passes the transfer information to the network through Dialogic ports:
- ATTOOBCOURTESY The PSTN Connector receives an outbound trigger or request from the network through Dialogic ports:
  The PSTN Connector sends an INVITE to the platform.
  - The platform initiates call setup and the PSTN Connector sends a `gc_AcceptCall` message to the AT&T network.
  - When the call is established, the platform sends a `200 OK` message to the PSTN Connector.
  - The PSTN Connector response with an ACK response and the two way media session is established.
  - The platform sends a REFER message that includes the `X-Genesys-Transfer-Method=ATTOOBCOURTESY` and the `X-Genesys-GVP-UUI` custom headers to the PSTN Connector.

        - The PSTN Connector sends a `202 Accepted` response to the platform and then sends a FACILITY message to the network with the target party number and UUI.
        - After the network passes on the information to the target party, it sends a FACILITY message to the PSTN Connector with the transfer results (success or failure).
        - The PSTN Connector passes this information to the platform with a NOTIFY message.
        - The call is disconnected on the PSTN side, and the platform issues a BYE message to the PSTN Connector.
        - The PSTN Connector responds with `200 OK`, and the call is released.
    - ATTOOBCONSULT The PSTN Connector receives an outbound trigger or request from the network through Dialogic ports:
        - The PSTN Connector sends an INVITE to the platform.
        - The platform initiates call setup the PSTN Connector sends a `gc_AcceptCall` message to the network.
        - When the call is established, the platform sends a `200 OK` message to the PSTN Connector.
        - The PSTN Connector response with an ACK response and the two way media session is established.
        - The platform sends a REFER message that includes the `X-Genesys-Transfer-Method=ATTOOBCONSULT` and the `X-Genesys-GVP-UUI` custom headers to the PSTN Connector.
        - The PSTN Connector sends a `202 Accepted` response to the platform and then sends a FACILITY message to the network with the target party number and UUI.
        - After the network passes on the information to the target party, it sends a FACILITY message to the PSTN Connector with the transfer results (success or failure).
        - The PSTN Connector passes this information to the platform with a NOTIFY message (including the Return Code).
        - The platform responds with a `200 OK` message.
        - The call is disconnected on the PSTN side, and the platform issues a BYE message to the PSTN Connector.
        - The PSTN Connector responds with `200 OK`, and the call is released.
    - ATTOOBCONFERENCE The PSTN Connector receives an inbound trigger or request from the network through Dialogic ports:.
        - The PSTN Connector receives an INVITE from the platform that includes the `X-Genesys-Transfer-Method=ATTOOBCONFERENCE` and `X-Genesys-GVP-UUI` custom headers, and the Call ID.

- As the platform is initiating call setup, the PSTN Connector sends a FACILITY message for redirection to the network that includes the target party number and the UUI.
- The network responds with a FACILITY ACK, and the PSTN Connector sends a 180 Ringing message, followed by a `200 OK` message to the platform.
- The platform sends an RTCP JOIN packet which enables the PSTN Connector to retrieve the caller on hold by sending a FACILITY message to the network.
- With the caller on hold, the PSTN Connector mutes the caller leg to the platform and activates media on the agent leg to the platform, and then enables whispering to the agent.
- The caller is taken off hold and the PSTN Connector mutes the agent leg to the platform, and the activates media on the caller leg to the platform.
- When the transfer is completed successfully, the PSTN Connector is connected to the target party and the FACILITY ACK from the network indicates success.
- The PSTN Connector disconnects the call as soon as the session termination is received from the platform.

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

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

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

Table: SIP Transfer Methods and Supported VoiceXML Transfer Types

SIP transfer method	Supported transfer type	Notes

HKF	◦ Blind ◦ Consultation Whisper transfer supported	◦ The DTMF digits are flash or other configured digits, followed by a phone number. ◦ A different configured sequence of flash and digits can be dialed to abort the transfer.
REFER	◦ Blind ◦ Consultation	◦ The default platform method for `type=blind`. ◦ Transfer connect timeout is not supported. ◦ The platform can be configured to send an INVITE hold to the caller. ◦ For type=consultation, the platform also supports the NOTIFY method for notification of the transfer result. If the transfer fails, the platform takes the original caller off hold, and the VoiceXML application proceeds with the caller. **Note:** Transfer audio does not work with the method REFER. ◦ The platform can be configured to send a BYE request to the caller, or to wait for a BYE from the caller. ◦ For transfer requests from the Call Control Platform, the Media Control Platform sends a REFER request to the Call Control Platform, which throws a dialog.transfer event to the CCXML application. The type attribute for the event is always set to blind, whether the request from the VoiceXML application was for a blind transfer or consultation (supervised) transfer.

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

		◦ For a whisper transfer, if the called party rejects the transfer request, the platform sends a BYE to the called party to disconnect the call. ◦ Non-whisper transfers support the connectwhen=immediate attribute in the VoiceXML application. If this value is specified, the media path is established between the caller and the called party as soon as the media session is ready.
MEDIAREDIRECT	◦ Blind ◦ Consultation ◦ Bridge Whisper transfer supported	◦ For a whisper transfer, a media channel is established between the called party and the Media Control Platform for the consultative part of the transfer, if necessary. If the called party rejects the request, the platform sends a BYE request to the called party to disconnect the call. The media path and interaction between the platform and the caller then resumes. ◦ If the caller disconnects during the transfer (that is, if the platform receives a BYE), the platform sends a BYE to the called party to disconnect the call. ◦ If the called party disconnects during the transfer, the platform updates the caller s media path back to the platform, using a new re-INVITE if necessary.

Table: PSTN Transfers and Supported VoiceXML Transfer Types

PSTN transfers	SIP transfer method	Supported transfer type	Notes
AT&T Courtesy Transfer	ATTCOURTESY ATTOOBCOURTESY	◦ Blind	◦ In-band and out-of-band transfers are supported. ◦ User-to-User Information (UUI) message in transfer supported.
AT&T Consult Transfer	ATTCONSULT ATTOOBCONSULT	◦ Blind ◦ Consultation	◦ In-band and out-of-band transfers are supported. ◦ UUI message in transfer is supported for out-of-band only. ◦ Transfer audio is not supported.
AT&T Conference Transfer	ATTCONFERENCE ATTOOBCONFERENCE	◦ Blind ◦ Consultation ◦ Bridge Whisper transfer supported	◦ In-band and out-of-band transfers are supported. ◦ UUI message in transfer is supported for out-of-band only. ◦ Whisper transfer is supported when type= consultation.
Dialogic Bridge Transfer	BRIDGE	◦ Blind ◦ Consultation ◦ Bridge Whisper transfer supported	◦ UUI message in transfer is supported.
Dialogic Blind Transfer	REFER	◦ Blind	◦ UUI message in transfer is supported ◦ When `type=blind` and `connectwhen=immediat` the REFER message is se after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.

Single B Channel Transfer over ISDN (for NEC NEAX 61 switch)	NEC61ISDN	◦ Blind	◦ UUI message in transfer is supported.
Two-B Channel Transfer (TBCT)	REFERJOIN	◦ Blind ◦ Consultation Whisper transfer supported	◦ UUI message in transfer is supported. ◦ When type=blind and connectwhen=immediate the REFER message is se after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.
Release Link Trunk (RLT) Transfer	REFERJOIN	◦ Blind ◦ Consultation Whisper transfer supported	◦ UUI message in transfer is supported. ◦ When type=blind and connectwhen=immediate the REFER message is se after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.
Explicit Call Transfer (ECT)	REFERJOIN	◦ Blind ◦ Consultation Whisper transfer supported	◦ UUI message in transfer is supported. ◦ When type=blind and connectwhen=immediate the REFER message is se after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.

Q Signaling (Q.SIG) Transfer	REFERJOIN	◦ Blind ◦ Consultation Whisper transfer supported	◦ UUI message in transfer is supported. ◦ When type=blind and connectwhen=immediate the REFER message is se after the outbound call receives alerting. For any other combination, the REFER with Replaces message is sent.

Table: PSTN Connector- and NGI-supported Transfers

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

Transfer (out-of-band)					
AT&T Conference and Transfer (out-of-band)	ATTOOBCONFERENCE	blind / consultation	N/A	Yes	Yes
Single B channel blind transfer over ISDN for NEC NEAX 61 switch	NEC61ISDN	blind	N/A	No	Yes
Two B-Channel Transfer (TBCT)	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes
Release Link Trunk Transfer (RLT)	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes
Explicit Call Transfer (ECT)	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes
Q.SIG Call Transfer and Path Replacement	REFERJOIN	blind / consultation	immediate / answered	Yes	Yes

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

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

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

Implications of Transfer Method Transfer Type Combinations

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

IPv6 Support in Inbound and Outbound Transfers

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

Conferencing

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

NETANN

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

Platform-level configuration options (in the conference configuration section) support standard NETANN conference requirements, such as configurable participant roles (talk-only, listen-only, or full duplex), audio-gain parameters, and the video-output algorithm (first, loudest, or no video).

MSML

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

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

Debugging VoiceXML Applications

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

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

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

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

HTTP Basic Authentication

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

How the Call Control Platform Works

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 by using other kinds of SIP messages Transfers. The platform can also initiate outbound calls by sending SIP INVITE requests either through the Resource Manager or directly to the destination.

The platform can also receive requests for CCXML sessions directly from an external HTTP client.

For more detailed information, see the 8.5 CCXML Reference.

Incoming Connections

The Call Control Platform handles service requests for incoming connections in a typical call flow, 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.
3. The Call Control Platform can accept, reject, join, or redirect the connection. Configuration options enable you to customize some of the SIP responses that the Call Control Platform sends to the Resource Manager for the events. For more information, see the section about customizing SIP responses in the GVP 8.5 User's Guide.
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 by inserting service prerequisites for the IVR Profile, the SIP Request-URI includes a CCXML parameter, which specifies the URI 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 (in the ccpccxml.default_uri configuration option).
    - Call Control Platform HTTP requests comply with HTTP 1.1. For HTTPS, the Call Control Platform supports HTTP over Secure Socket Layer (SSL) 3.0 and HTTP over 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.
    - The following parameters are also supported in the HTTP/HTTPS fetch:
        - `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 (?).

5. The CCXMLI compiles and interprets the initial page, and all subsequent pages, so that the Call Control Platform can execute the application.
    - A CCXML page may transition to another page as it is executed, 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.
    - The Call Control Platform supports receiving DTMF events in SIP INFO messages, and it 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 the 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.
7. For each CCXML session, the Call Control Platform generates call-detail records, which it sends to the Reporting Server. For information about the CDR attributes, see CDR Reporting.
8. For each CCXML session, the Call Control Platform sends logs and metrics (CCXML application event logs) to the log sinks and, from there, to the Reporting Server. For more information about metrics, see Metrics. For descriptions of the Call Control Platform metrics, see the GVP 8.5 Metrics Reference Guide. In GVP 8.1 and above the Call Control Platform supports Operational Reporting (OR).
9. If it is configured to do so (see the description of the `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 place outbound calls by starting a new CCXML session, if it receives a session creation request directly from an HTTP client. Alternatively, it can place an outbound call within the context of an existing session.

The CCXML application uses the `<createcall>` tag to create the connection, and it specifies the destination of the call (in the dest attribute) as a SIP URI. The value of the dest attribute is used in the Request URI that the Call Control Platform sends to the Resource Manager to place the call. The Resource Manager, in turn, forwards the request either to another SIP Proxy that has been configured in a route set for the Call Control Platform, or to the SIP Server.

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

If an outbound call (connection or conference leg) is not joined to any call and the user does not set the caller information, the caller URI is configurable.

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 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 the section about configuring device profiles in the GVP 8.5 User's Guide.

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, using the 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* below).
- 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* below).

Default Device Profiles

By default, VP Call Control Platform 8.5 is provisioned with the following device profiles for SIP devices, by order of precedence:

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

For the property values that have been defined for the preprovisioned device profiles, see the GVP 8.5 User's Guide.

If your deployment uses SIP devices that are not adequately represented by the default device profiles, you must either provision additional device profiles or modify an existing device profile. For more information, see the section about configuring device profiles in the GVP 8.5 User's Guide.

How the Fetching Module Works

The Media and Call Control Platforms use the Fetching Module to fetch documents and perform caching. The Fetching Module maintains a high-performance in-memory cache and interfaces with the on board Squid Caching Proxy.

In GVP 8.1.2 and later releases:

- Fetching Module is no longer a separate component (integrated with the Media and Call Control Platforms).
- Squid is an optional component.

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: Caching

- Non-HTTP/1.1-Compliant Caching
- HTTP/1.1-Compliant Caching
- Squid Configuration File
- Squid Log Files

Caching

Unlike visual browsers, there are no end-user controls in the VoiceXML interpreter (the NGI) context to enable stale content to be updated or refreshed. Instead, the VoiceXML document itself enforces cache refreshes, 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.

Tip

The legacy VoiceXML interpreter, GVPi, does not use the Fetching Module to fetch documents or perform caching; instead it uses the Page Collector module. For a description of how the Page Collector fetches and performs caching, see Page Collector Caching on page 48.

Non-HTTP/1.1-Compliant Caching

The Fetching Module caches documents in-memory, in accordance with configurable maximum age and URL substring parameters (the iproxy.cache_max_age, iproxy.cache_error_max_age, and iproxy.no_cache_url_substr configuration parameters).

HTTP/1.1-Compliant Caching

The GVP 8.1.1 and earlier 8.x releases of the Fetching Module use a caching proxy (Third-Party Squid) for HTTP/1.1-compliant caching. Although the caching proxy generates HTTP/1.0 requests, it supports HTTP/1.1 caching functionality.

Starting in GVP 8.1.2, the Fetching Module is integrated with the Media Control and Call Control Platforms and is HTTP/1.1-compliant. In addition, the Squid Caching Proxy is optional.

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 by using the Cache-Control and Expires HTTP response headers.
- If these headers are not present, the Fetching Module (or, in GVP 8.1.1 and earlier 8.x releases, Squid) uses heuristics to generate expiry times.
- The application developer can 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 through the Media Control Platform or Call Control Platforms (or, in GVP 8.1.1 and earlier 8.x releases, Squid) configuration.

Caching Behavior

The primary effect 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 none, 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 present in the response, it is used to control expiration times, and it overrides an Expiry time if it is specified.

Note: If the policy requires a fetch from the server, it is an optimization to perform a get if modified (the request includes an If-Modified-Since [IMS] header) on a document that is still present in the cache. Squid performs 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 nonzero value means that the Fetching Module might be forced to get a fresh copy of a resource that may not yet have expired in the cache. Setting maxage to zero (0) means that Squid is unconditionally forced to get a fresh copy.
- Using `maxstale` enables the application developer to specify that an expired copy of a resource that is not too stale (according to the rules of HTTP/1.1) can 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.

> Tip
>
> Like other caching proxies that support maxage and maxstale, the Fetching Module 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 might 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 concept of resource age, the interpreter context computes the age of a resource from the time that it was received. If the protocol does not support the concept of resource staleness, the interpreter context treats the resource as having expired immediately upon receipt.

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

Table: Using maxage and maxstale Attributes

Desired behavior	maxage	maxstale	Notes
Client control over expiry	**<desired_expiry>**	0	• Caching is based on the Expires header. • Refetch is based on maxage and maxstale. • Uses IMS.
Expired document acceptable	**<large_value>**	**<desired_maxstale>**	• Caching is based on the Expires header. • Refetch occurs 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; therefore, the maxage and maxstale parameters have no meaning for that page. However, they do have meaning for the initial root page.

Configuration parameters `vxmli.initial_request_maxage` and `vxmli.initial_request_maxstale` in the Media Control Platform set the values of the `maxage` and `maxstale` parameters for the initial root page. 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 expiration information as part of a Cache-Control header (using maxage/maxstale), this information will be used to control cache expiry.

Expiration Model

The Fetching Module uses a refresh-rate model, instead of 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, the Fetching Module checks freshness requirements when objects are requested:

- If an object is fresh, it goes directly to the client.
- If an object is stale, an If-Modified-Since request is made and sent to the web server.

For information about how to use HTTP caching to improve performance, see Optimizing Performance through HTTP Caching.

Squid Configuration File

The Squid configuration file (`C:\squid\etc\squid.conf` for Windows and `<Directory>/etc/squid/squid.conf` for Linux) controls configuration of the caching proxy. The configuration file is a text file that contains pairs of keywords and values (with no equal sign [=] between them). For example, the following pair defines port 3128 as the TCP port that the caching proxy will use for receiving requests:

```
http_port 3128
```

In general, the default Squid configuration file should be suitable for most installations. However, you might need to modify it 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 want to change the Squid defaults for the expressions that Squid tries to match in SIP request-URI headers to control refresh behavior.
- You need to configure nonstandard safe ports or SSL ports for HTTP and SSL.

For more information about modifying the Squid configuration file, see the section about configuring the Squid caching proxy in the Genesys Voice Platform 8.5 User's Guide.

For detailed information about all Squid configuration items, see the Squid Configuration Guide at the ViSolve Squid Support Service website.

Changes to the Squid configuration file do not take immediate effect in the running configuration.

Squid Log Files

The caching proxy logs can provide useful information to help you identify performance issues or resolve VoiceXML or CCXML application problems.

Access Logs

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

```
C:\squid\var\logs\ (Windows) or /var/log/squid (Linux).
```

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 file has ten fields. A single hyphen (-) indicates unavailable data.

For detailed information about the fields in the Squid access.log file, see the caching reference information appendix in the Genesys Voice Platform 8.5 User's Guide.

For information about how to schedule log rotations and manage the cache manually, see Managing the Cache.

How the MRCP Proxy Works

This section provides information about the following topics, to explain how the MRCP Proxy (MRCPP) Server performs its role in a GVP deployment:

- Operational Overview
- Resource Management
- High Availability
- Data Collection and Logging
- Support for Multiple Speech Servers

Operational Overview

MRCP Proxy accepts client requests (from Media Control Platform) and sends requests to ASR and TTS speech servers by using MRCPv1.

MRCP Proxy also supports a subset of RTSP over persistent TCP connections for client or server interactions. MRCPv1 does not require full support of RTSP, therefore, SETUP, TEARDOWN, DESCRIBE, and ANNOUNCE requests only are supported.

Resource Management

MRCP Proxy obtains a list of MRCPv1 resources from Management Framework to maintain an up-to-date picture of the resource pool. ASR and TTS speech server are added as connections in the MRCP Proxy Application object and become the resource access point. The MRCP Proxy uses the information that is configured in the provision section of the speech resource Application object to determine how the requests for resources will be routed.

Resource Load Balancing

MRCP Proxy supports round-robin load balancing for eligible speech resources in the deployment. Eligibility is determined in the following manner:

- MRCP Proxy sends ping messages (RTSP DESCRIBE) regularly to the speech resources.
- If the ping fails, the speech resource is isolated and set to the unavailable state. Requests are not routed to that resource until the resource becomes available. The

time between ping requests is controlled by the provision.vrm.proxy.ping_interval parameter in the ASR or TTS resource configuration option.
- Only the speech resources with the engine name (for example, provision.vrm.client.resource.name) that are requested by the Media Control Platform are considered for matching.

Resource and Application Updates

The MRCP Proxy receives and processes periodic updates from Management Framework for its configured Application objects and resources in the following way:

- If it receives an update from the Management Framework for its own Application object, all changes to the vrmproxy.timeout.* parameters are accepted and the MRCP Proxy uses the new timeout values.
- If it receives an update that a new connection is added to its Application object, and the connected object is identified as an ASR or TTS resource, the resource is added to its resource pool.
- If it receives an update that an existing connection is deleted from the its Application object, and the connected object is identified as an ASR or TTS resource, the resource is deleted from the resource pool.
- If it receives an update that the provision section of an ASR or TTS resource object has changed, it takes effect immediately.

High Availability

MRCP Proxy can be deployed in HA warm active standby mode. Two servers are required for this configuration one configured as the primary server and the other as the backup server. Management Framework s Solution Control Server determines which server is active and which is on standby at any given time.

The standby MRCP Proxy puts itself into suspended mode and does not submit data to the Reporting Server nor does it respond to incoming RTSP requests; only the active server performs these functions.

When failover occurs, the existing TCP connections are terminated and all of the existing ASR/TTS sessions are lost. In addition, when the standby machine becomes active, the peak ASR and TTS usage counter is reset to zero.

Data Collection and Logging

MRCP Proxy uses the Operational Reporting (OR) interface to send ASR and TTS usage data to the Reporting Server in real time. It submits the following usage information:

- ASR and TTS peak data for a specific speech resource
- ASR and TTS arrival data for a specific speech resource
- ASR and TTS peak data for a specific tenant
- ASR and TTS arrival data for a specific tenant
- ASR and TTS peak data for a specific IVR Profile
- ASR and TTS arrival data for a specific IVR Profile
- ASR and TTS peak data for the entire deployment

MRCP Proxy reports ASR and TTS usage data for tenants, IVR Profiles, or the entire deployment. The MRCP Proxy receives the tenant and profile information from the Media Control Platforms for each speech resource request. This information is sent to the Media Control Platform originally from the Resource Manager, based on the tenant/profile mapping.

Support for Multiple Speech Servers

MRCP Proxy supports Primary and Backup speech server lists, based on the configuration.

- MRCP Proxy will route the request to one of the primary speech servers in round-robin fashion.
- If none of the speech servers listed in the primary group is available, then MRCP Proxy will route the request to one of the backup speech servers.

MRCP Proxy supports connecting Speech Servers in the connection tab. See the procedure Adding a Speech Server as Primary or Backup.

How the Supplementary Services Gateway Works

Read here about how the Supplementary Services Gateway (SSG) performs its role in a GVP deployment:

- Operational Overview
- Requests and Responses
- Asynchronous Result Notifications
- Call Initiation Through SIP Server
- Call-Progress Detection
- Port-Availability Notifications
- Persistent Storage
- Processing Requests
- Database Cleanup

Operational Overview

The Supplementary Services Gateway receives requests for outbound-call initiation from third-party Trigger Applications (TA). The requests are validated and placed in persistent storage in the Supplementary Services Gateways external database. If the outbound call succeeds (or fails after specified number of retries), the Supplementary Services Gateway notifies the trigger application with a result notification URL in the form of an HTTP request (which the trigger application includes in the call initiation request).

Embedded HTTP Server

The Supplementary Services Gateway has an embedded HTTP server which communicates with the trigger application to service HTTP GET, POST, and DELETE requests.

The embedded HTTP server supports both HTTP and HTTPS over Secure Socket Layer version 1 (SSLv1), SSL version 2 (SSLv2), and Transport Layer Security version 1 (TLSv1).

The default page or identifier for the embedded HTTP server is SSG, and is configured in Genesys Administrator with the HTTPDefaultPage parameter in the http section of the Supplementary Services Gateway Application object. HTTP or HTTPS URIs targeted for the Supplementary Services Gateway use the following format:

```
http(s)://<ssg host>:<http/https port>/SSG?...
```

Secure Communications

The Supplementary Services Gateway also supports HTTPS for secure communication.

Outbound-Call Establishment

When the Supplementary Services Gateway receives an HTTPS POST trigger from the trigger application with CreateRequest in the body, it initiates an outbound call by sending a TMakePredictiveCall request to SIP Server. SIP Server establishes two call legs; one to Media Control Platform through the Resource Manager and one to the external party. The call legs are then bridged, which invokes a third-party call-control scenario.

Requests for Service

When the request reaches the Resource Manager it is treated like any other request for service within GVP. The Resource Manager determines the type of service and application profile that will be used to fulfill the request and sends it to the Media Control Platform, which then provides the media-centric services for the Supplementary Services Gateway specifically VoiceXML applications. See also, Call Initiation Through SIP Server.

Component Management

Like all other GVP components the Supplementary Services Gateway is managed (stopped, started, or restarted) and monitored by the Genesys Administrator web interface. It also receives configuration and provisioning information from the Configuration Server.

Requests and Responses

The Supplementary Services Gateway and the trigger application use the HTTP request/response standard used in server/client environments. The Supplementary Services Gateway sometimes acts as the server and sometimes as the client, depending on whether it is requesting or providing information. (See Supplementary Services Gateway Interfaces).

This section describes the following types of requests and the corresponding responses:

- Create Requests
- Query Status Requests
- Cancel Requests

Create Requests

The Supplementary Services Gateway receives HTTP triggers from the trigger application for single and bulk outbound-call requests. The call triggers include a Token which uniquely identifies the trigger application submitting the request. The Supplementary Services Gateway responds to these call triggers by generating a RequestID and managing the status of outbound-call requests.

Outbound Requests HTTP POST

TAs use the HTTP POST method to submit outbound-call requests to the Supplementary Services Gateway. HTTP POST is used to create, query, and cancel requests. The body of a single POST can contain a single CREATE, QUERY, or CANCEL request, multiple (bulk) CREATE, QUERY, or CANCEL requests, or any combination of all three requests. The Supplementary Services Gateway does not impose a limit on the size of an HTTP POST request.

Request Validation

The Supplementary Services Gateway validates each HTTP POST request sent from the TAs based on the following criteria:

- The body of each HTTP POST request must conform to the schema that is defined for the HTTP POST method.
- Each POST request must include the TenantName parameter in the query string of the POST Request URI for example, <host>:9800/SSG?TenantName=<Tenant_Name>
- The Content-Type for each request must be text/XML or application/XML.

If the POST request meets this criteria, it is validated and added to the Supplementary Services Gateways persistent storage (an external database).

Request Acceptance

After validation, the Supplementary Services Gateway parses the XML data in the POST request body and generates a unique identifier (RequestID) for each request. The RequestID is stored in the Supplementary Services Gateways database along with the request. The Supplementary Services Gateway sends the RequestID and Token back to the trigger application to indicate that the request has been accepted and simultaneously processes each request in storage. A detailed description of this process is described in Basic Outbound-Call Flow on page 464.

Each HTTP POST request must adhere to the XML schema. The CreateRequest part of the POST body must include certain mandatory attributes, such as: Token, IVRProfileName, Telnum, NotificationURL, MaxAttempts, and TimeToLive. For example:

```
<CreateRequest Token="Token" MaxAttempts="2" TimeToLive="123s"
IVRProfileName="Application" Telnum="9884719189"
NotificationURL="http://182.123.12.12/DIR/OutURL.xml"
Ani="12345"></CreateRequest>
```

For a detailed description of the attributes of the CreateRequest, see the Genesys Voice Platform 8.5 User's Guide.

Responses to Outbound Request HTTP 200 OK

In most cases, the Supplementary Services Gateway responds to requests from the trigger application with HTTP 200 OK responses. The response section of the 200 OK message can contain single or bulk requests, depending on whether the POST (to which it is responding) is a single or bulk request. The response contains the RequestID, the Token, and the request result (success or failure) which is in a format defined by the XML response schema. For more information about the XML response schema, see the Genesys Voice Platform 8.5 User's Guide.

Success and Failure Response Types

The Supplementary Services Gateway generates responses to indicate success or failure, based on the following methods of validation:

- If a request is successful, the response sent to the trigger application includes the RequestID and Token for the request, or for each request within the bulk request with a SUCCESS response type. The trigger application must include the RequestID in any further requests (such as, status querying or request cancellation).
- If a request fails during request validation or when it is being stored in the external database, a FAILURE response type is sent within the 200 OK response, along with the Token, a Reason Code, and Reason. See the following SSGResponse part of a 200 OK bulk response:

```
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<ResponseElement ResponseType="SUCCESS" Token="T1001"
RequestID="123435"/>
<ResponseElement ResponseType="SUCCESS" Token="T1002"
RequestID="123436"/>
<ResponseElement ResponseType="FAILURE" Token="T1003"
ReasonCode="120" Reason="DB Insertion failed"/>
</SSGResponse>
```

- If parsing or validation fails for a bulk POST request, the entire request fails and the Token is not passed back to the trigger application. However, if it is parsed and validated successfully, but later encounters an operational error, (for example, a request fails when it is added to the database) the bulk response contains a mix of SUCCESS and FAILURE response types and the Tokens for each request are passed back to the TA.
- If a POST request contains a mixture of CREATE, QUERY, and CANCEL requests, and the number of database records is approaching the maximum during processing, the Supplementary Services Gateway inserts them into the database until the maximum threshold is reached and then, executes the QUERY and CANCEL

requests. The Supplementary Services Gateway sends a FAILURE response for the remaining requests with a Reason Code, and Reason.

The Supplementary Services Gateway also generates HTTP 500 responses to indicate internal errors. For a detailed description of the attributes of the SSGResponse part of the response and a complete list of status codes in the response, see the Genesys Voice Platform 8.5 User's Guide.

Query Status Requests

The trigger application uses two methods to query the status of previously submitted requests that are stored in the Supplementary Services Gateway database: HTTP GET for a single query and HTTP POST for single or bulk queries. When the trigger application sends GET or POST query requests, they must contain the RequestID and TenantName parameters. The request is passed on in the HTTP GET query string, or in the body of the HTTP POST request. In the HTTP POST query request, the Content-Type must be text/xml or application/xml and must conform to the Supplementary Services Gateway XML schema.

Responses to Status Queries

The following responses are the same for both the GET and POST methods of querying requests:

- If the Supplementary Services Gateway finds the request (or each request if POST is used) in its database and obtains the status of the request, a 200 OK response, with a SUCCESS ResponseType is sent to the trigger application. The response also contains the RequestID, Token, and other attributes (see the example in this section).

The Content-Type in the 200 OK response is text/xml. If the Content-Type of the POST request is neither text/xml or application/xml, the Supplementary Services Gateway returns a 415 status code in the body of the response, indicating an invalid content type was used.

- If query request parsing fails, mandatory attributes are missing, or database operation fails, only the RequestID, ReasonCode, and Reason parameters are passed back to the TA.
    - If validation or parsing fails, the entire POST or GET request fails and the Supplementary Services Gateway generates a ReasonCode and Reason (or failure description).

- If validation and parsing succeeds, but there are other failures for example, a specific RequestID is not found in the database the RequestIDs are passed back in the 200 OK response.

The following is an example of the SSGResponse part of a 200 OK bulk query response:

```
<SSGResponse>

<ResponseElement ResponseType="SUCCESS"
Token="T1001"RequestID="123435"
TenantName="Environment" IVRProfileName="Application" Telnum="11011"
NotificationURL="http://182.24.129.82/Dir/Response.asp"
AttemptsMade=4
MaxAttempts=7 TimeToLive="12000s"
TTLRemaining="3477s"Status="Waiting to be processed"/>

<ResponseElement ResponseType="FAILURE" RequestID="1234"
ReasonCode="404" Reason="RequestID not found in the Database"/>

</SSGResponse>
```

For a complete list and description of the HTTP request and response attributes, see the Genesys Voice Platform 8.5 User's Guide.

Cancel Requests

Trigger applications use two methods to cancel pending outbound requests: HTTP DELETE to cancel a single request and HTTP POST to cancel a single or bulk request. When the trigger application sends DELETE or POST requests, they must contain the RequestID and TenantName parameters. The request is passed on in the HTTP DELETE query string, or in the body of the HTTP POST request. The Content-Type of the HTTP POST cancel request must be XML text or an XML application that conforms to the XML schema.

Responses to Cancel Requests

A request that is not in progress (the TMakePredictiveCall request has not been sent to SIP Server), can be cancelled and the record deleted from the Supplementary Services Gateway database. However, if the request is already in progress, the Supplementary Services Gateway does not attempt to delete the request from its database, but sends a FAILURE response type in the 200 OK response.

The responses to requests for the DELETE and POST methods of cancellation requests are the same as for query requests. See Responses to Status Queries.

The following is an example of the SSGResponse part of a 200 OK bulk cancellation response:

```
<SSGResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<ResponseElement ResponseType="SUCCESS" RequestID="1121245"/>
<ResponseElement ResponseType="FAILURE" RequestID="1000000"
ReasonCode="106"Reason="Invalid RequestID"/>
</SSGResponse>
```

Asynchronous Result Notifications

The Supplementary Services Gateway notifies the trigger application about the final outcome of the outbound request in the form of a NotificationURL, which is a mandatory parameter in the CreateRequest. The Supplementary Services Gateway performs an HTTP GET on the Notification URL and appends certain query-string parameters that indicate the status of the outbound request

Asynchronous Result Notification Success

When a call is made through a trunk group DN and a TreatmentApplied event is received from the SIP Server, the Supplementary Services Gateway deems the call successful, appends the query string, and sends the Notification URL to the trigger application.

Notification URL Query-String Parameters

Among the query-string parameters that are passed back to the trigger application in the Notification URL is the CallUUID, which is a unique ID that is generated by the SIP Server for the call. When the outbound call has been attempted multiple times, only the last calls UUID is sent in the Notification URL. In addition, the Status parameter contains the reasons (separated by a colon [:]) that each prior attempt failed for example,

```
AnsweringMachineDetected:RingNoAnswer:DestinationBusy
```

The following is an example of a Notification URL that is sent when the outbound call is successful:

```
//test.genesyslab.com/trigger/
result.asp?Token=T3001&RequestID=123345&TenantName=Environment&IVRProfileName
8GQE8C05B56SV2HRTTJ8M9RFR8000001&Result=SUCCESS&Status=DestinationBusy:RingNo
```

Asynchronous Result Notification Failed

Requests can fail at two stages of the process: before they are stored in persistent storage and after they have been processed. An outbound request fails after processing for various reasons: It exceeded the maximum attempts (MaxAttempts in the request), the time-to-live (TTL) expired, or a permanent error caused it to fail.

The query-string parameters that are passed back to the trigger application in the Notification URL are the same as for a success notification, and the Status parameter contains the reasons (separated by a colon [:]) that each prior attempt failed. For example,

```
ExternalError:RingNoAnswer:DestinationBusy:MaxAttempts Exceeded
```

Call Initiation Through SIP Server

The Supplementary Services Gateway receives HTTP requests from the trigger application along with parameters in the body of the POST requests. Acting as a T-Lib client, the Supplementary Services Gateway uses SIP Servers T-Lib interface to send a TMakePredictiveCall, with the parameters sent as extensions. SIP Server then establishes two call legs by sending one INVITE request to the Media Control Platform (through Resource Manager) and another to the external party.

SIP Server, acting as a T-Server, establishes two call-legs one to the Media Control Platform through the Resource Manager and one to the external party. The call legs are then bridged, invoking a third-party call. The call leg that is established to the Media Control Platform, contains parameters that are passed on in the INVITE messages to Resource Manager as Request URI parameters or special headers. The Resource Manager uses these parameters to determine if a specific IVR Profile is required. After the call legs are established, the VoiceXML application associated with the IVR profile is played for the external party.

Connection to SIP Server in HA Mode

To Supplementary Services Gateway establishes a connection with SIP Server in HA mode by obtaining the configuration details for both the primary and secondary SIP Server through CCILib.

After the connection to the primary SIP Server is configured in the Supplementary Services Gateway Application (in the Connections tab), it uses the port ID in the primary SIP Server to establish a connection with the secondary SIP Server. It attempts to establish a connection with the primary SIP Server first. If there is no response, it tries with secondary SIP Server by using a simple round robin mechanism.

Secure Connections Through TLS

SIP Server can connect to the Supplementary Services Gateway on any of its secure or unsecured ports. Secure ports are configured by creating security certificates, which enable the Supplementary Services Gateway to interact with SIP Server by using TLS.

For a procedure that describes how to create security certificates, see "Chapter 3: Configuring Common Features" in the Genesys Voice Platform 8.5 User's Guide.

DNs for Resource Manager and External Party

The Resource Manager trunk-group DNs are registered on the Supplementary Services Gateway in the Tenant1 section and contains the following parameters, including the mandatory TGDN parameter:

```
[Tenant1]
TGDN=<tg-dn>
RPDN=<rpdn>
AccessGroup=<grp-name>
DialPrefix=<DialPrefix-name>
```

(The value that is configured in the TGDN parameter is used as the tenant name.)

The external-party Trunk DNs are configured on the SIP Server. SIP Server selects the external-party DN, based on the mandatory telnum parameter, which is one of the parameters in the body of the HTTP request.

Call-Progress Detection

The Supplementary Services Gateway and SIP Server support Call-Progress Detection (CPD) on the Media Gateway or on the Media Server module of the Media Control Platform. CPD is optional, however if it is configured on both the Media Gateway and Media Server, the Media Gateway takes precedence as the CPD provider. SIP Server receives the CPD result from the CPD provider and passes it back to the Supplementary Services Gateway through a T-Event.

CPD Control Parameters

Trigger applications send CPD control parameters to the Supplementary Services Gateway in the CreateRequest part of the HTTP POST requests. All of the CPD parameters are optional. The value of the record parameter specifies if the CPD part of the call is recorded `true` (or `1`) if the CPD is to be recorded and `false` (or `0`) if it is not. The value of the preconnect parameter is mapped to the cpd-on-connect extension in the TMakePredictiveCall request and specifies when to start CPD `true` if CPD is to be started when the first packet is received, and `false` if CPD is started when the call is connected. If there are no CPD parameters in the CreateRequest part, SIP Server relies on its own CPD configuration.

For a complete list and description of CPD parameters, see the Genesys Voice Platform 8.5 User's Guide.

The following is an example of the CreateRequest part of a POST request with CPD parameters:

```
<CreateRequest Token="Token" MaxAttempts="2" TimeToLive="123s"
IVRProfileName="Application" Telnum="9884719189"
NotificationURL="http://182.123.12.12/DIR/OutURL.xml" Ani="12345">

<cpd record="false"
postconnecttimeout="6000ms"
rnatimeout="60s"
preconnect="true"
detect="all"/>

</CreateRequest>
```

For information about how the Media Control Platform (Media Server) supports CPD, see How the Media Control Platform Works.

Port-Availability Notifications

The Supplementary Services Gateway relies on T-lib event notifications from SIP Server to receive notification of available ports. The Resource Manager and Media Control Platform have a finite number of ports available for outbound calls; therefore, the Supplementary Services Gateway must be aware of the maximum number of ports that are available for a tenant and the number of ports that are available at any given time.

Subscription Request

The Supplementary Services Gateway sends the subscription request to SIP Server as a RequestRegisterAddress extension in the resource DN (configured in the Supplementary Services Gateway). SIP Server sends notifications on behalf of the DN using the EventResourceInfo event with extensions that include the total number of ports that are allocated and the total number of ports that are available for outbound calls. If the Resource Manager is unavailable, SIP Server returns zeros (0) for total-ports and available-ports parameters.

The Resource Manager provides port-availability notifications that include data that is specified for each tenant. When the Supplementary Services Gateway receives these notifications for a tenant trunk group, it uses these new values to manage the tenant campaigns.

Persistent Storage

The Supplementary Services Gateway uses an SQLite external database for its persistent storage, supported on Windows and Linux. It stores requests for outbound-call initiation persistently for the following reasons:

- Although a limited number of Resource Manager and Media Control Platform ports are available for outbound calls, outbound requests are not rejected, because they can be stored and executed as ports become available.
- Previous requests can be retrieved from the database across Supplementary Services Gateway restarts. Incomplete calls are reinstated as new calls and re-attempted.
- The progress and status of each request can be maintained in the database.

There is a threshold for the maximum number of records allowed in the database. When the maximum number of records is reached, further requests from trigger applications are rejected with the appropriate error code in the response. See also, Responses to Outbound Request—HTTP 200 OK.

Processing Requests

The Supplementary Services Gateway processes HTTP requests in batches based on information received in EventResourceInfo messages sent every five (5) seconds from SIP Server. The EventResourceInfo messages contain the total number of GVP ports, the number of available GVP ports, and the TenantName.

Fetching from Database to InMemQ

Requests are fetched from the database and placed in the Supplementary Services Gateway in-memory queue (InMemQ). The number of fetched requests can be configured with the Request Batch Size parameter using one the following values:

- `TotalPorts` Use the total ports returned in the EventResourceInfo message from SIP Server. This is the default value.
- `AvailablePorts` Use the available ports returned in the EventResourceInfo message from SIP Server.
- `NNN` A specified number, for example, 610.

When the InMemQ falls below a certain percentage of the Request Batch Size parameter the next database fetch cycle is triggered and the state of each fetched record is marked INITIATED in the database. The default percentage is 25% and can be configured using the Queue Low Watermark parameter.

Allocation of InMemQ for IVR Profiles

A separate InMemQ exists for each IVR Profiles, the contents of which is processed using a round-robin algorithm. If there are requests in the database for more than one IVR Profile, a portion of the batch size is allocated to each application. The Application Slot Calculation parameter controls how the batch size is divided using one of two values:

- **PROPORTIONATE** The batch size is divided in the same proportion as the number of requests pending for the IVR Profiles. This is the default value.
- **EQUAL** The batch size is divided equally among the IVR Profiles.

Prioritizing Requests

When requests are fetched from the database for a specific IVR Profiles, they are prioritized in the following ways:

- Requests with earlier `NextRetryTime` values are picked up ahead of those with later NextRetryTime values.
- If requests have the same `NextRetryTime` values, the requests with least remaining time-to-live (TTL) have priority over those with higher remaining TTL.
- If requests have the same `NextRetryTime` values and same remaining TTL, the requests with the higher number of attempts have priority over those with fewer number of attempts.

Each IVR Profile can have a combination of new requests in the database and requests that were already processed (the call failed and they will be re-attempted). The value of `EqualPriorityToNewAndOld` determines how the Supplementary Services Gateway picks up requests, for example, if the value is:

- **`False`** The requests for each IVR Profile are picked up based on the NextRetryTime, remaining TTL, and number of attempts. This is the default value.
- **`True`** For each IVR Profile in a fetch cycle an equal amount of new and previously-attempted requests are picked up.

Processing Requests from `InMemQ`

The requests in `InMemQ` are passed to the SIP Server to make outbound calls. The Supplementary Services Gateway checks the `AvailablePorts` parameter in the `EventResourceInfo` message to determine the number of GVP ports available for outbound calls. The number of calls the Supplementary Services Gateway is able to pass to SIP Server is determined by the value of the `PortLoadFactor` parameter, which has an `nnn` value (expressed as a percentage of the available ports). The default value is `100`.

After the number of outbound calls is determined, the Supplementary Services Gateway throttles them rather than send them to SIP Server all at once (which could result in high Calls Available Per Second [CAPS] and overload the components). The throttling functionality is controlled by the value `X` of the `pacing.caps.CallRequestsPerSecond` parameter that the Supplementary Services Gateway sends to SIP Server. The Supplementary Services Gateway only attempts `X` calls per second to SIP Server. It continues to make `X` calls every second until the number of requests that are configured in the `PortLoadFactor` parameter is exhausted.

Error Handling

Failed requests are categorized as permanent or temporary failures and handled in the following ways:

- Permanent failures The record is marked as PROCESSED in the database, the result is marked as FAILURE, and the deleteflag is set.
- Temporary failure, where either the number of attempts exceeds the maximum number of attempts (`#Attempts>MaxAttempts`) or the TTL has expired The record is processed the same as for a permanent failure.
- Temporary failure that have attempts left and TTL remaining The `NextRetryTime` is calculated:

- For internal failures (within GVP), the number of attempts is not updated in the database. Requests are pushed either to the back or in front of the `InMemory` Queue. The `NextRetryTime` value is calculated based on the `NextRetryInterval` parameter value. The default value is `10` seconds.
    - For external failures, the number of attempts is increased in the following ways:
    - For Busy or SIT Tones (other than those that are treated as permanent failures) The `NextRetryTime` parameter value is paced closer together and spread out later.
    - For all other external failures (such as `Answering Machine` and `Ring No Answer`) The `NextRetryTime` parameter value is paced equally. (The remaining TTL is divided equally among the remaining attempts.)

After the `NextRetryTime` parameter value is calculated, the request is either left in the `InMemQ` (if the current time `>= NextRetryTime`) or put back into the database and processed later.

Calls that succeed are marked as PROCESSED in database, the result is marked as SUCCESS, and the deleteflag is set.

Database Cleanup

The persistent storage database is monitored and cleaned up at regular intervals (configured with the `Clean Interval` parameter; default value = 180 [seconds]). Cleanup occurs in three stages:

- The database is trawled, and all records that have `deleteflag` set are collected.
- A SUCCESS or FAILURE Notification URL is sent to the trigger application for each record. If an error occurs during invocation of the Notification URL, an SNMP trap is generated.
- The selected records are deleted from the database.

Support for Nuance SessionXML

GVP 8.1.7 and above support SessionXML for:

- Nuance NR10 over MRCP v2 (recognition)
- Vocalizer 5.7 over MRCPv2 (synthesis)

You can use a single SessionXML file for both recognition and synthesis. Your Nuance documentation explains in detail how to create a SessionXML file.

MRCPv2 Support

When receiving an MRCP request, the MRCPv2 client searches the session configuration parameter object for the Nuance Speech Synthesizer (NSS) `session.xml` file URL, which is specified by including `gvp.config.vrm.nsssessionxml` in the RURI of the call.

See Resource Manager Support.

- If the URL is not specified, the MRCPv2 client retains the current behavior.
- If the URL is specified, the MRCPv2 client sends the SessionXML data to the NSS. This action is controlled by the engine-specific provision parameter `vrm.client.SendSessionXML`. This parameter enables/disables this functionality:
    - `true` enables the SessionXML file contents to be sent to NSS server.
    - `false` (default) disables transmission.
    - MRCPv2 supports only the local SessionXML file that is specified in `vrm.nsssessionxml`. It supports only `file://` protocol and absolute path.

Tip

The MRCPv2 client embeds the SessionXML file the contents as-is into the body of the SDP of the INVITE message, when it requests a Speech Resource.

The MRCPv2 client does not validate the contents of the specified SessionXML file. If that file is empty or cannot be read, MRCPv2 returns an error.

Resource Manager Support

On a per application and per tenant basis, the RM allows you to configure the NSS SessionXML URL by specifying it as the configuration parameter `gvp.policy.speech-resources.nsssessionxml`. RM looks first for this Type-III policy parameter in the IVR Profile and if the search is fruitless, then searches the tenant hierarchy from the bottom up. When found, the RM passes the URL to the MCP handling the request as the SIP Request URI parameter `gvp.config.vrm.nsssessionxml`.

Tip

Configure this parameter exactly as you would any other IVR Profile/tenant-level configuration parameter.

How Logging and Reporting Works

GVP Reporting refers to the GVP logging and reporting feature, which provides these features and services:

- Accumulating key measurements and data that describes the calls being processed by the deployment
- Infrastructure capable of reliable delivery of data to a relational back end
- Near real-time reporting about operational aspects of the deployment
- Historical reporting about VoiceXML and CCXML application usage

For an overview of the GVP Reporting architecture, see GVP Reporting Architecture. Read here about:

- GVP Logging
  - Logs
  - Metrics
  - Log Sinks
- CDR Reporting
- OR Service
- VAR Per-Call IVR Actions Reporting Service
- SQA Service
- Reporting Client
- Reporting Server
- Reporting Web Services

GVP Logging

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

- *logs* are events at the level of the component, or GVP Application object.

- *metrics* are events at the level of the VoiceXML or CCXML application.

Logs

Logs include three important elements by which they can be filtered: severity, module ID, and specifier.

Severity

Like most other Genesys components, GVP components can raise events at various levels, however, they can also log them further by the following levels of severity (in order of descending 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 that 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 the Module IDs and specifiers that are used in GVP 8.5, see the Genesys Voice Platform 8.5 User's Guide.

Additional Log Data

GVP 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 (DBID of the GVP application that raised the log), the GVP session ID, and the UTC timestamp. Metrics can therefore be mapped to CDRs, which provide more information, such as the call start time, call end time, IVR Application ID (DBID of the IVR Profile), Tenant ID, and local and remote SIP URIs.

The GVP session ID is also unique to the deployment, where the environment may include deployments at multiple sites or geographic locations.

For more information about GVP IDs, see the section about GVP identifiers in the Genesys Voice Platform 8.5 User's Guide.

Log Delivery

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

By default, log files are stored in the following location:

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

The Management Framework Adapter sink passes along GVP log messages to its own logging system. In general, GVP logging is mapped to the Management Framework logging levels in the following way:

- Metrics are mapped to the interaction level and have IDs between `50000-55000`.
- Critical, Error, and Warning logs are mapped to the standard level.
- Notice and Info logs are mapped to the trace level.
- Debug logs are mapped to the debug level and usually have an ID of `20000`.

For a list of Management Framework IDs for GVP log messages, see the LMS file for each GVP component. The LMS file contains a mapping of GVP log messages to Management Framework IDs. The LMS file is located in the `<Install_Dir>\config` directory for each GVP component.

For more information about the GVP log messages, see the appendix *"Module and Specifier IDs"* in the Genesys Voice Platform 8.5 User's Guide.

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 8.5, see the Genesys Voice Platform 8.5 Metrics Reference Guide.

- 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 amount of time to fetch a VoiceXML page is measured and logged.

VAR Metrics

Voice Activated Response (VAR) metrics are events that the Media Control Platform generates 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 the Media Control Platform reference information appendix in the Genesys Voice Platform 8.5 User's Guide.

For more information about using `<log>` tags in VoiceXML applications, see the Genesys Voice Platform 8.1 Genesys VoiceXML 2.1 Reference Help.

Metrics Delivery

Metrics are delivered to the log sinks for the component (see Log Sinks). Upstream metrics, which are also referred to as call events, are metrics that are configured to be sent to the Data Collection Sink (DATAC), and then to the Reporting Server for storage and reporting purposes. The DATAC also computes service-quality measurements based on the logs and metrics that are forwarded to it. The Media Control Platform and 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, that receive a real-time stream of logs or metrics, as defined by filters that you can configure (in the `ems.logconfig.<Sink Name>` and `ems.metricsconfig.<Sink Name>` parameters).

The log sinks enable GVP 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-based 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. (This sink is not applicable for the Resource Manager or Fetching Module.) The `ems.dc.default.metricsfilter` configurable option on the Media Control Platform and Call Control Platform 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 management information bases (MIB).

The log sinks are dynamic link libraries (DLL) that are loaded dynamically at runtime. By default, the following log sinks are attached to each component:

- Resource Manager and Fetching Module MFSINK and TRAPSINK.
- Media Control Platform and Call Control Platform DATAC, MFSINK, and TRAPSINK.

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 can generate log events, therefore, they have one or more log sinks attached to them. For example, starting in GVP 8.1, DATAC has an MFSINK, which enables DATAC logs to be delivered to Message Server or to file.

CDR Reporting

Call-detail records are records that describe key attributes of a call session that the deployment is processing, or has processed.

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. This is the ID that the Resource Manager assigns to all calls that come into GVP. For more information about GVP session-IDs, see the Genesys Voice Platform 8.5 User's Guide.

The intervals at which the Reporting Client submits CDRs to the Reporting Server depends on the configuration. For configuration considerations, see the description of the ems.rc.cdr.batch_size configuration parameter in the section about configuring GVP Reporting in the Genesys Voice Platform 8.5 User's Guide.

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 times.
- IDs for the VoiceXML or CCXML application; Media Control Platform or Call Control Platform session; Resource Manager session; and overall Genesys session (UUID).
- IDs for the tenant for which the call is associated. The CDR report returns CDRs for all tenants that are in the tenant hierarchy.

- Call type Available types are:
    - `Inbound (1)` For the Resource Manager and Media Control Platform.
    - `Outbound (2)` For the Resource Manager and Media Control Platform.
    - `Bridged (3)` For the Media Control Platform.
    - `Unknown (4)` For the Resource Manager.
    - `New Call (5)` For the Call Control Platform.
    - `Create-CCXML (6)` For the Call Control Platform.
    - `External (7)` For the 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.

Attributes Specific to the Media Control Platform

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).

The Media Control Platform adds the following attributes to the CDR:

To capture usage information:

- **ASR** If ASR is used at any point during the call.
- **TTS** If TTS is used at any point during the call.
- **VOICEXML** If VoiceXML was used during the call.
- **NATIVECPA** If native media server CPD/CPA was used during the call.
- **GATEWAYCPA** If gateway-based CPD/CPA was used during the call.

To capture information about recording execution:

- **LOCALREC** If a local recording was executed during the call.
- **MSRREC** If a media stream replication recording was executed during the call.

To capture information about conference, bridging, and connection establishment:

- **CONF** If conferencing was established during the call.
- **BRIDGING** If bridging was established during the call.

- **VIDEO** If a video connection was established during the call.
- **CODEC** If any transcoding was used for this call (it does not matter which leg).

To capture MSML requests information:

- **MSPLAY** If MSML <play> was requested.
- **MSCOLLECT** If MSML <collect> or <dtmf> was requested.

CDR Attributes Specific to the Call Control Platform

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: <call-params> records the relevant parameters (for example, the UUID of the connection).

The reason that the CCXML session ended:

- **EXIT** The <exit> tag was executed.
- **KILL** The session was terminated by a ccxml.kill.unconditional or 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 ended 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 it enables the Call Control Platform component to accumulate statistics about CCXML session peaks. The statistics are submitted to the Reporting Server, through the Reporting Client:

Call Arrivals

- Counts are derived from the CDRs as they are submitted or updated.
- The Resource Manager submits arrivals for the CTI and PSTN Connectors.

Call Peaks

- Statistics are derived from counts of the maximum number of concurrent calls that are observed within a given 5-minute time period.
    - The Resource Manager submits peaks for the deployment as a whole, for the CTI and PSTN Connectors individually, and for each IVR Profile that is processed on a per-tenant basis.
    - The Media Control Platform submits peaks for itself only.

The Reporting Client submits OR data to the Reporting Server at the interval that is configured in the ems.ors.reportinginterval parameter. The default is once per minute.

VAR Per-Call IVR Actions Reporting Service

The VAR Per-Call IVR Actions Reporting Service is a Reporting web service endpoint, which is available at the following URL: `/ems-rs/HIST/CDRs/MCP/VAR/actions`.

It lists the IVR actions that are occur within the lifetime of an active Media Control Platform session. Per-call IVR actions are logged with the following labels: `com.genesyslab.var.ActionStart`, `com.genesyslab.var.ActionEnd`, and `com.genesyslab.var.ActionNotes`.

This service supports `session-id` parameter, which works in conjunction with the comp-id parameter. The `session-id` parameter selects a single call that matches a session ID that is local to a given call processing server. Session IDs are not necessarily globally unique, therefore, this parameter must be accompanied by the `comp-id` parameter.

The VAR Per-Call IVR Actions Reporting also supports the `gvp-guid INSERT_TEXT` parameter. The Resource Manager manages the GUID and passes it to all call processing servers that provide service for a specific GVP session. When this parameter is specified, the report returns information for all Media Control Platform sessions that are associated with a specified GVP session (if it has been served by multiple Media Control Platform sessions).

All calls associated with a specific Genesys Management Framework session can also be selected, by using the genesys-uuid as the identifier. No two session IDs, GVP GUIDs, or Genesys UUIDs can be specified at the same time, otherwise an HTTP 400 error code is returned.

For a complete description of the Per-Call IVR Actions Report, see "Chapter 19: Historical Reports" in the Genesys Voice Platform 8.5 User's Guide'.

SQA Service

The Service Quality Analysis (SQA) service on the call processing servers, provides statistics on the service quality of GVP deployments and sends this data to the Reporting Server through the Reporting Client. This differs from the analysis of operational logging and reporting, which typically focus on the availability and performance of servers and system components. Events that affect service quality might not show up in any operational logs.

Service-quality measurements account for all calls to the system. The platform itself measures the calls being handled, rather than using a sampling methodology where periodic test calls are made to the system. SQA gathers information from the platform and the applications running on the platform and prepares reports on quality at regular interval.

> Tip
>
> The ems.dc.enableSQA parameter can be set to false if service quality, latency, and call-failure tracking information is not required. When this parameter is disabled, the Reporting Client does not send this data to the Reporting Server.

SQA and NGI Compatibility

SQA is currently designed to work properly only with MCPs using NGI.

Some SQA metrics may show up for non-NGI configurations, because some configurations of GVP can be deployed with other interpreters (such as the Legacy GVP Interpreter). But beginning with release 8.1.5, MCP does not support those interpreters.

SQA Metrics

The Reporting Server Client obtains data about service quality from the call processing components and forwards it to the Reporting Server. The following types of service-quality data are accessible through the various SQ reporting UIs in the Monitoring > Voice Platform pane in the Genesys Administrator:

Service-Quality Calculations

The period of time for which service quality is calculated can be configured in the Media Control Platform Application. The ems.dc.serviceQualityPeriod parameter is configured with values that divide evenly into 1 hour (the default value is 15 minutes).

The following functions occur within each interval:

The Reporting Client forwards accumulated metrics to the Reporting Server, such as:

- The number of completed calls on each Media Control Platform. A call is considered complete if either an incall_end or outcall_end event is logged or the call completes abnormally (see Abnormally Terminated Calls).
- The number of failed calls on each Media Control Platform.

SQ metrics are calculated, such as:

- The percentage of successful calls on each Media platform.
- The number of completed and failed calls, as well as the percentage of successful calls for the cluster.
- Aggregated hourly, daily, weekly, and monthly summaries of the number of completed calls, number of failed calls, and percentage of successful calls. These summaries are accumulated for each Media Control Platform and for the entire cluster.
- Deletion of statistics for the basic service-quality period (15 minutes) after a configurable period of time (2 days). Hourly, daily, weekly, and monthly statistics are also deleted after a configurable period of time.
- System dialog messages that are generated if SQ data retention periods are configured in such a way that prevents statistics aggregation from being done properly.
- Logged events that contain the service-quality percentage for a specific application in the cluster. These events can be used to trigger alarms when service quality falls

below the configured threshold. Logged events are not generated when the number of completed calls in the service-quality period is less than the configured threshold.

- The percentage of successful calls for each application that has been executed for each Media Control Platform.
- The percentage of successful calls for each application that is executed in the deployment. This percentage is derived from aggregated results that are reported by the individual Media Control Platforms.

Service-Quality Failures

The SQA service provides metrics for the following failure types:

Failed Logging Sessions

When the VoiceXML interpreters on the Media Control Platform generate a `<log>` with a `com.genesyslab.quality.failure` or `com.genesyslab.quality.failure` label, the logging session is considered to have failed. SQ Failure types include data relating to latency, application, and audio gap failures (see Call Failures).

Latency Intervals

SQA uses the intervals-between-events data that is derived from component latency reporting and compares it with configured thresholds. Latency periods are measured in milliseconds(ms).

Abnormally Terminated Calls

When calls are abnormally terminated, they are marked with the Abnormal Termination failure type. The following call events are some examples of calls that are considered abnormally terminated:

- An `incall_initiated` event is logged, but no associated incall_begin or `incall_rejected` event is logged.
- An `incall_begin` event is logged, but a corresponding `incall_end` is not.
- The `incall_end` event is logged with the `syserr` reason code.
- The Resource Manager logs a `sip_session_timeout` event for the call.
- A `bridge_begin` event is logged, but no corresponding `bridge_end` event.

Call Failures

The SQA service gathers and stores call-failure information for each session that ends in each service-quality period. Call Failure Records are maintained in the database for a configured period of time (by default, `1` year).

The default value for the Call Failure Records can be overridden in the `gvp.rs.db.retention.sq.failures` configuration option. The SQA service stores the following information for each call failure:

- Session ID
- Session end time
- Application (associated with the session)
- Failure type
- Time of failure (within ms)
- Cause-of-failure description (value string)

Reporting data can be generated for various types of call failures and the SQ Failure Details report can be filtered to provide details about each of the specific call-failure types, including short strings for System Error call failures, which describe the cause-of-failure.

For more information about the configuration options that relate to all aspects of SQA, see the Genesys Voice Platform 8.5 User's Guide.

Reporting Client

The Reporting Client on each component provides reliable delivery of DATAC logs and metrics, CDRs, service-quality metrics, 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 is submitted to the Reporting Client is eventually sent to the Reporting Server, even if the Reporting Server is unavailable for an extended period of time due to an outage.

> **Tip**
>
> Data that is stored in memory is lost if the call-processing component shuts down unexpectedly. Data is persisted to disk only if the Reporting Client cannot successfully deliver the data to an available Reporting Server.

For improved performance, the Reporting Client can be configured to send CDRs and metrics in batches. However, this can result in slight delays in data delivery. For more information, see the description of the `ems.rc.cdr.batch_size` parameter in the Genesys Voice Platform 8.5 User's Guide.

Support for Reporting Server in TLS mode

The Reporting Client can connect to a Reporting Server in TLS mode. Whether or not it uses an encrypted connection, depends on how the activemq.connectionMode option in the messaging section of the Reporting Server (to which it is connected) is configured.

The Reporting Client s rc.certificate configuration option in the ems section contains the file name of the TLS certificate in Privacy Enhanced Mail (PEM) format. The certificate is required to connect to the Reporting Server (ActiveMQ) over TLS.

Reporting Server

As shown in the figure GVP Reporting Architecture, the services that the Reporting Server provides include the following:

- **Storage services** Logs and metrics that the Reporting Client (on the components) delivers 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 are displayed on the Monitoring tab in Genesys Administrator. For more information, see Reporting Web Services.
- **Service Quality (SQ) Alarm Generator** Alarms are generated through Reporting Web Services (in Genesys Administrator) when service quality falls below configured thresholds.
- **VAR Stats Generator** The Reporting Server computes VAR statistics, based on the VAR-specific metrics that it receives (see VAR Metrics).
- **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, SQ, 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 upon request. For example, you can request a monthly report for January before January has ended.) 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 per day, at a configurable time. The data-retention policies are also configurable. For more information, see the section about configuring database retention policies in the Genesys Voice Platform 8.5 User's Guide.
- **Database Partitioning** The Reporting Server supports partitioning for Oracle 10g or 11g Enterprise Edition, and Microsoft SQL Server 2008 Enterprise Edition, and provides compatible schemas. Partitioning is automatically enabled during installation if either of these database editions is selected.

Tip

Genesys recommends that you not change the partitioning mode of operation or the number of partitions (even after the Reporting Server is started) because of issues that might arise if the database schema or stored data is changed.

- Queries the SNMP MIBs from the Supplementary Services Gateway components and provides summarized data to the Reporting Web Services.

SNMP Query and Trap Generation

To manage SNMP query and trap generation for multiple Reporting Server instances in your environment, you can configure Reporting Server connections to the SNMP Master Agent. After the connections are established, the Reporting Server queries and generates traps in the following way:

- The Reporting Server finds the first Management Framework connection to an SNMP Master Agent and attempts to use the agent on this connection.
- If the agent is not reachable, the Reporting Server attempts to connect to the agent at regular intervals, which is configured by using the `connection_delay_sec` option in the `agentx` section of the Reporting Server Application. The default value is 60 seconds. By using the `max_connection_attempt` configuration option in the `agentx` section, the number of times the connection is re-attempted can also be configured.

For a complete list of Reporting traps, see the GVP 8.5 SNMP MIB Reference.

Reporting Web Services

Reporting Web Services can be deployed over HTTP or HTTPS and is deployed by default, at the following URL:

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

In multi-tenant environments, Reporting Web Services provides the `http://<Reporting Server host name>8080/ems-rs/tenants` URL, which returns a complete list of tenants, including their names and DBIDs, that are provisioned for the environment.

When SQA is enabled in the deployment, these reports are accessible:

Report	Use this URL to access
SQ Summary	`http://<Reporting Server host name>8080/ems-rs/sqa/servicequality`
Failure Details	`http://<Reporting Server host name>8080/ems-rs/sqa/failures`
Latency	`http://<Reporting Server host name>8080/ems-rs/sqa/latency/details`
Latency Histogram	`http://<Reporting Server host name>8080/ems-rs/sqa/latency/histogram`

The reporting services return results (reports) as XML documents that conform to available Regular Language for XML Next Generation (RelaxNG) schemas, therefore, the GVP Reporting data is available to third-party reporting products, and on the Monitoring tab in the Genesys Administrator GUI. User interfaces for Service Quality Advisor (SQA) Reporting, VAR Reporting, CDR Reporting, and Operational Reporting can be used to view and filter reports and statistical data.

Tip

Browse to: `http://<Reporting Server host name>:8080/ems-rs/components` to test Reporting Server.

For detailed information about the XML schemas for GVP Reporting Web Services, contact Genesys Customer Care.

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 the Genesys Voice Platform 8.5 User's Guide.

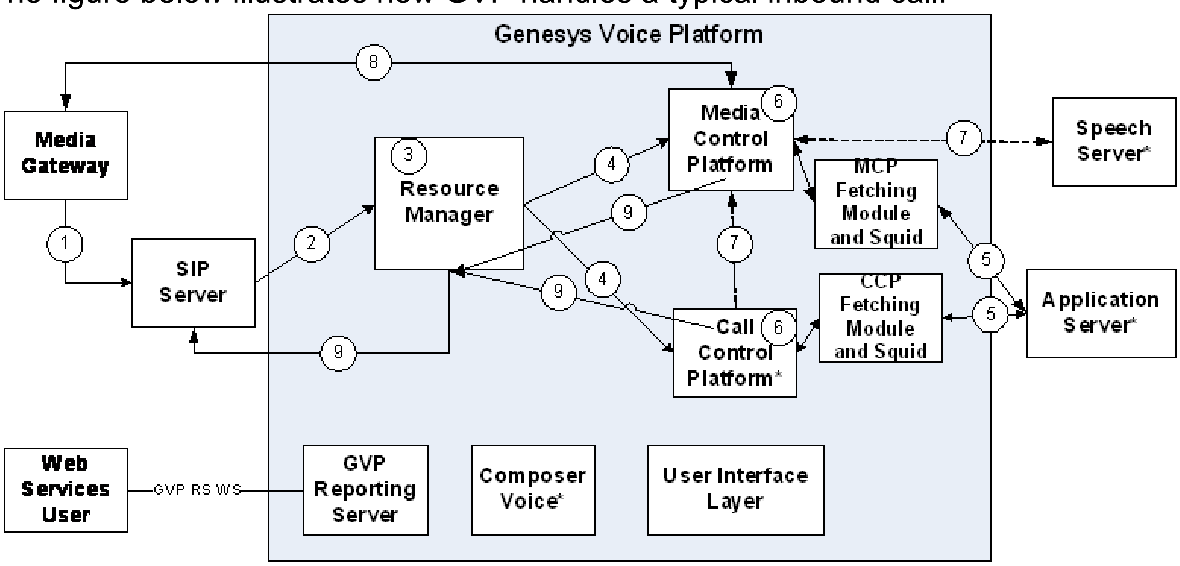
GVP Call Flows

This topic describes some sample basic Genesys Voice Platform (GVP) call flows.

- Basic Inbound-Call Flow
- Basic Outbound-Call Flow
- Basic CTI Call Flow (Inbound)
- Basic CTI Connector/ICM Call Flows (Inbound)
    - Cisco ICM Type 8 Deployment Call Flow
    - SCI Transfer Call Flow
    - Inbound Call Using CRI
- Basic PSTN Call Flow (Inbound)
- Basic PSTN Call Flows (Outbound)
    - Initiated by the Supplementary Services Gateway
    - Resulting From an Inbound Call Transfer

Basic Inbound-Call Flow

The figure below illustrates how GVP handles a typical inbound call:



Note: Additional connections not shown here include:
- TCP connections between GVP Reporting Server and RM, MCP, and CCP
- Connections between Management Layer and all other Genesys components

*Optional in VPS 8.0

[+] Basic Inbound-Call Flow Description

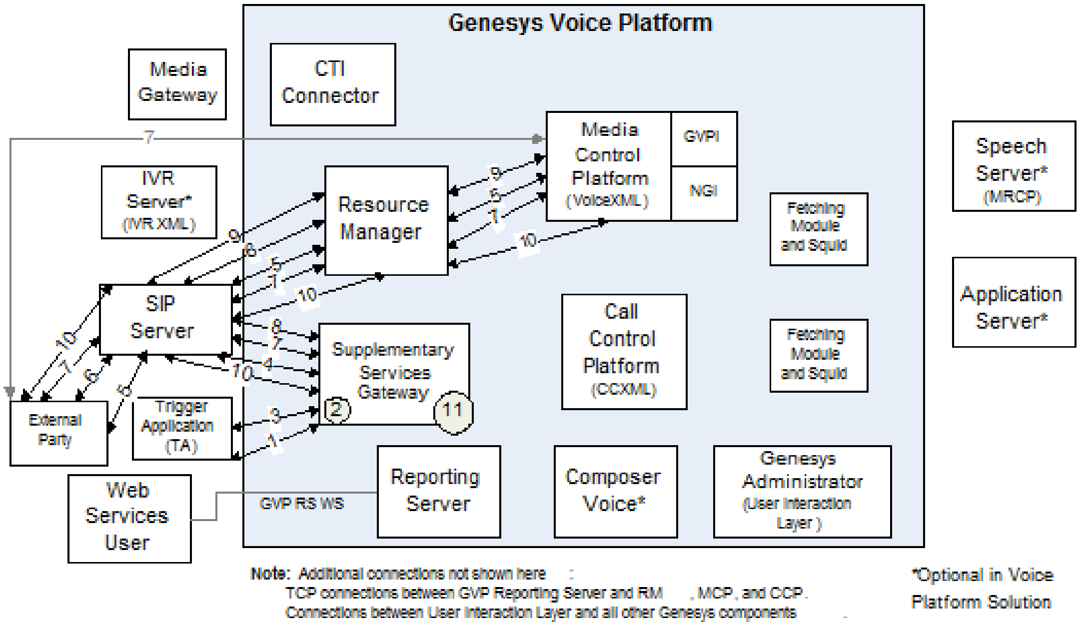
1. A call comes in to the Session Initiation Protocol (SIP) Server from an external source through a third-party media gateway.
2. The SIP Server passes the call to the VP Resource Manager (SIP INVITE).

3. The Resource Manager determines what to do with the call. If the Resource Manager accepts the call, it matches the call to an Interactive Voice Response (IVR) Profile and selects a resource. For more information about how the Resource Manager selects services and resources, and how it enforces policies, see How the How the Resource Manager Works.
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.
5. The Fetching Module for that Media Control Platform or Call Control Platform resource fetches the required Voice Extensible Markup Language (VoiceXML) or Call Control XML (CCXML) page from the application server (file, Hypertext Transfer Protocol (HTTP), or Secure HTTP (HTTPS) request).
6. The VoiceXML Interpreter (Next Generation Interpreter [NGI] or GVP Interpreter [GVPi]) on the Media Control Platform or CCXML Interpreter (CCXMLI) on the Call Control Platform interprets the page and executes the application (VoiceXML or CCXML).
7. Depending on the application, the Media Control Platform or Call Control Platform requests (through the Resource Manager) and uses 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 by using Media Resource Control Protocol (MRCPv1 or MRCPv2).
    - If the Call Control Platform requires conference or audio play/record services, it obtains them from a Media Control Platform resource.

   The Media Control Platform or Call Control Platform sends all requests for services from other GVP components through the Resource Manager (SIP or Network Announcement [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).

Basic Outbound-Call Flow

The figure below illustrates how GVP handles a typical outbound call (without CPD).



[+] Basic Outbound-Call Flow Description

Call Triggers

1. Trigger applications send HTTP POST or GET requests to the Supplementary Services Gateway (SSG) to request outbound-call initiation. The request includes a Token that uniquely identifies the request.

Validation

2. The Supplementary Services Gateway performs validation on each incoming request:
   - It checks the HTTP POST query string for the `TenantName`, for example, `<host>:9800/SSG?TenantName=<Name of the Tenant>` and rejects the request if the `TenantName` is not present.
   - It checks to ensure that all mandatory parameters are included in the `CreateRequest` part (`Token`, `IVRProfileName`, `Telnum`, and `NotificationURL`) and that it adheres to the defined XML schema.

- After validation, a `RequestID` is created for each `CreateRequest` and sent back to the trigger application in the HTTP response.

Success or Failure Response

3. The Supplementary Services Gateway sends a `200 OK` single or bulk response to the trigger application (depending on the whether the POST is a single or bulk request).
    - If request validation is successful, the Supplementary Services Gateway generates a unique (internal) ID or RequestID for each CreateRequest and inserts the request into the database. The `RequestID` and `Token` are passed back to the trigger application in the `200 OK` response with a SUCCESS ResponseType.
    - If validation fails when request is being parsed or when it is inserted into the database, the Supplementary Services Gateway inserts a FAILURE ResponseType in the `200 OK` response.
    - If parsing or validation of bulk POST requests fails, the entire request fails and the `200 OK` response contains a ReasonCode and Reason (or failure description).
    - If parsing or validation of bulk POST requests succeed, but an operational error occurs for example, the insertion of a specific record into the database fails the `200 OK` response contains a combination of SUCCESS and FAILURE ResponseTypes and the Tokens for each request.

Request to SIP Server

4. After the Supplementary Services Gateway accepts the request, it parses the XML data in the body of the POST request, and extracts the requests and its parameters.
    - It invokes a `TMakePredictiveCall` T-Lib request to initiate the outbound call through SIP Server.
5. SIP Server establishes two call legs; one to the Media Control Platform through the Resource Manager and one to the external party.
    - It sends a SIP INVITE request to the Media Control Platform through the Resource Manager and a SIP INVITE request to the external party.
6. Both the Resource Manager and external party send a `200 OK` response.

Call Establishment

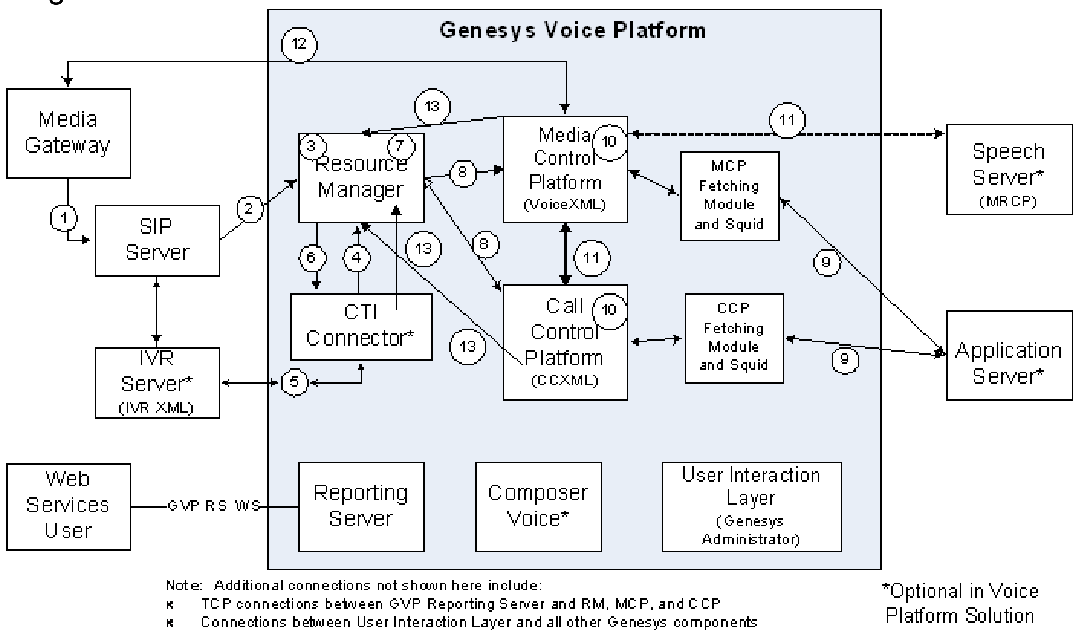
7. SIP Server sends an acknowledgement (ACK) to the Resource Manager and the external party.
    - The two call legs are bridged and the RTP session begins between the Media Server module of the Media Control Platform and the external party.
    - Simultaneously, SIP Server sends a message to the Supplementary Services Gateway indicating the call legs have been established and bridged (`EventEstablished` with a `CallStateOK` extension).
8. The Supplementary Services Gateway sends a request to SIP Server (TApplyTreatment) to execute the prepared dialog with the Media Server (Media Control Platform).
9. SIP Server then sends an INFO MSML message to the Media Server (Media Control Platform through Resource Manager).
    - It then sends a message back to the Supplementary Services Gateway to indicate that the call treatment has been applied (`TreatmentApplied` event).
10. At the end of the call the Resource Manager (for the Media Control Platform) responds with `200 OK` and BYE messages to the SIP Server.
    - SIP Server sends `200 OK` and BYE messages to the external party.
    - SIP Server sends an EventReleased message to the Supplementary Services Gateway.

Call Treatment Applied

11. After receiving the TreatmentApplied event from SIP Server, the Supplementary Services Gateway marks the call as a success in the database. Later, when the database records are cleaned up, the Supplementary Services Gateway sends this information to the trigger application in a Notification URL, along with the Token, RequestID, TenantName, IVRProfileName, CallUUID (unique ID that is generated by SIP Server), and other details.

Basic CTI Call Flow (Inbound)

The figure below illustrates how GVP handles an inbound call through IVR server while using the CTI Connector.



[+] Basic CTI Call Flow (Inbound) Description

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 VP Resource Manager (SIP INVITE).
3. The Resource Manager checks the SIP INVITE header. If the request is from a resource that is provisioned to be managed by the Resource Manager, the call is accepted, and a new session is created.
4. If the resource is a gateway resource that has the `use-cti` parameter set to `1`, the Resource Manager routes the call to a CTI Connector Resource group and selects a tenant or owner for the gateway resource. The Resource Manager forwards the request to a member of the CTI Connector group based on its load balancing scheme and adds the `gvp.rm.cti-call=1` parameter to the X-Genesys-GVP-Session-ID header.
5. When the CTI Connector receives the SIP INVITE from Resource Manager, the CTI Connector (acting as a B2BUA) fetches ANI, DNIS, CONNID, and UUID from the IVR Server.
6. The CTI Connector sends a new SIP INVITE to the Resource Manager with extension SIP header configured with:
    - the `user-part` of the Request-URI set to DNIS
    - the `user-part` of the FROM header set to ANI

- the `user-part` of the TO header set to DNIS

7. The Resource Manager searches the SIP INVITE from the CTI Connector based on the `sip-header-for-cti-dnis` parameter and maps the call to an IVR Profile, adding the `gvp.rm.tenant-id` parameter to the SIP header.
8. The Resource Manager sends the call to a Media Control Platform or Call Control Platform resource (SIP INVITE).
9. The call proceeds as in Steps 4 to 8 in Basic Inbound-Call Flow. In addition to the ASR, TTF, conference or audio play/record services described in that procedure's Step 7, the Media Control Platform or Call Control Platform can request and use additional CTI Connector services. The NGI voice application uses send or receive extensions to invoke the SIP INFO message between the MCP and the CTI Connector through Resource Manager to request the following mid-call features:
   - GetData
   - SetData (Add and Replace)
   - DeleteData (DeleteOne and DeleteAll)
   - GetStat
   - PeekStat
   - AccessNumGet

   Tip

   The send and receive extensions are used by NGI voice applications only (not by GVPi voice applications).

   Three transfer types are supported using CTI Blind, Bridge, or CTI transfer. The NGI voice application sends a request to transfer to a route DN to the CTI Connector with the Request URI parameter `RouteRequest=1`:

   - If the request is for a Blind transfer (SIP REFER), the VoiceXML session is terminated when the transfer to a route DN is successful.
   - If the request is for a Bridge transfer (SIP INVITE), the outbound-call leg from the Media Control Platform to the route DN is active until a connection to an available agent is complete.
10. The Resource Manager ends the call when one of the parties (the SIP end-user, the Media Control Platform, the Call Control Platform, or the CTI Connector) disconnects (SIP BYE) or when the call is transferred out of GVP (SIP REFER).
11. If the CTI Connector initiates the SIP BYE message, the Resource Manager passes interaction data back to the Media Control Platform in the BYE message. The Resource Manager then processes a custom SIP header, `X-Genesys-GVP-CDR`, and passes it on to the Reporting Server in the final CDR for the call.

Basic CTI Connector/ICM Call Flows (Inbound)

The call flows in this section illustrate how the CTI Connector and Cisco Intelligent Contact Management (ICM) framework handle call setup through ICM's Service Control Interface (SCI) and Call Routing Interface (CRI) for an inbound call.

Cisco ICM Type 8 Deployment Call Flow

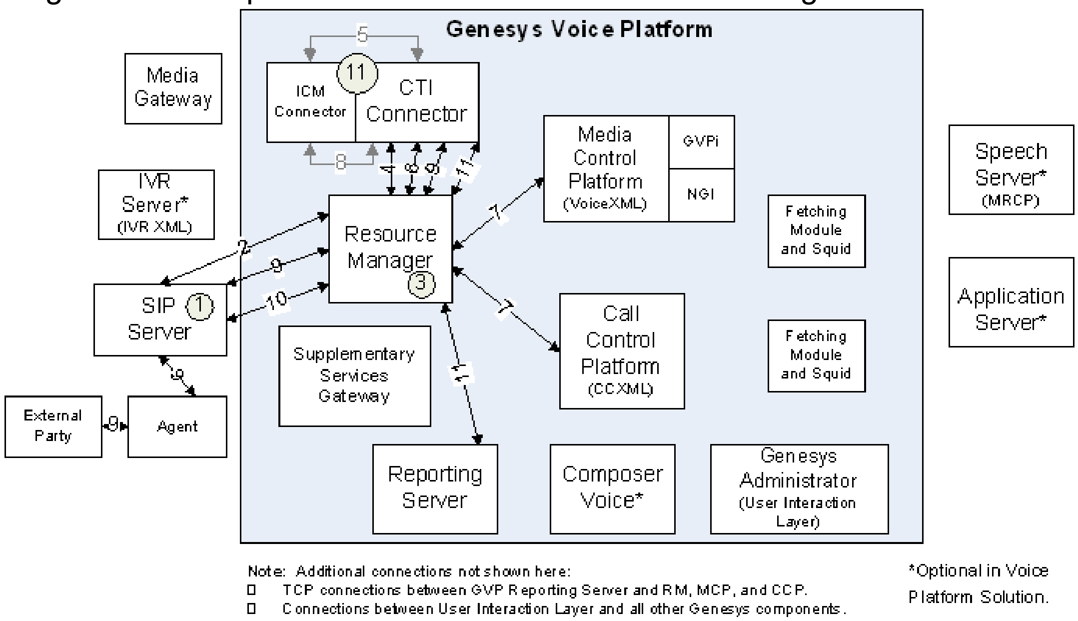
This section describes how GVP handles Cisco ICM Type 8 deployment calls, and does not include a diagram:

[+] Cisco ICM Type 8 Deployment Call Flow Description

1. The RM receives a call but does not fetch the IVR profile, because the use-cti parameter is set to `1`. Instead, the RM forwards the call to the CTI Connector.
2. Upon receiving a SIP_INVITE message from the RM, the CTI Connector prepares a REQUEST_INSTRUCTION message, because the enablePreRouting parameter is set to `true`.
3. The CTI Connector sends the Translation Route Number received as a user-part in the TO header of the SIP INVITE message in the DNIS field of the REQUEST_INSTRUCTION message to ICM.
4. When the RUN_SCRIPT_REQ message is received from ICM for the first time, the CTI Connector sets the user-part of the TO header in the INVITE message to RM with either ScriptID or Call Variable value, depending on the DNISIndicator parameter's value.
5. The RM fetches the IVR Profile and initiates a NETANN dialog toward MCP to execute the appropriate VXML application, and passes an empty ScriptID to it. When the VXML application completes execution, the MCP disconnects the call.
6. ICM optionally may send subsequent RUN_SCRIPT_REQUEST messages. For each of these subsequent messages, CTI Connector initiates a NETANN dialog to RM in the same IVR profile context. Since RM already knows about the IVR Profile, it replaces the SCRIPTURL with the initial-page-url that is configured in the IVR profile, and forwards it to the MCP. This continues till the caller disconnect the call, or all the scripts have executed from the ICM point of view.

SCI Transfer Call Flow

The figure below depicts a basic inbound call transfer through ICM s SCI interface.



[+] SCI Transfer Call Flow Description

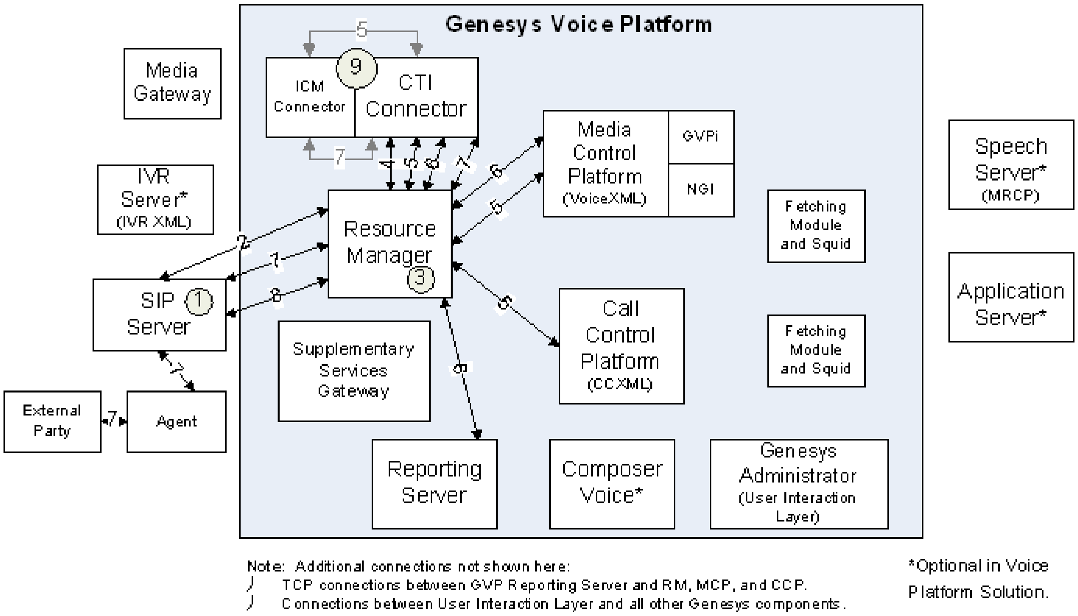
1. A call comes in to the SIP Server from an external source through a third-party media gateway.
2. The SIP Server passes the call to the Resource Manager using a SIP INVITE message.
3. The Resource Manager checks the header in the INVITE message to determine if the request is from a resource that is provisioned to be managed by the Resource Manager. If it is, the call is accepted and a new session is created.
4. If the resource is a gateway resource that has the use-cti configuration option set to `2`, the Resource Manager routes the call to a CTI Connector resource group and selects a tenant or owner for the gateway resource. The Resource Manager forwards the request to a resource in the CTI Connector group, based on its load balancing scheme, and then:
    - Adds the `gvp.rm.cti-call=1` option to the X-Genesys-GVP-Session-ID header.
    - Adds all of the CTI-related service parameters that are configured in the IVR Profile. (For example, `cti.icm.enableBridgeXfer`, `cti.icm.ScriptMapping`.)
    - Adds the `gvp-tenant-id=<tenant name>` in the `Request-URI` parameter.

    Where `<tenant name>` is the name of the tenant that to which the call belongs.)

5. When the CTI Connector receives the INVITE message from Resource Manager, the CTI Connector (acting as a B2BUA) interacts with the ICM and waits for its instructions, either to execute the scripts, transfer the call to an agent, or release the call.
6. After receiving a Script_Execution request (which includes the scriptID) from the ICM:
7. The CTI Connector generates a new SIP dialog towards the Resource Manager.
8. The Resource Manager sends the call to the Media Control Platform or Call Control Platform resource using an INVITE message.
    - The call proceeds as in Steps 4 to 8 in Basic Inbound-Call Flow.
    - When script execution is completed, the result (either success or failure) is sent back to the ICM. The result might also include Caller Entered Digits (CED), ICM call variables, or ECC variables.
    - The ICM might have multiple script execution requests before it send a request to transfer the call.
9. When all the scripts are executed, the ICM instructs the CTI Connector to initiate the transfer by returning the label and by specifying the type of transfer:
    - By default, the CTI Connector invokes a BLIND transfer to connect the caller to an agent by using the REFER method on the initial caller leg.
    - In the instruction, the ICM indicates that the BRIDGE transfer is to be used.
10. The CTI Connector then initiates a new SIP dialog towards an agent (through the Resource Manager) and bridges it with the caller leg.
    - The CTI Connector checks the IVR Profile s cti.icm.enableBridgeXfer configuration option value to identify the type of transfer. If the configuration option is enabled, the CTI Connector uses the BRIDGE transfer, otherwise it uses the BLIND transfer.
11. The Resource Manager ends the call when either the SIP end-user, Media Control Platform, or CTI Connector disconnects using a BYE message, or when the call is transferred out of GVP using a REFER message.
12. After the call is terminated:
    - The CTI Connector informs the ICM of call termination.
    - The CTI Connector passes the call disposition, (COMPLETED_IN_IVR, TRANSFERRED_TO_AGENT, or ABANDONED_IN_QUEUE) in the X-Genesys-GVP-CDR custom SIP header to Resource Manager. The Resource Manager then passes it to the Reporting Server in the final CDR for the call.

Inbound Call Using CRI

The figure below depicts a basic inbound call transfer through ICM s CRI interface.



[+] Inbound Call Using CRI Description

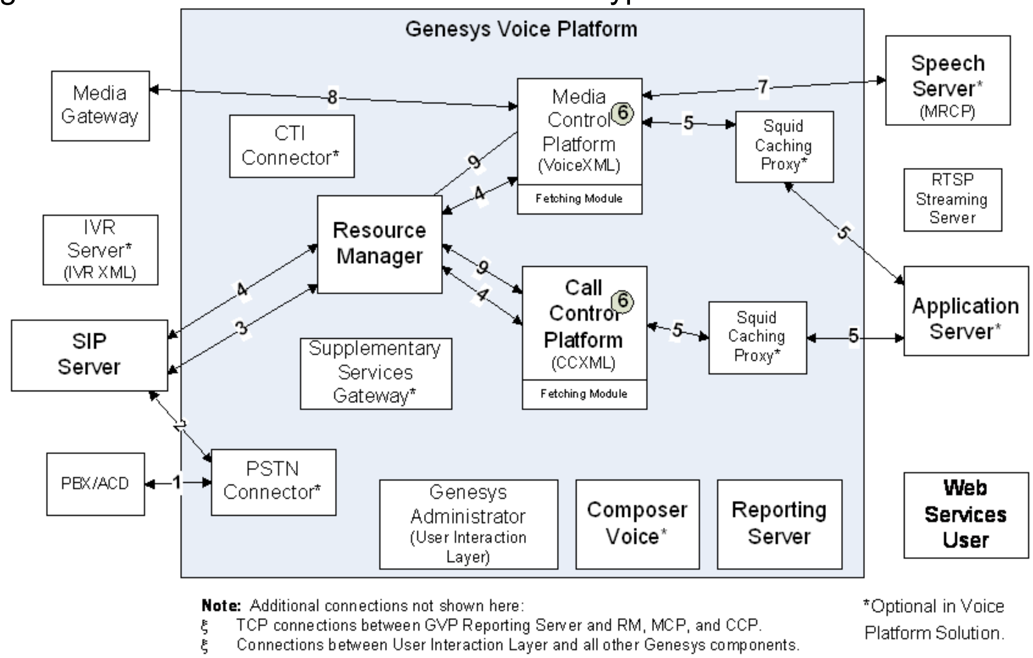
1. A call comes in to the SIP Server from an external source through a third-party media gateway.
2. The SIP Server passes the call to the Resource Manager using a SIP INVITE message.
3. The Resource Manager checks the header in the INVITE message to determine if the request is from a resource that is provisioned to be managed by the Resource Manager. If it is, the call is accepted and a new session is created.
4. If the resource is a gateway resource that has the `use-cti` configuration option set to `2`, the Resource Manager routes the call to a CTI Connector resource group and selects a tenant or owner for the gateway resource. The Resource Manager forwards the request to a resource in the CTI Connector group, based on its load balancing scheme, and then:
   - Adds the `gvp.rm.cti-call=1` option to the X-Genesys-GVP-Session-ID header.
   - Adds all of the CTI-related service parameters that are configured in the IVR Profile. (For example, `cti.icm.enableBridgeXfer`, `cti.icm.ScriptMapping`.)
   - Adds the `gvp-tenant-id=<tenant name>` in the `Request-URI` parameter.

   Where `<tenant name>` is the name of the tenant that to which the call belongs.)

5. When the CTI Connector receives the SIP INVITE from the Resource Manager (acting as a B2BUA) it informs the ICM that a new call has arrived and starts playing the VoiceXML application that is configured in the IVR Profile, as follows:
    - CTI Connector generates a new SIP dialog towards Resource Manager.
    - The Resource Manager sends the call to a Media Control Platform using a SIP INVITE message.
    - The call proceeds as in Steps 4 to 8 in Basic Inbound-Call Flow.
    - The VoiceXML application can send Caller Entered Digits (CED), ICM call variables, or ECC variables to CTI Connector that are passed to the ICM.
6. After the VoiceXML application plays the IVR, it sends a requests to the ICM and through a ROUTE request, determines which agent receives the call.
7. The NGI voice application sends a TRANSFER request to the CTI Connector with the ROUTE request in the `Request-URI` parameter set to `1`. The CTI Connector interacts with the ICM to obtain the final destination label, and determines the next step, as follows:
    - If the request is made using a SIP REFER message, the CTI Connector initiates a BLIND transfer using the REFER method on the caller leg.
    - If the request is made using a SIP INVITE message, the CTI Connector initiates a new SIP dialog to the agent and bridges the call with it.
8. The Resource Manager ends the call when either the SIP end-user, Media Control Platform, or CTI Connector disconnects using a BYE message, or when the call is transferred out of GVP using a REFER message.
9. After the call is terminated:
    - The CTI Connector informs the ICM of call termination.
    - The CTI Connector passes the call disposition, (COMPLETED_IN_IVR, TRANSFERRED_TO_AGENT, or ABANDONED_IN_QUEUE) in the X-Genesys-GVP-CDR custom SIP header to Resource Manager. The Resource Manager then passes it to the Reporting Server in the final CDR for the call.

Basic PSTN Call Flow (Inbound)

The figure below illustrates how GVP handles a typical inbound call from the PSTN network.



[+] Basic PSTN Call Flow (Inbound) Description

1. A call comes in from an external source through the TDM network, and The PSTN Connector detects an inbound-call trigger (through the Dialogic port).
2. The PSTN Connector converts the TDM signals to a SIP INVITE message adding the `X-Genesys-GVP-IVR-port` (used by the CTI Connector), and `X-Genesys-GVP-PSTNC-DBID` custom headers before sending it to SIP Server.
    - If the `prefix` parameter is configured for the PSTN Connector Trunk DN, then PSTN Connector also sends the `X-Genesys-GVP-Trunk-Prefix` custom header in the INVITE message.

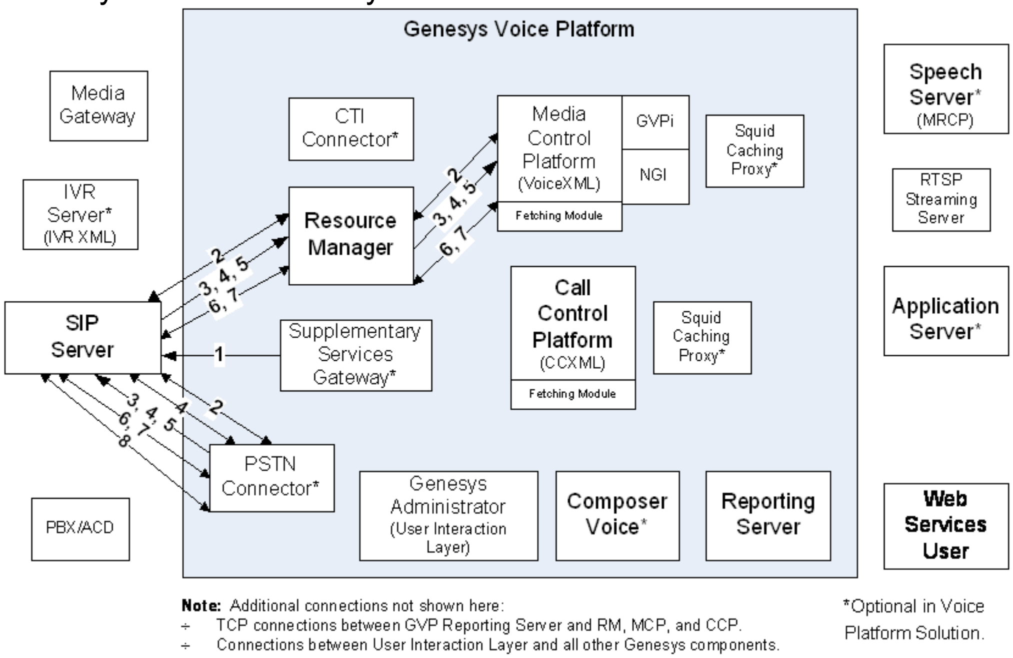
   The RTP prefill information is also added in the SDP to enable faster-than-real-time RTP from the Media Control Platform.
3. SIP Server passes the request to the Resource Manager (SIP INVITE).
4. When the Resource Manager receives the INVITE request, it passes it to the Media Control Platform.
5. The call proceeds as in Steps 4 to 9 in Basic Inbound-Call Flow.

Basic PSTN Call Flows (Outbound)

This section describes two basic outbound PSTN call flows.

Initiated by the Supplementary Services Gateway

The figure below illustrates how GVP handles a typical outbound call initiated by the Supplementary Services Gateway to the PSTN network:



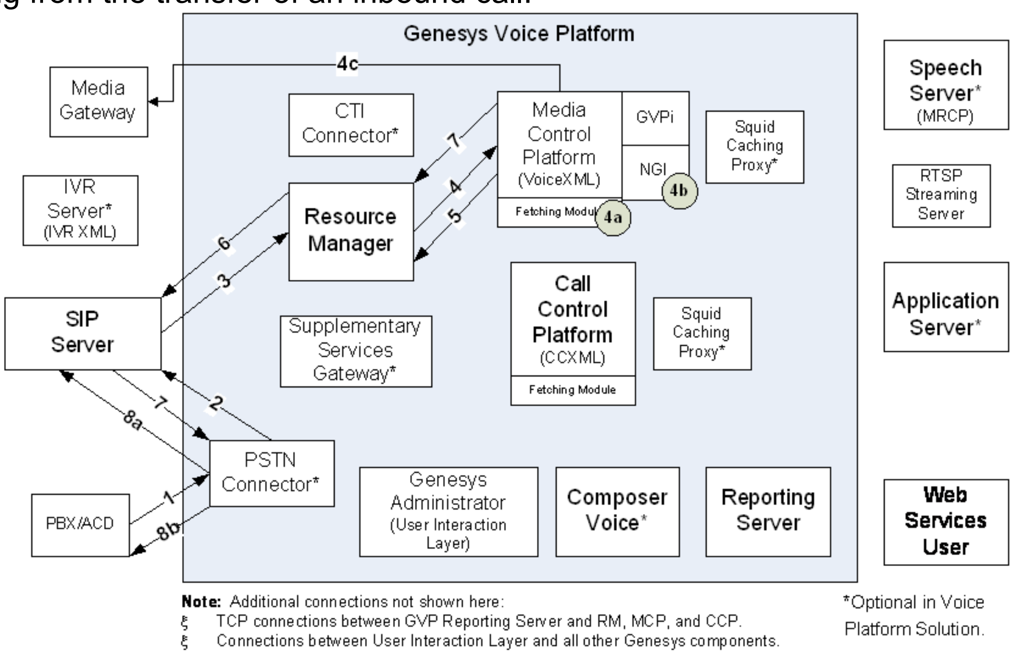
[+] Initiated by the Supplementary Services Gateway Description

1. The Supplementary Services Gateway triggers an outbound call through the TLib interface to SIP Server.
2. SIP Server sends a request for an outbound call to both the PSTN Connector and the Media Control Platform (through the Resource Manager). In this case SIP Server acts as a third-party call controller.
3. When the Media Control Platform receives the request for outbound-call initiation, it sends a SIP `200 OK` message (along with SDP negotiation information) to the Resource Manager.
4. SIP Server passes the SDP negotiation information (copied from the 200 OK message) in an INVITE message to the PSTN Connector.
5. The PSTN Connector sends a SIP 100 Trying message to the SIP Server to indicate that call initiation is in progress
6. The PSTN Connector extracts the ANI from the FROM header of the INVITE message. If the ANI is not specified, PSTNConnector is used as the default ANI.
7. When the TDM call is in the alerting state, the PSTN Connector sends a 180 Ringing message back to the SIP Server.
8. When the call is established between the PSTN Connector and the destination party, the PSTN Connector initiates the media session with the Media Control Platform by sending a SIP 200 OK message, including an X-Genesys-GVP-IVR-port custom

header. If the PSTN Connector accepts the SDP negotiation information from the initial INVITE, the 200 OK contains SDP in the reply.
9. If the PSTN Connector is unable to complete the call setup, it sends a SIP error code in the response to indicate the cause of the failure. For a complete list of SIP error codes, see the GVP 8.5 User's Guide.
10. The Media Control Platform sends a SIP ACK message (through the Resource Manager) to the PSTN Connector and the Media Control Platform, and the two-way media session is established.
11. If the PSTN caller hangs up the session terminates and the Media Control Platform sends a SIP BYE message (through the Resource Manager) to the PSTN Connector.
12. The call is dropped and the Dialogic port is cleared on the PSTN side. The PSTN Connector sends a SIP 200 OK to the Media Control Platform and the call is released.

Resulting From an Inbound Call Transfer

The figure below describes how GVP handles a typical outbound call to the PSTN network resulting from the transfer of an inbound call:



[+] Resulting From an Inbound Call Transfer Description

1. A call comes in from an external source through the TDM network and The PSTN Connector detects an inbound call trigger (through the Dialogic port).
2. The call proceeds as in Steps 2 to 4 in Basic PSTN Call Flow (Inbound).
3. 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.

4. The call proceeds as in Steps 5 to 9 in Basic Inbound-Call Flow.
5. When the Media Control Platform transfers the call, it generates an outbound INVITE (or REFER) message to the Resource Manager, passing on the `X-Genesys-Trunk-Prefix` header that was received in the inbound INVITE message.
6. The Resource Manager applies the prefix to the user part of the Request URI and the TO header of the outbound INVITE message (or to the `Refer-To` header for the outbound REFER message), and sends it to SIP Server.
7. SIP Server searches all available Trunks DNs to find the PSTN Connector DN that best matches the prefix to ensure the chosen PSTN Connector is the same one that initiated the inbound call.
    - If the value of the replace-prefix option is set to <empty string> in the Trunk DN, SIP Server strips the prefix from the INVITE message before sending it to the PSTN Connector.
8. The call can proceed in one of two ways, depending on the type of transfer:
    a. For bridge transfers (INVITE from the Media Control Platform):
        - The PSTN Connector performs route-based dialing, or port-based dialing, depending on how the VoiceXML application is configured, and pass on an Authorization code if this information is specified in the value of the `X-Genesys-GVP-PSTNC-Data` custom SIP header such as, `Route=n;PortDlgc=x-y;AuthCode=<code>`.
        - The call proceeds as in Steps 5 to 10 in the Initiated by the Supplementary Services Gateway Call Flow.
    b. For blind transfers (REFER from the Media Control Platform):
        - The PSTN Connector initiates the appropriate blind transfer on the network side and both call legs (TDM and SIP) are terminated.

Specifications and Standards

This topic describes the specifications and standards that Genesys Voice Platform supports.

- Specifications
- Related Standards
- RFC 5552 Support

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.
- Media Resource Control Protocol (MRCP) Specification Requirements for Distributed Control of Automatic Speech Recognition (ASR) Speaker Identification/Speech Verification (SI/SV), and Text-to-Speech (TTS) Resources (2005). MRCP version 1(MRCPv1) (2006).
- 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.
- Semantic Interpretation for Speech Recognition Semantic Interpretation for Speech Recognition (SISR) Version 1.0, W3C Recommendation, 5 April 2007.
- CCXML Specification W3C Voice Browser Call Control: CCXML Version 1.0, W3C Working Draft, 29 June 2005.

The following specifications and recommendations are published and maintained by the International Telecommunications Union, Telecommunication Standardization Sector (ITU-T):

- I.431 Specification Layer 1 specifications for ISDN PRI networks using either an E1 or T1 circuit. The I.431 standard represents the PRI physical layer.
- Q931 Recommendation ISDN user-network interface layer 3 specification for basic call control.

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 the Internet Engineering Task Force (IETF) publishes and maintains. For more information, see `http://www.ietf.org`.

The IETF standards that GVP supports include the following:

- RFC 1738 Uniform Resource Locators.
- RFC 1808 Relative Uniform Resource Locators.
- RFC 1867 Form-Based File Upload in HTML.
- RFC 2046 Multipurpose Internet Mail Extensions (MIME), Part Two: Media Types.
- 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 2327 SDP: Session Description Protocol.
- 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 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers.
- RFC 2616 Hypertext Transfer Protocol HTTP/1.1 (subset).
- RFC 2782 A DNS RR for specifying the location of services (DNS SRV).
- 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 3023 XML Media Types.
- RFC 3261 SIP: Session Initiation Protocol.
- RFC 3262 Reliability of Provisional Responses in the Session Internet Protocol (SIP).
- RFC 3263 Session Initiation Protocol (SIP): Locating a SIP Server.
- RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP).
- RFC 3265 SIP-Specific Event Notification.
- RFC 3266 SDP: Session Description Protocol.
- 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 3323 A Privacy Mechanism for the Session Internet Protocol (SIP)

- RFC 3325 Private Extensions for the Session Internet Protocol (SIP) for Asserted Identity within Trusted Networks.
- RFC 3326 The Reason Header Field for the Session Internet Protocol (SIP).
- RFC 3455 Private Header (P-Header) Extensions to SIP.
- RFC 3515 The SIP REFER Method.
- RFC 3550 RTP A Transport Protocol for Real-Time Applications.
- RFC 3581 An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing.
- RFC 3611 RTP Control Protocol Extended Reports (RTCP XR)
- RFC 3624 The Media Gateway Control Protocol (MGCP) Bulk Audit Package.
- RFC 3711 The Secure Real-Time Protocol (SRTP).

Notes:
  - Some options in the SRTP specification are not supported, including: the Master Key Index (MKI), key derivation rates other than zero, the cipher F8, anti-replay lists with sizes other than 128, and the use of the packet index to select between master keys.
  - Media Server (MS) does not support the FEC_ORDER and FEC_KEY session parameters.
    …where FEC stands for Forward Error Correction. Refer to Section 6 of RFC 4568 for details.
  - MS ignores the WSH (Window Size Hint) parameter and assumes a value of 128.
  - MS supports only the default master key lifetime.
  - MS rejects the media line containing crypto: attribute if the RTP profile signaled (specified?) is not RTP/SAVP.
  - MS rejects the media line if the keying method is not inline: (the only keying method that is specified in RFC 4568).

- RFC 3891 The Session Internet Protocol (SIP) Replaces Header.
- RFC 3892 The Session Initiation Protocol (SIP) Referred-By Mechanism.
- RFC 3984 RTP Payload Format for H.264 Video.
- RFC 3986 Uniform Resource Identifier (URI): Generic Syntax.
- RFC 4028 Session Timers in the Session Initiation Protocol (SIP).
- RFC 4240 Basic Network Media Services with SIP GVP provides support for conference services and dialogs.
- RFC 4244 An Extension to the Session Internet Protocol (SIP) for Request History Information.
- RFC 4463 Media Resource Control Protocol (MRCP) for client control of media resources, such as ASR and TTS.
- RFC 4566 SDP: Session Description Protocol.
- RFC 4568 Session Description Protocol (SDP) Security descriptions for media streams.

- RFC 4733 RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals.
- RFC 4753 Elliptic Curve Project (ECP) Groups for Internet Key Exchange (IKE) and IKEv2.
- RFC 4867 RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codes.
- RFC 5168 XML Schema for Media Control.
- RFC 5552 SIP Interface to VoiceXML Media Services.
- RFC 5707 Media Server Markup Language (MSML).
- Most extensions from the proposed IETF draft SIP Interface to SIP Interface to VoiceXML Media Services. See *RFC 5552 Support*, below.

The platform also includes complete support for many network-related protocols and other protocols (for example, TCP/IP and SNMP). Contact Genesys for more information.

RFC 5552 Support

This section describes various aspects of RFC 5552 (previously named *Burke Draft*) and features of the RFC 5552 that GVP 8.5.x supports.

RFC 5552 describes the Session Initiation Protocol (SIP) interface to VoiceXML media services. This section covers the following RFC 5552 sections:

- VoiceXML session establishment and termination
- Media support
- Returning data to the application server
- Outbound calling
- Call transfer

The table below lists and describes RFC 5552 requirements, as well as the ones that GVP 8.5 supports:

Table 54: RFC 5552 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]	Supported

	• `aai` [RFC4627]	
	Support for incorrectly formed requests with the 4xx class response.	Not supported
	Support for repeated init-parameters that are 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 an INVITE request 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.	Supported
	• If the Request-URI does not include a voicexml parameter, and the VoiceXML Media Server does not elect to use a default page, the VoiceXML Media Server must return a final response of 400 Bad Request, and it should include a Warning header that contains a three-digit code of 399 and a human-readable error message. • If the VoiceXML document cannot be fetched or parsed, the VoiceXML Media Server must return a final response of 500 Server Internal Error and should include a Warning header that contains a three-digit code of 399 and a human-readable error message.	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 a VoiceXML application until a re-INVITE request with port information is sent, if any of the	Supported

	following conditions are met: starting a Dialog-INVITE request without media; a 200 OK with offered media; an ACK with media, but the media ports are set to zero (0); or an INVITE request with SDP without media lines, followed by a regular INVITE, 200 OK, or ACK flow.	
	Support for a re-INVITE request that disables media stream without affecting the executing VoiceXML application when 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
	Support for evaluating the following session variables: • `session.connection.protocol.name` to sip • `session.connection.protocol.version` to 2.0	Supported
	Support for use of the `session.connection.protocol.sip.requesturi` variable as an associative array that is formed from the URI parameters.	Supported
	Support for use of the `session.connection.protocol.sip.media` variable as an array, where each array element is an object that has the `INSERT_TEXT`following properties: • type. • direction. • format. This parameter will be updated as the media values that are involved in the session change.	Supported
2.5	Support for sending a 200 OK message in response to a BYE request, 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>` message, the VoiceXML application completes, or the VoiceXML application has unhandled errors.	Supported
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 use of a re-INVITE request to modify a media session.	Supported
3.4	Support for the following codecs: • G.711 `u-law` and `A-law` using the RTP payload type 0 and 8 Set `mpc.codec` to `pcmu pcma`. • G722 to convert audio signals to uniform digital signals Set `mpc.codec` to `g722`. • H.263 Baseline Set `mpc.codec` to `h263`. • H 264 Baseline Set `mpc.codec` to `h264`. • AMR-NB audio Set `mpc.codec` to `amr`. • AMR-WB audio Set `mpc.codec` to `amr-wb`. • 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
4.1	Support for returning data to the application server with an HTTP Post by using the `<submit>`, `<subdialog>`, or `<data>`	Supported

	elements.	
4.2	Support for returning data to the application server by using the SIP expression or 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 use of the _reason reserved name to differentiate between a BYE result from a `<disconnect>` element and a BYE result from an `<exit>` element. For example, `_reason = exit`.	Supported
	Support for use of the _exit reserved name if the expr attribute is specified on the `<exit>` element, instead of the namelist attribute. For example, `_exit=<value>`.	Supported
5.0	Support for triggering outbound calls by using third-party call control [RFC3725].	Supported
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:	Supported

	• 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.	
	Support for appending the aai or aaiexpr attribute to the Refer_To URI as a parameter that is 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 that is 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 form attribute of the <transfer> request to noanswer after issuing a CANCEL request when `connectiontimeout` expires.	Supported
	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 that is 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 response 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

Prerequisites and Planning

This chapter describes the prerequisites and planning considerations for the deployment of Genesys Voice Platform (GVP) 8.5 on Windows and Linux operating systems and includes information about the required software. It contains the following sections:

- GVP Installation DVDs
- Prerequisites
- Dialogic Telephony Cards
- Antivirus Software
- Host Setup
- PSTN Connector and GVPi Support in 8.1.5
- Voice Platform Solution Components
- HMT Permissions and Access Rights

GVP Installation DVDs

Genesys Voice Platform 8.1.3 and above components are shipped on two DVDs one containing the Genesys Voice Platform components and one containing the Genesys Media Server components. The 8.1.2 and earlier components are shipped on one DVD. The components on each DVD are listed in the table below:

Table: Contents of Release DVDs

Component	8.5.1	8.5.0	8.1.7	8.1.6	8.1.5	8.1.4	8.1.3	8.1.2
Genesys Voice Platform (DVD #1)								
Resource Manager (RM)								?
Media Control Platform (MCP)								?
Call Control Platform (CCP)	?	?	?	?	?	?	?	?
Reporting Server (RS)					?	?	?	?
Squid Caching Proxy	?	?	?	?	?	?	?	?
Supplementary Services Gateway (SSG)	?	?	?	?	?	?	?	?
Computer Telephony Integration (CTI) Connector	?	?	?	?	?	?	?	?
Public Switched Telephone Network (PSTN) Connector						?	?	?
Policy Server (PS)	?	?	?	?	?	?		

Media Resource Control Protocol (MRCP) Proxy	?	?	?	?	?	?		
Genesys Media Server (DVD #2)								
Resource Manager (RM)	?	?	?	?	?	?	?	NO DVD
Reporting Server (RS)	?	?	?	?				NO DVD
Media Control Platform (MCP)	?	?	?	?	?	?	?	NO DVD
GVP Reporting Plugin for GAX	?	?	?	?				NO DVD
Management Information Bases (MIB)	?	?	?	?	?	?	?	NO DVD

Important

The Fetching Module is integrated with the Media Control and Call Control Platforms and is included in those Installation Packages (IP). The Squid Caching Proxy, and CTI Connector are included for Windows only. The PSTN Connector is for Windows and Linux. The Genesys Composer installation software ships on a separate DVD.

The PSTN Connector and GVPi are not included in the releases above 8.1.4, but they are still supported. For more information about how these components are supported, see PSTN Connector and GVPi Support in 8.1.5.

Tip

GVP is installed, provisioned, and managed by using Genesys Administrator. Ensure that you have Genesys Administrator installed as part of your deployment. For information about Genesys Administrator, see the Framework Deployment Guide.

Prerequisites

Important

Genesys recommends that you review Preparing the Hosts for GVP and Task Summary: Preparing Your Environment for GVP, Task Summary: Preparing Your Environment for GVP (Windows), or Task Summary: Preparing Your Environment for GVP (Linux) before you install any software.

Software Requirements for Windows

The table below summarizes the software requirements for GVP 8.5 deployments on Windows.

Table: Software Requirements for Windows

Category	Requirements and comments
Operating system on GVP servers	
Genesys Voice Platform 8.5 (Mandatory)	Microsoft Windows Server 2003, SP2 and 2008: • 64-bit binaries running on 64-bit OS (optimal performance) • 32-bit binaries running on 64-bit OS • 32-bit binaries running on 32-bit OS Microsoft Windows Server 2008 R2: • 64-bit binaries running on 64-bit OS (optimal performance) • 32-bit binaries running on 64-bit OS **Notes:**

	• Microsoft Visual C++ is installed automatically with the GVP IPs.
Operating system supporting components	
Reporting Server and Policy Server	Sun Java Runtime Environment (JRE) 7.0 or later **Notes:** • Download the Sun JRE platform software from the Sun Microsystems website. • If using Windows 2008 64-bit, download the 64-bit Sun JRE platform.
Microsoft Internet Information Services (IIS) 6.0 components (Mandatory for GVP 8.1.1 and earlier 8.x releases)	Common files: • IIS Manager Snap-In for Microsoft Management Console • World Wide Web server Install these component from the Windows 2003 CD either by using Add/Remove Programs or downloading them from the Microsoft website. **Notes:** • In GVP 8.1.1 and earlier releases, IIS was required to host inline and universal hotkey grammar files that were fetched by ASR. • In GVP 8.1.2 and later releases, IIS is not required, unless you are using GVPi. • If GVPi is used, IIS MUST be installed before installing the MCP.
Reporting Server Database requirements	Required only on GVP servers that have Reporting Server DB installed: • Microsoft SQL Server 2008 (clustered and/or replicated), or 2005 SP2 (Standard and Enterprise editions), or Oracle 10g, 10g Real Application Cluster (RAC), or 11g RAC Database Server (Standard and Enterprise editions) **Notes:**

	• For additional information about supported operating systems for the Reporting Server Database, see Host Setup on page 193. • Download the SQL Server or the Oracle Database Server software from the vendor's website. It is your responsibility to obtain the appropriate licenses for this software.
Management and monitoring tools (Optional)	• Genesys Simple Network Management Protocol (SNMP) Master Agent • SNMP Network Management Software (NMS)(Optional) **Notes:** • The Genesys SNMP Master Agent is installed on the same host(s) as the VP Resource Manager, VP Media Control Platform, VP Call Control Platform, and VP Fetching Module components. • Install the Genesys SNMP Master Agent software from the Genesys Management Framework Installation CD. You can use any type of SNMP NMS for example, HP OpenView, BMC Software Patrol, or Ipswitch WhatsUpGold.
Specific services and settings (Mandatory)	You must configure certain specific services and settings on each host before you install GVP. For more information, see Windows Services and Settings
Web browser (for administration)(Mandatory)	Used only from the administrator's desktop: • Microsoft Internet Explorer (IE) 6.0, SP1, up to and including 8.0. • Firefox 2.x, up to and including 3.6.
Third-party supporting components	
Third-party TDM interface (Mandatory for PSTN Connector only)	If you are installing the PSTN Connector: • Dialogic v6.0 • Dialogic Service Update 241 for Windows • Dialogic Service Update 327 for Linux
Automatic speech recognition (ASR) (Optional)	Genesys recommends that the ASR servers are installed and operational before you install the Genesys Voice Platform. Genesys has validated the following third-party ASR software:

	• Nuance Recognizer 10.2.3 with Nuance Speech Server (NSS) 6.2.4 • Nuance Recognizer 9.0.18 with Nuance Speech Server (NSS) 5.1.7 • Telisma Telispeech ASR 2.0 SP1. • IBM WebSphere Voice Server (WVS) 6.1.1 ASR or higher. **Notes:** • It is your responsibility to obtain the software and the appropriate licenses. Media Resource Control Protocol version *(MRCPv1) and MRCP version 2 (MRCPv2) are supported. • For more speech information, see the Genesys Supported Media Interfaces Reference Manual.
Text-to-speech (TTS) (Optional)	Genesys recommends that the TTS servers are installed and operational before you install the Genesys Voice Platform. Genesys has validated the following third-party TTS software: • Nuance Vocalizer 6.0.2 with Nuance Speech Server (NSS) 6.2.5 • Nuance Vocalizer 5.7.3 with Nuance Speech Server (NSS) 6.2.4 • IBM WebSphere Voice Server (WVS) 6.1.1 TTS or later, with IBM TTS connector **Notes:** • It is your responsibility to obtain the software and the appropriate licenses. • MRCPv1 and MRCPv2 are supported. • For more speech information, see the Genesys Supported Media Interfaces Reference Manual.

Software Requirements for Linux

The table below summarizes the software requirements for GVP 8.5 deployments on Linux:

Table: Software Requirements Linux

<table>
<tr><th>Category</th><th>Requirements and comments</th></tr>
<tr><th colspan="2">Operating System on GVP Servers</th></tr>
<tr><td>For Genesys Voice Platform 8.5 (Mandatory)</td><td>Red Hat Enterprise Linux 5.x Advanced Platform
• 64-bit binaries running on 64-bit OS (optimal performance)
• 32-bit binaries running on 64-bit OS
• 32-bit binaries running on 32-bit OS
• Red Hat Enterprise Linux 4 Advanced Server
• 32-bit binaries running on 32-bit OS</td></tr>
<tr><td>Reporting Server and Policy Server</td><td>Sun Java Runtime Environment (JRE) 7.0 or later.

Notes:

• Download the Sun JRE platform software from the Sun Microsystems website. If using Windows 2008 64-bit, download the 32-bit Sun JRE platform.</td></tr>
<tr><th colspan="2">Operating System Supporting Components</th></tr>
<tr><td>Reporting Server Database</td><td>Required only on GVP servers that have Reporting Server DB installed:
• Oracle 10g, 10g, or 11g Real Application Cluster (RAC) Database Server (Standard or Enterprise editions).

Notes:

• For additional information about supported operating systems for the Reporting Server Database, see Host Setup on page 193.
• Download the Oracle Database Server software from the Oracle website. It is your responsibility to obtain the appropriate licenses for this software.</td></tr>
<tr><td>Apache HTTP Server (Mandatory for GVP 8.1.1 and earlier 8.x releases)</td><td>• httpd-2.0 or later.

Notes:

• Install Apache on the Media Control Platform and Call Control Platform host(s) before you install the GVP components.
• In GVP 8.1.1 and earlier releases, Apache HTTP Server was required to host inline and universal hotkey grammar files that were fetched by ASR.
• In GVP 8.1.2, Apache is no longer required. The Media Control Platform now transmits these grammars by default in the MRCP requests.</td></tr>
</table>

<table>
<tr><td>Management and monitoring tools (Optional)</td><td>Genesys Simple Network Management Protocol Master Agent.
SNMP Network Management Software (optional).
Notes:
The Genesys SNMP Master Agent is installed on the same host(s) as the VP Resource Manager, VP Media Control Platform, VP Call Control Platform, and VP Fetching Module components.
Install the Genesys SNMP Master Agent software from the Genesys Management Framework Installation CD. See . You can use any type of SNMP NMS for example, HP OpenView, BMC Software Patrol, or Ipswitch WhatsUpGold.</td></tr>
<tr><td>Specific services and settings (Mandatory)</td><td>Notes:
You must configure certain specific services and settings on each host before you install GVP.
For more information, see the Task Summary: Preparing Your Environment for GVP (Linux), on page 339.</td></tr>
<tr><td>Web browser (for administration) (Mandatory)</td><td>Used only from the administrator's desktop:
Microsoft Internet Explorer 6.0, SP1 or 7.0</td></tr>
<tr><td colspan="2">Third-party Supporting Components</td></tr>
<tr><td>Automatic speech recognition (Optional)</td><td>Genesys recommends that the ASR servers are installed and operational before you install the Genesys Voice Platform. Genesys has validated the following third-party ASR software:
Nuance Recognizer 10.2.3 with Nuance Speech Server (NSS) 6.2.4
Nuance Recognizer 9.0.18 with Nuance Speech Server (NSS) 5.1.7
Nuance Recognizer 9.0.16 with Nuance Speech Server (NSS) 5.0.10
Telisma Telispeech ASR 2.0 SP1.
IBM WebSphere Voice Server (WVS) 6.1.1 ASR or higher.
Notes:
It is your responsibility to obtain the software and the appropriate licenses. MRCPv1 and MRCPv2 are supported.</td></tr>
</table>

	• For more speech information, see the Genesys Supported Media Interfaces Reference Manual.
Text-to-speech (Optional)	Genesys recommends that the TTS servers are installed and operational before you install the Genesys Voice Platform. Genesys has validated the following third-party TTS software: • Nuance Vocalizer 6.0.2 with Nuance Speech Server (NSS) 6.2.5 • Nuance Vocalizer 5.7.3 with Nuance Speech Server (NSS) 6.2.4 • IBM WebSphere Voice Server (WVS) 6.1.1 TTS or later, with IBM TTS connector **Notes:** • It is your responsibility to obtain the software and the appropriate licenses. • MRCPv1 and MRCPv2 are supported. • For more speech information, see the Genesys Supported Media Interfaces Reference Manual.

Dialogic Telephony Cards

The PSTN Connector relies on Dialogic hardware and software to provide a gateway solution for customers with existing TDM-based networks to simplify the integration and migration to the GVP IP-based solution.
These Dialogic telephony cards are supported on Windows:

- Dialogic DM/V480A
- Dialogic DM/V1200BTEP
- Dialogic DM/V960A
- Dialogic D240JCT
- Dialogic DM/V600A
- Dialogic D480JCT
- Dialogic DM/V1200A
- Dialogic D300JCT
- Dialogic DM/V600BTEP
- Dialogic D600JCT

The PSTN Connector does not impose a limit on the number of cards it can support. Limitation arises from the number of PCI slot in the machine and the load on the PCI bus.

Dialogic Software

GVP supports Dialogic v6.0 with Service Update 241 for Windows, and Service Update 327 with RHEL Linux.

For information about how to install and configure the PSTN Connector, see Installing GVP with the Deployment Wizard and Provisioning the PSTN Connector.

Antivirus Software

Antivirus software can affect system performance and call response time. In an ideal deployment, antivirus software is disabled on GVP systems. However, Genesys understands the need to have antivirus protection on servers and, therefore recommends, at a minimum, that you exclude the GVP directory from virus scanning, and that you schedule system scans to occur at times when traffic is low.

Also, be aware that antivirus software may interfere with the installation of GVP during initial deployment. Make sure that the server is not running antivirus software, or any other third-party software, during installation.

Host Setup

GVP provides some flexibility in combining various components on one host; however, the following restrictions apply:

- If you are installing Genesys Administrator and (a single instance of) the Media Control Platform on the same host, you must install GVP by using the manual procedures and ensure that Genesys Administrator is shut down during the installation. Genesys does not recommend that you install Genesys Administrator on a host that has multiple instances of the Media Control Platform.
- If the Resource Manager is in active standby High Availability (HA) mode, Genesys recommends that other SIP components that communicate with the Resource Managers are installed on different servers, unless they support static routing and do not interfere with the Resource Manager's HA mechanism. When the Resource Manager is in active backup mode, it uses Network Load Balancing (NLB) (on Windows) or Virtual IP takeover (on Linux or Windows). Other SIP HA components (for example, SIP Server) that use the same HA mechanism as the Resource Manager can interfere if deployed on the same servers within the cluster. In addition, when a Virtual IP address is used, Windows NLB has a limitation, where the Virtual IP always resolves to localhost on servers within the NLB cluster.

- If you are installing the Media Control Platform and the PSTN Connector on the same host, ensure the value of the `rtpthreadlevel` option in the `mpc` section of the Media Control Platform to TIME_CRITICAL.

The following are some additional restrictions or requirements:

- In GVP 8.1.2 and Genesys Administrator 8.0.3, multiple instances of the Media Control Platform on a single server are supported. See Deploying Multiple Media Control Platforms.
- In GVP 8.1.2 and above, the Fetching Module is integrated with the Media and Call Control Platforms and the Squid proxy is optional.
- The Reporting Server can be deployed with one Resource Manager instances only, unless the Resource Manager is deployed in HA mode. When the Resource Manager is in HA mode, the Reporting Server recognizes the HA pair as a single instance.
- The Reporting Server Database (DB) is supported in the following ways:
    - The Reporting Server DB does not have to reside on the server where Reporting Server is installed.
    - The Reporting Server DB can be installed on Windows or Linux.
- The Reporting Server DB and the Reporting Server can be installed on different operating systems. (For example, the Reporting Server can be on Windows and the DB on Linux).

Tip

There are additional restrictions for the Reporting Server host if it is configured for High Availability, see Reporting Server High Availability.

- You can mix GVP components that are installed on different operating systems within a deployment.

PSTN Connector and GVPi Support in 8.1.5

The PSTN Connector and Legacy GVP VoiceXML interpreter (GVPi) are not included in the 8.1.5 and above releases, but they are still supported and can be deployed in 8.1.5 environments. However, at least one instance of the 8.1.4 Media Control Platform is required to provide capability-based routing of PSTN Connector and GVPi requests for media services.

In other words, the 8.1.4 Media Control Platform can interoperate with all GVP 8.1.5 components, but the 8.1.4 PSTN Connector and GVPi can not.

For example, your 8.1.5 environment might be deployed in the following way:

- A pool of 8.1.5 Media Control Platform instances is deployed to enable new media capabilities like video.
- A pool of 8.1.4 Media Control Platform instances is deployed to support GVPi and/or the PSTN Connector (together in the same environment).

To do this, you must provision two separate MCP Logical Resource Groups (LRGs), each with different capabilities. For example:

- The 8.1.5 MCP LRG must have video and other media capabilities configured.
- The 8.1.4 MCP LRG must have the GVPi and PSTN Connector.
- The Resource Manager s capability-based routing feature must be configured to ensure that the PSTN Connector (using GVPi) calls are handled by the 8.1.4 MCP Resource Group only.
- To ensure the correct LRG processes the calls, create an IVR profile that requests the PSTN Connector/GVPi capability-based routing features (they must match the ones that are defined in the LRG).

To configure the Resource Groups and IVR Profiles to support this configuration, see the sections Using Resource Groups and Creating IVR Profiles and DID Groups, and "IVR Profile Configuration for GVPi" in the Genesys Voice Platform 8.5 User's Guide.

Voice Platform Solution Components

This section describes the recommended, required and optional components, and the dependencies present in a successful deployment of a Voice Platform Solution.

> **Important**
>
> The table below lists the versions of Management Framework components and SIP Server that are recommended for each GVP release. However, the newest GVP version may still be compatible with a previous version of SIP Server, Genesys Administrator or Configuration Server. Please verify

with Genesys Customer Care if you wish to keep a previous version of any of these components.

Table: Versions Compatible With GVP

GVP	Management Framework		SIP Server
	Genesys Administrator	Configuration Server	
8.5.1	8.1.3	8.1.3	8.1.1
8.5.0	8.1.3	8.1.3	8.1.1
8.1.7	8.1.3	8.1.3	8.1.1
8.1.6	8.1.3	8.1.2	8.1.0
8.1.5	8.1.2	8.1.1	8.1.0
8.1.4	8.1.0	8.1.0	8.0.4
8.1.3	8.0.3	8.02	8.0.3
8.1.2	8.0.3	8.02	8.0.3
8.1.1	8.0.11	8.0.1	8.0.2
8.1.0	8.0.1	8.0.1	8.0.2

Tip

If you plan to install the MRCP Proxy and Policy Server, you must upgrade to Genesys Administrator 8.1.0 and Configuration Server 8.1.1. If not, the 8.0.2 versions are acceptable and compatible with all other GVP 8.1.4 components.

Voice Platform Solution and Dependencies

The following is an overview of a VPS and the associated dependencies:

- A centralized instance of Genesys Management Framework that includes the following components:
    - Configuration Database
    - Log DB Server

  - Microsoft SQL Server or Oracle Database Server
  - Configuration Server
  - Genesys Administrator
  - Solution Control Server
  - Solution Control Interface (optional)
  - Message Server
  - Local Control Agent required on all GVP 8.5.0 hosts
  - Optional: Genesys SNMP Master Agent on all GVP 8.5 hosts
    (See Table: Versions Compatible With GVP for Management Framework versions that are compatible with each GVP release.)
- Session Initiation Protocol (SIP) Server
- IVR Server 8.0
- Stat Server
- Universal Routing Server
- T-Server (switch-specific)
- Voice Platform (VP) Resource Manager:
  - Mandatory component one or more per deployment
  - Can be deployed as an active active or active standby pair for high availability
  - Prerequisite: Local Control Agent
  - Optional: SNMP Master Agent
- VP Media Control Platform:
  - Mandatory component one or more per deployment
  - Prerequisite: Local Control Agent
  - Optional: SNMP Master Agent
- VP Call Control Platform:
  - Optional component one or more per deployment
  - Prerequisite: Local Control Agent
  - Optional: SNMP Master Agent
- VP Reporting Server:
  - Optional component one or more per deployment
  - Prerequisite: Local Control Agent
  - Prerequisite: Database Server
    (Microsoft SQL Server 2005, 2008 or Oracle 10 g, 11g)
  - Prerequisite: Sun Java Runtime Environment (JRE) 6.0, Update 19 or later; Sun JRE 7.0 or later recommended
  - Optional: SNMP Master Agent
- VP CTI Connector:
  - Optional component one per deployment
  - Prerequisite: Local Control Agent
  - Prerequisite: IVR Server or Cisco Intelligent Contact Management (ICM) (based on the deployment)

    - Optional: SNMP Master Agent
- VP PSTN Connector:
    - Mandatory component for TDM integration many per deployment
    - Prerequisite: Local Control Agent
    - Prerequisite: Dialogic v6.0 with Service Update 241
    - Optional: SNMP Master Agent
- VP Supplementary Services Gateway:
    - Optional component many per deployment
    - Prerequisite: Local Control Agent
    - Optional: SNMP Master Agent
- VP Policy Server
    - Optional component many per deployment
    - Prerequisite: Local Control Agent
    - Optional: SNMP Master Agent
- VP MRCP Proxy
    - Optional component many per deployment
    - Prerequisite: Local Control Agent
    - Optional: SNMP Master Agent

Tip

You can deploy many UCM Connectors in your environment. However, a single UCM Connector can interact with only one Cisco T-Server.

Alternatively, a single Cisco T-Server can interact with multiple UCM Connectors.

VPS Components Minimum Deployment

At a minimum, the following components are required to deploy the VPS:

- Management Framework components
- Genesys Administrator
- SIP Server
- GVP components
    - One Resource Manager
    - One Reporting Server (optional)
    - One Media Control Platform
    - Fetching Module
    - Squid Caching Proxy

Tip

For GVP version 8.1.2 and above, the Fetching Module is integrated with the Media Control and Call Control Platforms and is no longer a separate Installation Package. Also, the Squid caching proxy is optional.

Optional Components

The following components are optional:

- One or more additional Supplementary Services Gateways More than one instance can communicate with the same SIP Server, but each Supplementary Services Gateway instance must have a unique Resource DN.
- Multiple VP Resource Managers For high availability in active standby and active active HA modes.
- Multiple VP Reporting Servers For high availability in Active Standby.
- One or more additional VP Media Control Platforms with VP Fetching Module and VP Squid Depends on sizing.
- One or more VP Call Control Platforms with VP Fetching Module and VP Squid Depends on sizing.
- SNMP Master Agent See Voice Platform Solution and Dependencies.
- CTI Connector See CTI Connector and How the CTI Connector Works.
- PSTN Connector Optional only for customer who do not use traditional TDM technology in their environment, otherwise, it is required. See PSTN Connector and How the PSTN Connector Works.
- Policy Server Optional, but recommended for enterprise environments that include multi-tenant hierarchies. See How the Policy Server Works.
- MRCP Proxy Optional, but recommended in environments where MRCPv1 ASR/TTS usage reporting is required. See How the MRCP Proxy Works.

Options to Deploying VP Reporting Server

Genesys recommends that you deploy at least one VP Reporting Server per deployment. When VP Reporting Server is installed, GVP Reporting data can be viewed on the Monitoring tab in the Genesys Administrator GUI. VP Reporting Server also provides an API, which allows GVP reporting data to be used with third party reporting products.

If you do not require GVP historical reporting in your deployment, you can deploy VP Reporting Server without a Reporting Server database. This deployment option retains support for the GVP dashboard reports. If you do not require the historical or dashboard reports, installation of the VP Reporting Server is not required.

Startup Sequence for the VPS

The table below describes the recommended startup sequence that is used to start the VPS successfully at initial startup, or if any component in the solution is stopped and must be restarted. It includes only those components that are listed in VPS Components Minimum Deployment on page 198.

See the following procedures describe ways to stop and start GVP Application and Solution objects:

- Procedure: Starting and Stopping GVP Solution Objects
- Procedure: Starting and Stopping GVP Application Objects
- Procedure: Configuring Application Objects to Start Automatically

Table: Startup Sequence VPS (Minimum Deployment)

Requirement	VPS component
Components that must be operational before you start the GVP components	• Management Framework components • SIP Server • Database Server prerequisite for the Reporting Server, but optional.
GVP Components	• Reporting Server • Resource Manager • Media Control Platform • Call Control Platform

HMT Permissions and Access Rights

If you are deploying GVP in a multi-tenant environment, you must ensure that the service provider or GVP enterprise manager is the only user assigned to the super-users access group, and therefore, is solely responsible for managing DID Groups and defining tenants. In addition, to maintain numbering and naming uniqueness, which is a GVP requirement, tenants must not be assigned edit permissions for their own configurations. However, tenant users can be assigned read permissions, which enable them to read and modify their configurations and reports.

The tenant that is defined as the parent becomes the reference entry point in the tenant hierarchy. The parent tenant with read permissions can view their child tenants and their configurations and reports, but cannot view the child tenants below them (their grandchild tenants).

Preparing the Operating System for GVP

This topic describes how to prepare the Windows operating system for Genesys Voice Platform (GVP) 8.5.x deployments. For information about the software prerequisites when deploying GVP on the Windows platform, see Prerequisites.

Tip

- There are no requirements to prepare the Linux platform for GVP. This section contains information about the Windows platform only.
- GVP supports installation on virtual machines created by VMware ESXi5 software. All the same requirements for physical GVP host systems apply. See the VMWare ESXi5 manuals and the operating system vendor documentation for installing virtual machines.

Windows Services and Settings

Complete the tasks to configure Windows services and modify Registry settings on each host before you install GVP. See the Task Summary below.

Important

When you name a computer, do not use the underscore (_) character, even though Windows setup permits it. Using the underscore character causes serious problems with several web services used by the GVP software.

Task Summary: Specifying Windows Services and Settings

Objective	Related Procedures and Actions
Note: Some of the services shown here may no longer be available on Windows 2008/ 2012.	

Modify Windows Registry settings.	**1. On Windows 2008:** Accommodate environments that handle a large number of concurrent calls or enable quick usage of MRCPv2 ASR & TTS resources by changing the following Windows Registry parameter on all GVP hosts before you begin the deployment. If you have planned your environment to reach call volumes of more than 200 concurrent calls, use the regedit.exe command to add the `DWORD` parameter to the following registry key: `HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/tcpip/Parameters` a. Select **Edit > New**. b. Click the DWORD value, enter TcpTimedWaitDelay with decimal value: 30 (or hex value: 1e) c. Exit the Registry and reboot the computer. The minimum value for this parameter is 30 seconds. `DWORD` resets any value less than 30 to the default of 240 seconds.
Modify Windows OS settings.	**2. On Windows 2008 only:** Set the IP DiffServ bits on outgoing packets by defining the QoS Policy in the QoS Packet Scheduler, which is included in the OS. For instruction about how to define the IP DiffServ bits on outgoing packets per executable or per port, see the article, *Creating and Editing the QoS Policy in the Tech Center Library* on the Microsoft website. • Changing the QoS policy is optional; do not do it casually.
	3. On Windows 2008 only: • If you are installing the PSTN Connector and Dialogic, disable Physical Address Extensions (PAE) by executing the following commands in the CLC: `C:\bcdedit /set nx OptOut`

	`C:\bcdedit /set pae ForceDisable` Then, restart the server.
Enable or disable the required services, and set service start modes.	Modify the following services as indicated: • Alerter: Disabled • Application Management: Manual • Com + Event System: Manual • Computer Browser: Disabled • Event Log: Automatic • Internet Information Server (IIS) Admin Service: Automatic[A] • Indexing Service: Disabled • License Logging: Disabled • Messenger: Disabled • Net Logon: Automatic • NT LAN Manager (LM) Security Support Provider: Manual • Plug and Play: Automatic • Protected Storage: Automatic • Remote Procedure Call (RPC): Automatic • Remote Procedure Call (RPC) Locator: Manual • Server: Automatic • System Event Notification: Automatic • Task Scheduler: Automatic • TCP/IP NetBIOS Helper: Automatic • Telephony: Manual • Uninterruptible Power Supply (UPS): Manual • Workstation: Automatic • World Wide Web Publishing Service: Automatic[B]
Specify the recommended system performance settings.	See Procedure: Configuring Settings for System Performance below.

A: Not required for the default setup of MCP component. If you require inline grammars on the MCP, then consult Microsoft documentation on how to add the appropriate roles and features in Windows 2008/2012 to enable the Internet Information Services (IIS). Also, refer to the GVP 8.5 User's Guide or the Voice Platform Media Control Platform Configuration Options for how to configure inline grammars.
B: Inline grammars, when enabled, must be hosted on the local web server (MCP only). Please refer to the Microsoft documentation on how to add the appropriate roles and features in Windows 2008/2012 to enable the World Wide Web Publishing Service. Also,

refer to the GVP 8.5 User's Guide or the Voice Platform Media Control Platform Configuration Options for how to configure inline grammars.

Procedure: Configuring Settings for System Performance

Complete this procedure for each Windows server that will host GVP components, to maximize host performance.

1. Go to the **Control Panel >System> Advanced** tab.
2. In the Performance section, click **Settings**.
   The Performance Options page appears.
3. Select the **Advanced** tab.
4. In the Processor scheduling section, select **Background services**.
5. Set the virtual memory size:
   a. In the Virtual memory section, select **Change**.
      The Virtual Memory page appears.
   b. Select **Custom size** and then set the following:
      - Initial size (MB): 1.5 times your RAM
      - Maximum size (MB): 2 times your RAM
   c. Select **Set**.
6. Click **OK** to exit all dialog boxes.
7. When prompted, restart the computer.

Preparing the Hosts for GVP

In a solution environment that includes Management Framework, the Configuration Server propagates configuration information to the servers that are hosting Genesys components. To facilitate this, you must install the Genesys Local Control Agent (LCA) on each GVP host.

In addition, use Genesys Administrator to create each new host in the Configuration Database, so that the Configuration Server can detect their presence.

Configuring Hosts in the Configuration Database

This section contains the following procedures, which describe how to prepare the host(s) before the GVP components are installed:

- Procedure: Configuring a Host in Genesys Administrator
- Procedure: Installing the Local Control Agent (Windows)
- Procedure: Installing the Local Control Agent (Linux)

Procedure: Configuring a Host in Genesys Administrator

- Configuring a host enables it to communicate with the Configuration Server.
- Each new host is controlled and monitored by the LCA.

1. Verify that the Genesys Administrator web application is installed on the Management Framework host.
2. Obtain the Universal Resource Locator (URL) of Genesys Administrator.
3. In a web browser, type the Genesys Administrator URL. For example: `http://<Genesys Administrator host>/wcm/`
4. At the Login dialog box, enter the information as shown in the table below.

Table: Genesys Administrator Login

Field	Description
User Name	Enter the user name, typically default
Password	Enter the password, typically password.
Application	Enter the application name of the Configuration Server, typically default.
Host Name	Enter the host name of the Configuration Server for example, `ConfgS1`.
Port	Enter the port number of the Configuration Server, typically `2020`.

5. Click **OK**.
   The Genesys Administrator graphical user interface (GUI) appears.
6. On the Provisioning tab, click `Environment > Hosts> New`.
7. In the General section of the Configuration tab, enter the information that identifies the host, as shown in the table below.

Table 15: Host Properties Genesys Administrator

Field	Description
Name	Enter the host name of the GVP host for example, MCP1
IP Address	Enter the IP address of the GVP host.
OS Type	From the drop-down list, select the OS type.
OS Version	Enter the version number of the OS that is installed on the host.
LCA Port	The LCA port number 4999 is entered by default.
Solution Control Server	Browse to select the Solution Control Server (SCS).
Stat	Enter a checkmark in Enabled.

Tip

When you are entering the host name for Linux hosts, ensure that the host name that is created in the Configuration Database is identical to the host name of the Linux host (they are case-sensitive). If the host names do not match, the installation fails when the hostname command is executed.

8. Save the configuration.
9. Install the LCA on the host. See Procedure: Installing the Local Control Agent (Windows) or Procedure: Installing the Local Control Agent (Linux).

Procedure: Installing the Local Control Agent on a Windows host

- Installing the LCA on each GVP host enables it to be controlled and monitored by the Solution Control Server, and also installs the Genesys Deployment Agent (GDA).

1. Verify that he servers where you are installing GVP components must meet the GVP system requirements. See Prerequisites and Planning.
2. The fully qualified domain names (FQDN) of Genesys servers do not contain special characters, such as the underscore (_). To ensure that Genesys software works properly, FQDNs must contain only alpha numeric characters, such as letters A Z, a z, digits 0 9; the one exception allowed is hyphens (-).
3. Third-party software, especially antivirus software, must be stopped on the servers on which GVP components will be installed.
4. You have obtained the Genesys Management Framework CDs, or a network path and the location the LCA software. For a description of the directory structure of the installation CDs, see Management Framework 8.5 Deployment Guide.
5. On the host, navigate to the directory that contains the installation files for the Local Control Agent and then execute the setup.exe file.
6. At the prompt, enter the information that identifies the host, as shown in the table below.

Table: Configuration Server Properties Deployment Wizard

Field	Description
Name	Enter the host name of the Configuration Server for example, Config1.
Port	Enter the port number of the Configuration Server. The default is `2020`.
User	Enter a user name for the Configuration Server typically, default.
Password	Enter a password for the Configuration Server typically, password.

Tip

When you install the LCA on the Supplementary Services Gateway host, ensure that the name and port number of the Configuration Server are the same as those that were configured for the Reporting Server. This is necessary because the Reporting Server polls the Supplementary Services Gateways data through SNMP and is made aware of its existence through the Configuration Server.

7. Click **Next**.
8. Restart the host computer.

9. After the host is restarted, open Windows Services, and verify that the Local Control Agent and the Genesys Deployment Agent services are installed and running.
10. Complete the preinstallation activities. See Preparing to Install GVP on Windows.

Procedure: Installing the Local Control Agent on a Linux host

- Installing the LCA on each GVP host enables it to be controlled and monitored by the Solution Control Server, and also installs the Genesys Deployment Agent (GDA).

1. Verify these prerequisites:
   The servers where you are installing GVP components meet the GVP system requirements. See Prerequisites and Planning.
   - The fully qualified domain names (FQDNs) of Genesys servers do not contain special characters, such as the underscore (_). FQDNs can contain only standard characters, such as letters A Z, a z, digits 0 9, and hyphens (-).
   - Third-party software, especially antivirus software, is stopped on the servers on which GVP components will be installed.
   - You have obtained the Genesys Management Framework installation disks, or a network path and the location the LCA software.
2. At the Linux host, log in as root by typing `su`.
3. Log in as root and enter the path to the directory that contains the LCA installation package.
4. Run the sh install.sh command.
   The installation script is initiated.
5. At the prompt, enter the information that identifies the host, as shown in Table: Configuration Server Properties Deployment Wizard.

   > **Tip**
   >
   > When you install the LCA on the Supplementary Services Gateway host, ensure that the name and port number of the Configuration Server is the same as the one that was configured for the Reporting Server. This is necessary because the Reporting Server polls the Supplementary Services Gateways data through SNMP and is made aware of its existence through the Configuration Server.

6. At the prompt, enter the destination directory, for example:
   `/opt/genesys/lca`

Tip

Genesys recommends that you use the destination directory that is shown in the example.

7. Configure the GDA to start automatically when the server is restarted, for example: Enter `/etc/rcX.d /etc/rc.d/init.d/gctigda`.
8. Configure the LCA to start automatically when the server is restarted, for example Enter `/etc/rcX.d /etc/rc.d/init.d/gctilca start`. You can start the LCA and GDA manually, at the Linux prompt, with these commands: `/etc/init.d/gctilca start` and `/etc/init.d/gctigda start`
9. Complete the preinstallation activities. See Preparing to Install GVP on Linux.

Installing GVP

Task Summary: Preparing Your Environment for GVP

Objective	Related Procedures and Actions
Plan the Deployment	For specific restrictions and recommendations to consider, see Preparing the Hosts for GVP.
Install Common Genesys Framework Components	1. **[+] Install Management Framework** a. Deploy Genesys Management Framework and ensure that it is fully operational and running. See the Framework Deployment Guide. b. Management Framework is the centralized element management system for all Genesys software. 2. **[+] Install Genesys Administrator** Install Genesys Administrator and ensure that it is fully operational, using instructions in the Framework Deployment Guide. Genesys Administrator is the centralized management interface for all Genesys software. 3. **[+] Install Genesys SNMP Master Agent** a. Install and configure the SNMP Master Agent on the same host(s) as the Resource Manager, Media Control Platform, Call Control Platform, Supplementary Services Gateway, CTI Connector, and PSTN Connector components. b. After the SNMP Master Agent has been installed on the GVP hosts, you will assign the SNMP Master Agent to each component for which you want to capture alarm and trap information. This is a post-installation activity (see Procedure: Creating a Connection to a Server).

	c. The Genesys Media Server 8.5 DVD includes an MIB Installation Package that can be loaded on the SNMP management console (for example, HP Open View) in your environment. To install the MIBs, run the setup.exe file and specify the default installation path: `C:\Program Files\GCTI\gvp\VP MIB 8.5` **Tip** The SNMP Master Agent is required only if you are capturing alarm and trap information. For more information about installing the Management Framework and the SNMP Master Agent, see the Framework Deployment Guide. For more information about the MIBs, see the GVP 8.5 SNMP MIB Reference.
Install third party software	4. If you are using automatic speech recognition (ASR) and/or text-to-speech (TTS), install the third-party Media Resource Control Protocol (MRCP) speech server and ensure that it is operational. For more information about this software, see your MRCP vendor's documentation. For information about prerequisite software, see Prerequisites.
Prepare the Host(s)	5. Stop antivirus software that might be running on systems that will host GVP components. Check the vendor documentation for your antivirus software configuration. 6. Install the Local Control Agent on the GVP host(s). See Procedure: Installing the Local Control Agent (Windows) or Procedure: Installing the Local Control Agent (Linux).
Complete the prerequisites	7. **[+] Install the Squid caching proxy (optional)**

	Squid is optional —no longer a prerequisite for the Fetching Module. • Windows: Install the Squid caching proxy on the Media and Call Control Platform hosts. Linux: Squid is installed with the OS. • Windows: Check Windows Services to confirm that Squid is installed. Linux: Enter the command `rpm -qa | grep squid`. 8. **[+] Install the Apache Web Server.** On the Media Control Platform, install the Apache Web Server (Linux only). 9. **[+] Install Red Hat Package Manager.** Obtain the latest Red Hat Package Manager (RPM) for the Apache Web Server from the vendor's website and follow the instructions for installing it. a. Configure the Apache Web Server to start automatically at startup. b. Modify the `/etc/rc.d/rcX.d` file. c. Enter `/etc/init.d/httpd start`. For more information about the prerequisite software, see Prerequisites or visit the vendor's website. 10. **[+] Install Dialogic** To ensure that Dialogic functions properly after installation, disable the Physical Address Extension (PAE) on Windows 2008. To do so from the command line interface (CLI), enter these commands: • `C:\bcdedit /set nx OptOut` • `C:\bcdedit /set pae ForceDisable` (only if you are adding the PSTN Connector to your environment)

	11. Restart the server. For more information about installing and configuring Dialogic hardware and software, visit the vendor's website.

Task Summary: Deploying GVP with Genesys Administrator

Objective	Related Procedures and Actions
Configure the host(s)	1. Configure a new host in the Configuration Database for each computer that is hosting GVP components. See Procedure: Configuring a Host in Genesys Administrator.
Install GVP	2. Import the Installation Packages into the Genesys Administrator Repository. See Procedure: Importing the Installation Packages into the Repository. 3. Use the Genesys Deployment Wizard to install the GVP components with basic configuration. See Procedure: Using the Deployment Wizard to Install GVP **Tip** If the components are deployed on the same server (all-in-one deployment), be aware of port conflicts. To avoid port conflicts use the Genesys Deployment Wizard for all-in-one deployments.
Start the components	4. Configure the GVP components to start automatically. See Startup Sequence for the VPS and Starting and Stopping the Components.
Complete the post-installation activities	5. Configure the GVP components for the functionality that you want use in your deployment. See Task Summary: Post-Installation Configuration of GVP.

	Tip Before you begin to plan and configure your GVP resources, there is important information you should know about tenant permissions and assigning DID Groups in multi-tenant environments. See HMT Permissions and Access Rights.

Installing GVP with the Deployment Wizard

The Genesys Administrator wizard simplifies the GVP deployment by prompting you for the information that is required to install each component. The Import Wizard enables you to import the GVP installation packages into a repository. You can use the built-in Genesys Administrator Repository, or you can **[+] create your own repository.**

Procedure: Creating an IP Repository

To create an Installation Package Repository:

1. Create a shared folder (with read and write permissions) on the network server on which IPs will be stored.
2. Go to Deployment > Repository Management > Repositories IP List.
3. Click **New**.
4. In the Create Repository dialog box:
    a. Type the network path of the shared folder and the name of the new Repository. The path must be in the following format:
    `\\serverName\path\sharedFolder\`*`Repository_name`*
    ...and *`Repository_name`* is required onlt if you want to give the Repository a working name different than the file name.
    b. (Optional) Enter a brief description.
    c. Click **Create**.

The new repository folder is created in the shared folder, and appears in the list of Installation Package Repositories as either its filename, or if specified, the name that you entered in the final step above.

Then, use the Deployment Wizard to install the GVP components individually with a basic configuration, or to install multiple instances of the same component on different hosts.

Tip

If you are installing Genesys Administrator and the Media Control Platform on the same host, you must install GVP by using the manual procedures and ensure that Genesys Administrator is shut down during the installation. For the procedures to install GVP manually, see Installing GVP Manually on Windows and Installing GVP Manually on Linux.

The GVP installation CDs contain the installation packages for both Windows and Linux operating systems. See GVP Installation DVDs.

Use the following procedures to install GVP on the targeted host(s):

1. Procedure: Importing the Installation Packages into the Repository (using GA)
2. GAX Procedures to Upload, Copy, Deploy, Download, or Delete Installation Packages.
3. Procedure: Using the Deployment Wizard to Install GVP

Procedure: Importing the Installation Packages into the Repository

Do this before starting the Deployment Wizard to install the GVP components.

1. Verift that the installation packages are copied to, and accessible from, the server on which Genesys Administrator is installed or a network drive for which the appropriate permissions are set so that a remote user can access it. Ensure that the folder that contains the installation packages is shared.
2. Log in to Genesys Administrator.
3. On the Deployment tab, highlight the Repository into which you want the Installation Packages to be imported.
4. Click `Repository > Installation Packages > Import`.
   The Installation Packages Import Wizard appears.

5. Continue through the Deployment Wizard by performing either of these methods:
   [+] Installation DVD
   a. Select Installation DVD as the import source and click **Next**:
   b. In the DVD Source field, enter the Universal Naming Convention (UNC) path to the directory that contains the CDInfo.xml file, (for example, \\127.0.0.1\GVPCDShare\G254_8131001_ENU\) then click Next. The Installation Packages Repository appears in the Select Items page.
   c. Select the installation package that you want to import, then click **Next**. As the wizard begins to import the package, the Process import page appears, displaying a progress bar.
   d. After the installation package is imported, click Finish.
   e. Repeat Steps ii and iii to import additional installation packages.

   [+] Single Installation Package
   a. Choose Single Installation Package as the import source and click **Next**.
   b. In the IP Source field, enter the UNC path to the folder that contains the `ip_description.xml` file. For example: `\\127.0.0.1\GVPCDShare\G254_8131001_ENU\`
   c. In the Template Folder field, enter the path to the Templates folder on the DVD or a folder in a network directory, then click Next.

      > Tip
      >
      > When you enter the path to a network directory, do not include a mapped drive letter with a dollar sign ($). Enter the path, without the drive letter. For example: `\\Installation_Pkgs\gvp81\`.

      The Installation Packages Repository appears on the Select Items page.

   d. Select the IP template for the IP and version that you want to install, then click Next. As the wizard begins to import the package, the Process import page appears, displaying a progress bar.
   e. After the installation package is imported, click Finish.
   f. Repeat Steps iii and iv to import additional installation packages.
   g. Install the GVP components as described in Procedure: Using the Deployment Wizard to Install GVP.

Procedure: Using the Deployment Wizard to Install GVP

The Application objects are created automatically when you use the Genesys Deployment Wizard; therefore, you do not need to import the templates or create the Application objects manually. Use the Genesys Deployment Wizard for both Windows and Linux installation packages.

1. Verify that:
    - The appropriate prerequisite software is installed on each component, as required. See Prerequisites.
    - A new host is added in the Configuration Database for each GVP host. See Procedure: Configuring a Host in Genesys Administrator.
    - The LCA is installed on the GVP host(s). See Procedure: Installing the Local Control Agent (Windows).
    - The Installation Packages are imported to the Installation Packages Repository. See Procedure: Importing the Installation Packages into the Repository.
2. Log in to Genesys Administrator.
3. On the Deployment tab, click **Repository > Installation Packages**.
4. Highlight the installation package that you want to install (ensuring that the IP that you select matches the version of the OS on the host).
5. On the right side of the Installation Packages page, click the left-arrow button to view the Tasks pane.
6. In the Tasks panel, click the **Install Package** link.
   The wizard Single Installation Package Deployment Wizard page appears.
7. To enter the information about the target host, click **Next**:
    - **Application Name**: Enter a name to identify this instance of the application that is to be deployed.
    - **Target Host**: Select the host on which you want to install the Application object.

    Tip

    When you are installing multiple applications of the same type, use unique names in order to identify them easily.

8. To configure the parameters for the component, click **Next** and then, for each GVP component:
    - To specify the Working Directory, accept the default path to the directory where the installation package resides.

- To elect Linux 32-bit operating systems in Linux installations, select 32 from the drop-down menu in the field **DataModel**.
- **[+] Configure these additional Media Control Platform Parameters**
    - **Install Mode**: From the drop-down menu, select the audio files format:
        - Select MuLaw format in North America.
        - Select ALaw format in Europe.
    - **HTTP Proxy Usage**: From the drop-down menu, choose `Selected` or `Not Selected`.
    - Create a connection to the Reporting Server in each of the Media Control Platform, Call Control Platform, Resource Manager, and Supplementary Services Gateway Applications. See Procedure: Creating a Connection to a Server.
- **[+] Configure these additional Reporting Server Parameters**
    - **JavaHome Path**: Enter the path to the directory in which the Java binary files will reside for example: `<Installation directory>\Java\jre7\bin`
    - **DBMS engine**: Select the version of the DBMS engines that you want to install:
        - **MS SQL Server 2005 or MS SQL Server 2008 Standard Edition**
        - **MS SQL Server 2008 Enterprise Edition**
        - **Oracle 10g/11g Standard Edition**
        - **Oracle 10g/11g Enterprise Edition**
        - **No Database** allows Reporting Server to operate without a database).

> Tip
>
> DB Partitioning is supported when the Enterprise edition of the OS is selected only.

    - **DBMS host**: Enter the host name of the DBMS engine.
    - **DBMS port**: Enter the port number of the DBMS engine.
    - **Database name**: Enter the name of the database server that will be used by the Reporting Server for example, db_rs.
    - **DBMS user**: Enter a user name for the DBMS.
    - **DBMS password**: Enter a password for the DBMS.
    - **Reporting Server Port**: This field is populated automatically with 61616.

- **Web Server Port**: Retain the default port number 8080.

Note: To support SCAN addresses used in Oracle, set these Reporting Server options:
```
[persistence] hibernate.remote.database

[persistence] hibernate.remote.url =
jdbc:oracle:thin:@<SCAN address>:<port>/<service name>
```
The first of these means to set the `hibernate.remote.data` option to blank, where as for the second option you should replace the `<SCAN address>` with the FQDN of the scan address you've set up for Oracle RAC, `<port>` is the port number to access Oracle, and `<service name>` is the name of the Oracle service that has been set up to be used by the Reporting Server.

9. Enter a unique connection port (default is 5000). Genesys Administrator, through Solution Control Server uses this listening port to monitor, start, and stop this application after installation.
10. To view the Deployment Summary, click **Next**.
11. To start the Deployment, click **Next**.
A progress bar appears at the top of the Deployment page.
12. To view the Results page, click **Next**.
13. To exit the wizard, click **Finish**.
14. Configure the Application objects to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
15. Complete the post-installation activities for the GVP components. See Task Summary: Post-Installation Configuration of GVP.

Installing the VP Reporting Plugin for GAX

Generating Reports: Using GA vs. Using GAX

Genesys Administrator Extension (GAX) can now generate all reports that are available in Genesys Administrator (GA), and some new reports that GA does not offer. Configuration of GVP 8.1.7 and above, as well as the ability to generate most reports, remains in Genesys Administrator (GA).

Below is a breakdown of reports that you can generate with GAX vs. with GA.

Table: Where can I generate this report?

Exclusive to GAX	Generating these new reports: • VoiceXML Call Arrivals, Call Peaks, Call Durations. • Media Service Call Arrivals, Call Peaks, Call Durations. • ASR/TTS Call Durations.
Exclusive to GA	Configuring GVP.
Common to both GAX and GA	• Generating Call Browser Reports: Historical Call Status, In-Progress Call Status. • Generating VAR reports: VAR Call Completion, VAR IVR Action Usage, VAR Last IVR Action. • Generating these reports: ◦ Real-time Call Browser, IVR Profile Call Arrivals, IVR Profile Call Peaks, Tenant Call Arrivals, Tenant Call Peaks. ◦ Component Call Arrivals (RM, MCP, CCP, PSTNC, CTIC, ASR, TTS). ◦ Component Call Peaks (RM, MCP, CCP, PSTNC, CTIC, ASR, TTS). ◦ Call Dashboard, SSG Dashboard, Fetch Dashboard, PSTNC, CTIC Dashboard. • Generating Service Quality Reports: CallFailures, Call Summary, Latency Details, Latency Dashboard.

Report Types

A plugin for Genesys Administrator Extension (GAX) provides access to reports on GVP activities.

Table: VP Reporting Plugin for GAX Report Types

Service Quality Report	Generating Service Quality Reports: CallFailures, Call Summary, Latency Details.
Call Detail Record Browsing	Query and inspect records for calls processed by different GVP components. Real-time reporting and historical reporting supported.

Dashboard	Monitor in-progress calls, from the perspective of IVR Profiles or GVP components, in real time.
Operational Reporting	Generate reports on the rate of call arrivals or peak call volume, by IVR Profile or by GVP component.
Voice Application Reporting	Generate reports on the logical success and failure rates for calls and IVR Actions in a given IVR Profile.

Tip

VAR reporting data is available only for applications that leverage the VAR <log> interfaces Call Result, Action Start, Action End, and Custom Name/Value Pair.

Installing the VP Reporting Plugin for GAX

Procedure: Installing the VP Reporting Plugin for GAX from an Executable

- This installation enables Genesys Administrator Extension (GAX) to generate reports of GVP activities.
- The VP Reporting Plugin is an add-on component to an existing GAX installation. It is enabled automatically when the plugin files are installed into existing GAX directories.
- The VP Reporting Plugin for GAX supports, and is supported by, Reporting Sever 8.1.6 or later. It will not work with earlier versions.

1. Verify that the target GAX installation is version 8.1.3 MR1, or later.
2. Be prepared for these information requests and choices:
    - You will need the full path to your Tomcat installation.
    - You will either confirm the default installation directory, or enter a new one.
    - If the target installation directory is populated, you will choose an action:
    - Back up all files in the directory.
    - Overwrite only the files contained in this package.
    - Wipe the directory clean.
3. Stop Tomcat on the host running GAX.
4. Run the installation executable.
    - For Windows, this file is `<IP plugin directory>/setup.exe`.

   - For Linux, this file is `<IP plugin directory>/install.sh`.
5. Perform the installation steps, using the information that you gathered for the prerequisites.
6. Start Tomcat on the host running GAX.

- To use the plugin, open GAX and click **Voice Platform Reporting** from the Reports menu. See the procedure Generating a Report Using GAX in the GVP 8.5 User's Guide.
- See the plugin's online help to read about generating and understanding reports, and for help with selecting filters—an important aspect of generating a report).

Procedure: Installing the VP Reporting Plugin for GAX from Inside GAX

- This installation enables Genesys Administrator Extension (GAX) to generate report of GVP activities.
- The VP Reporting Plugin for GAX is an add-on component to an existing GAX installation. It will be enabled automatically when the plugin files are installed into existing GAX directories.
- The VP Reporting Plugin for GAX supports, and is supported by, Reporting Sever 8.1.6 or later. It will not work with earlier versions.

1. Be prepared to enter the directory path to an installation directory, or to a zipped file.
2. Verify that the destination GAX installtion is version 8.1.3 MR1 or later.
3. Select **Installation Packages** from the Configuration menu.
4. Click the plus icon (+) at the upper right of the Installation Packages window. The Software Installation Wizard dialog appears to the right of the current window, offering these **Import Type Selection** choices as radio buttons:
   - Installation Package Upload (includes templates)
   - Installation Package Upload (template uploaded separately)
   - UNC Path to Mounted CD or Directory
   - UNC Path to an Existing Administrator Repository
   - UNC Path to Zipped IPs from Support
5. Select the radio button that matches your installation source and click the **Next** button.
6. The next dialog will request input that corresponds to your choice in the previous step:
   - **Installation Package Upload (includes templates)** requires you to choose a zipped IP file.
   - **Installation Package Upload (template uploaded separately)** requires you to choose a zipped IP file, an XML template or an APD template.
   - Each of the three choices that begin with **UNC Path** requires a directory path that you may type or paste into the entry field.

You may see a request to correct an error; type or paste your correction. When GAX is ready to install, the **Finish** button will be enabled.

7. Click the **Finish** button and wait for the upload to complete. When you see the message, Import has started. You may now close this wizard, you can close the Software Installation Wizard dialog by clicking the **Close** button at the bottom right or the **X** icon at the top right. The Reporting Plugin is ready to install.
8. Select the item that you imported from the Installation Packages window. A dialog with that title (in this case: VP Reporting Plugin for GAX) appears to the right.
9. The VP Reporting Plugin for GAX dialog offers these actions:
    - **Download** Downloads the installation package to your computer.
    - **Delete** Erases the IP.
    - **Copy to Tenants** Copies the IP to the tenant(s) that you specify. You'll select a tenant and click Finish.
    - **Deploy Profile: install** Displays the IP Deployment Wizard start dialog. All following steps in this procedure are the result of this choice.
10. Click **Next** to display a list of host computers for possible installation.
11. Select one or more hosts for installation using the check box to the left of each host name, and click **Next**.
12. At the Application Parameters dialog, complete these fields:
    - Application name for host
    - Tenant Name
    - App port
    - Primary Configuration Server
    - Backup Configuration Server
    - Skip IP Re-install
13. Click **Next** when you have completed all mandatory fields (marked with a red *.) For tool tip help, click the Information (i) icon to the right of each field title.
14. At the `Silent.ini` Parameters dialog, complete the **IPCommon : InstallPath** field. The default answer offered is `C:\genesys\GCTI\`.
15. At the Deployment dialog, verify that the answers you gave are all correct. If they are correct, click **Finish** and wait for the installation to complete.

Privileges: GVP Reporting Plugin for GAX

Unrestricted use of the plugin requires the privileges listed in below. You may need to request them from your system administrator.

Table: VP Reporting Plugin for GAX Privileges

Privilege Name	Label	Purpose
GVP_RPT_SITES	GVP Sites Report Access	Access GVP sites.

GVP_RPT_CALL_BROWSER	Call Browser Report Access	Access and generate GVP Call Browser reports.
GVP_RPT_DASHBOARD	Dashboard Report Access	Access and generate GVP Dashboard reports.
GVP_RPT_OPERATIONAL_REPORT	Operational Report Access	Access and generate GVP Operational reports.
GVP_RPT_SQ_REPORT	Service Quality (SQ) Report Access	Access and generate GVP Service Quality (SQ) reports.
GVP_RPT_VAR_REPORT	Voice Application Reporter (VAR) Report Access	Access and generate GVP Voice Application Reporter (VAR) reports.

Preparing to Install GVP Manually on Windows

Creating Application Objects in the Configuration Database This topic describes how to install Genesys Voice Platform (GVP) manually on the Windows operating system (OS) by using the executable files in the GVP Installation Packages. It contains the following sections:

- Task Summaries
- Preinstallation Activities
- Installing GVP (Windows)

Task Summaries

Task Summary: Preparing Your Environment for GVP (Windows)

Objective	Related Procedures and Actions
Prepare Your Environment	For specific restrictions and recommendations to consider, see Preparing the Hosts for GVP.
	Install Common Genesys Framework 1. Management Framework. ◦ Install the latest Installation Package (IP) for the Genesys Management Framework; and ensure that it is

	fully operational and running. See the Framework 8.5 Deployment Guide. Management Framework is the centralized element management system for all Genesys software. **Install common Genesys Framework components** 2. Genesys Administrator. ◦ Install Genesys Administrator, and ensure that it is fully operational. See the Framework 8.5 Deployment Guide. Genesys Administrator is the centralized management GUI for all Genesys software. 3. Genesys SNMP Master Agent. ◦ Install and configure the SNMP Master Agent on the same host(s) as the Resource Manager, Media Control Platform, Call Control Platform, Supplementary Services Gateway, CTI Connector, and PSTN Connector components. (After an SNMP Master Agent is installed for each GVP component on the respective GVP hosts, you will assign the SNMP Master Agent to each component for which you want to capture alarm and trap information. This is a post-installation activity (see Creating a Connection to a Server). The Genesys Media Server 8.5 DVD includes an MIB Installation Package that can be loaded on the SNMP management console (for example, HP OpenView, BMC Patrol, or IBM Tivoli) in your environment. To install the MIBs, run the setup.exe file, and select the default installation path: `C:\Program Files\GCTI\gvp\VP MIB 8.5` **Tip** The SNMP Master Agent is required if you are capturing alarm and trap information, and can be used to provide data for certain GVP reporting dashboards. For more information about installing the Management

	Framework and the SNMP Master Agent, see the Framework 8.5 Deployment Guide. For more information about the MIBs, see the GVP 8.5 SNMP MIB Reference. **Install third party software** 4. Third-party hardware and software. ◦ If you are using automatic speech recognition (ASR) and/or text-to-speech (TTS), install the third-party Media Resource Control Protocol (MRCP) speech server and ensure that it is operational. ◦ For more information about this software, see your MRCP vendor's documentation. For information about prerequisite software, see Prerequisites. **Prepare the host(s)** 5. Stop antivirus software that might be running on systems that will host GVP components. Check the vendor documentation for your antivirus software configuration. 6. Install the Local Control Agent on the GVP hosts so that they are controlled and monitored by the Solution Control Server (SCS). See Procedure: Installing the Local Control Agent (Windows). **Complete the prerequisites** 7. Prepare the Windows platform for GVP: a. Install Microsoft Internet Information Services (IIS) on the Windows hosts. See Prerequisites for supported versions of Microsoft IIS. b. Configure the required Windows services and settings on the systems that will host GVP components. See Windows Services and Settings.

	Tip In GVP 8.1.3 and earlier 8.x releases, IIS was required to host inline and universal hotkey grammar files that were fetched by ASR. In GVP 8.1.4, IIS is no longer required. The Media Control Platform now transmits these grammars by default in the MRCP requests. 8. On Reporting Server and Policy Server hosts, install the Sun Java Runtime Environment (JRE). Genesys recommends JRE 7.0; versions as early as JRE 6.0 update 19 or later are compatible, but no longer updated by the publisher with security patches. For more information about the prerequisite software, see Prerequisites or visit the vendor's website. 9. If you are adding the PSTN Connector to your environment, install Dialogic. To ensure Dialogic functions properly after installation, you must disable the Physical Address Extension (PAE) on Windows 2008. From the command-line interface (CLI), enter: ◦ `C:\bcdedit /set nx OptOut` ◦ `C:\bcdedit /set pae ForceDisable` and then restart the server. For more information about how to install and configure Dialogic hardware and software, visit the vendor's website.

The table below, **Task Summary: Deploying GVP Manually (Windows)**, contains a list of tasks required to deploy GVP manually and includes links to detailed information required to complete these tasks.

Task Summary: Deploying GVP Manually (Windows)

Objective	Related Procedures and Actions

Configure the host(s)	1. Configure a new host in the Configuration Database for each computer that is hosting GVP components. See Procedure: Configuring a Host in Genesys Administrator.
Create the Application objects	2. Create the Application objects: a. Import the templates. See Procedure: Importing Application Object Templates Manually. b. Create the Application objects. See Procedure: Creating Application Objects Manually.
Install GVP	3. Install the GVP components using the procedures listed below, each on a separate tab in this topic: Manually Installing GVP on Windows: a. Installing the Media Control Platform (on Windows). b. Installing the Call Control Platform (Windows). c. Installing the Resource Manager (Windows). d. Installing the Reporting Server (Windows). e. Installing the Supplementary Services Gateway (Windows). f. Installing the CTI Connector (Windows). g. Installing the PSTN Connector (Windows). h. Installing the Policy Server (Windows). i. Installing the MRCP Proxy (Windows).
Start the components	4. Start the components manually (or configure the components to start automatically). See Startup Sequence for the VPS and Starting and Stopping the Components.
Complete the post-installation activities	5. Configure the GVP components for the functionality you want use in your deployment. See Task Summary: Post-Installation Configuration of GVP.

Preinstallation Activities

Before you begin the preinstallation activities, ensure that the Local Control Agent (LCA) is installed on each GVP host and that the hosts are configured in the Configuration Database. See Preparing the Hosts for GVP.

To install the Genesys Voice Platform components, create an Application object in the Configuration Database for each application you are installing.

Each object that is created in the Configuration Database requires an object template. The templates are imported from the GVP installation DVDs or from a shared network directory.

After a template is imported, it can be used for subsequent instances of the same component. For example, if you are installing more than one Media Control Platform host, you can use the same template for each Media Control Platform Application object.

Tip

As a best practice, when you are using these manual procedures, import all of the Application and Speech Resource object templates that you require before you begin to deploy the components.

See Table: GVP Component Templates and Metadata and Table: Speech Resource Templates and Metadata for the names and locations of the templates on the installation DVDs.

Creating Application Objects in the Configuration Database

This section describes how to create Application objects in the Configuration Database either by using a wizard in Genesys Administrator or by using a manual procedure. To create Application objects manually, you must first import an Application object template, and then use it to create Application objects. This section contains the following:

- Procedure: Using the Create New Application Wizard
- Procedure: Importing Application Object Templates Manually
- Procedure: Creating Application Objects Manually

Procedure: Using the Create New Application Wizard

This procedure creates Application objects in the Configuration Database for each GVP component.

The Create New Application Wizard in Genesys Administrator imports the Application object templates and creates the Application objects for you. If you use the Genesys Deployment Wizard to install GVP, you can omit this procedure, because the wizard imports the GVP component Application object templates and creates the Application objects for you.

1. Verify that the GVP Installation Packages you need are accessible from the DVD or from a shared network directory.
2. Log in to Genesys Administrator.

3. On the Provisioning tab, click **Environment > Applications**.
4. In the Task pane, select **Create Application**.
   The Create New Application Wizard appears.
5. Click **Browse for File** to import a template.
   If the templates were previously imported, you can use an existing template by selecting **Browse for Template**.
6. Click **Add** to navigate to the directory that contains the template (.apd) files.
7. Click **Next** to specify the metadata.
8. Click **Browse > Add** to import the metadata for the Application object your are creating.
9. Click **Next** to configure the application parameters.
10. In the Host field, click the Browse icon to select the host where you will install the application.
    In Genesys Administrator, the mandatory fields are marked with a red asterisk.
    In the wizard, all fields on the Application Parameters page are populated automatically, except the Host field.
11. After the host appears on the Application Parameters page, click **Create**.
    The Results page appears, to confirm the Application object is created.
12. Click **Finish**.

Procedure: Importing Application Object Templates Manually

Import the appropriate Application object templates to the Configuration Database manually before you install the Application object.

Use this procedure only if you are manually creating Application objects; otherwise, you can use the Genesys Administrator Create New Application Wizard. If you use the Genesys Deployment Wizard to install GVP, you can omit this procedure, because the wizard imports the GVP component Application object templates and creates the Application objects for you.

1. Verify the GVP hosts are prepared for deployment. See Preparing the Hosts for GVP.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, click **Environment > Application Templates**.
4. In the Tasks pane, click **Import Application Template**.
5. In the dialog box that appears, click **Add**.
6. In the Choose File dialog box, navigate to the directory that contains the GVP or Speech Resource Application object templates.
   The table below lists the file names and locations of the GVP Application object templates.

Table: GVP Component Templates and Metadata

Application object	File name
Location: <Genesys Solutions Dir>\Templates\<file_name>	
Resource Manager	VP_ResourceManager_85x.apd VP_ResourceManager_85x.xml
Media Control Platform	VP_MediaControlPlatform_85x.apd VP_MediaControlPlatform_85x.xml
Call Control Platform	VP_CallControlPlatform_85x.apd VP_CallControlPlatform_85x.xml
Reporting Server	VP_ReportingServer_85x.apd VP_ReportingServer_85x.xml
Supplementary Services Gateway	VP_SupplemetaryServicesGateway_85x.apd VP_SupplemetaryServicesGateway_85x.xml
CTI Connector	VP_CTIConnector_85x.apd VP_CTIConnector_85x.xml
PSTN Connector	VP_PSTNConnector_85x.apd VP_PSTNConnector_85x.xml
Policy Server	VP_PolicyServer_85x.apd VP_PolicyServer_85x.xml
MRCP Proxy	VP_MRCPProxy_85x.apd

	VP_MRCPProxy_85x.xml
GVP Reporting Plugin for GAX	VP_ReportingPlugin_GAX_MF810_850.apd VP_ReportingPlugin_GAX_MF810_850.xml
Call RecordingServer	VP_CallRecordingServer_850.apd VP_CallRecordingServer_850.xml

The table below lists the file names and locations of the Speech Resource Application objects.

Table: Speech Resource Templates and Metadata

Speech Resource Object	File name
Location: <Genesys Solutions Dir>\Templates\<file_name>	
MRCPv1 ASR	VP_MCP_MRCPv1_ASR_85x.apd VP_MCP_MRCPv1_ASR_85x.xml VP_MCP_MRCPv1_ASR_IBM_85x.apd VP_MCP_MRCPv1_ASR_IBM_85x.xml VP_MCP_MRCPv1_ASR_NUANCE_85x.apd VP_MCP_MRCPv1_ASR_NUANCE_85x.xml VP_MCP_MRCPv1_ASR_TELISMA_85x.apd VP_MCP_MRCPv1_ASR_TELISMA_85x.xml
MRCPv1 TTS	VP_MCP_MRCPv1_TTS_85x.apd VP_MCP_MRCPv1_TTS_85x.xml VP_MCP_MRCPv1_TTS_IBM_85x.apd VP_MCP_MRCPv1_TTS_IBM_85x.xml VP_MCP_MRCPv1_TTS_NUANCE_85x.apd VP_MCP_MRCPv1_TTS_NUANCE_85x.xml

MRCPv2 ASR	VP_MCP_MRCPv2_ASR_85x.apd VP_MCP_MRCPv2_ASR_85x.xml VP_MCP_MRCPv2_ASR_NUANCE_85x.apd VP_MCP_MRCPv2_ASR_NUANCE_85x.xml
MRCPv2 TTS	VP_MCP_MRCPv2_TTS_85x.apd VP_MCP_MRCPv2_TTS_85x.xml VP_MCP_MRCPv2_TTS_NUANCE_85x.apd VP_MCP_MRCPv2_TTS_NUANCE_85x.xml

6. Double-click <template_filename>.apd,
   where <template_filename> is the file name of the template that you want to import. The template is imported, and the Configuration tab appears.

   > Tip
   >
   > Some of the Speech Resource Application object templates are vendor-specific. Ensure that you are using the correct template, based on the vendor. See Table: Speech Resource Templates and Metadata.

7. Click **Import Metadata**.
8. In the Waiting dialog box, click **Add**.
9. In the Choose File dialog box, navigate to the directory that contains the Application object templates.
10. Double-click <template_file_name>.xml,
    where <template_file_name> is the name of the file that contains the metadata. The metadata for the template is imported and the Configuration tab appears.
11. In the General section, enter the information that identifies the template, as shown in the table below.

Table: Application Template Properties

Field	Description
Name	Enter a descriptive name for the template. For example, GVP_FM_template.

Type	From the drop-down list, select the template type: ◦ For the GVP Application objects select the template with the same name. For example, for the Resource Manager, select **GVP Resource Manager**. ◦ For all Media Resource Control Protocol (MRCP) Client objects and Recording Servers, select **Resource Access Point**.
Version	Enter the template version number. For example, 8.5 or select it from the drop-down list.
State enabled	Insert a check mark in the checkbox to indicate Enabled.

Tip

For each GVP component or MRCP speech resource you want to install, add a GVP or Speech Resource Application object template before you begin the installation.

12. Click **Save**.
13. Create the required Application objects in the Configuration Database. See Procedure: Creating Application Objects Manually.

Procedure: Creating Application Objects Manually

Use this procedure...

- ...to create an Application or Speech Resource object manually in the Configuration Database for the application or speech resource that you are installing.
- ...only if you are manually creating Application objects, otherwise you can use the Genesys Administrator Create New Application Wizard.

If you use the Genesys Deployment Wizard to install GVP, you can omit this procedure, because the wizard imports the GVP component Application object templates and creates the Application objects for you.

1. Verify that an Application or Speech Resource object template is imported for the type of object that you are installing. See Procedure: Importing Application Object Templates Manually.

2. Log in to Genesys Administrator.
3. On the Provisioning tab, select **Environment > Applications > New**. The **Browse..\Application Templates** dialog box appears, displaying the contents of the Application Templates directory.
4. Click the object template for the GVP or Speech Resource Application object that you want to create. See Table: GVP Component Templates and Metadata and Table: Speech Resource Templates and Metadata for a list of template file names.
The Configuration tab appears, with some of the fields in the General section populated and disabled.
5. In the Name field, enter the name of the application.
6. In the State field, retain the default value: `Enabled`.
7. In the Server Info section, enter the information as shown in the table below.

> Tip
>
> The table below lists only the required fields that is, those fields that have an asterisk in front of the field name. The required fields must be populated before you can save the configuration.

Table: Application Object Properties

Field	Description
Host:	Enter the name of the computer that is hosting the application. For example, GVP-host1 or browse to select from a list of available hosts.
Working Directory:	Enter any value in these fields as temporary placeholders. For example, a backslash (\). These characters are replaced by the proper values when the component is installed.
Command Line	
StartUp Timeout	Enter the time interval, in seconds, during which the User Interaction Layer should expect this application to start. The default is `90` seconds. If the application is configured with the `Autostart` configuration option

	set to `true`, this is also the time that Solution Control Server waits to start this application after initialization or a system restart.
ShutDown Timeout	Enter the time interval, in seconds, during which the User Interaction Layer should expect this application to shut down. The default is `90` seconds.
Redundancy Type	From the drop-down list, select the type of redundancy in which you want this application to run.
Timeout	Enter the time interval, in seconds, that the client application should wait between reconnect attempts if the initial attempt to connect to the server does not succeed. The default is 10 seconds.
Attempts	Enter the number of times that the client applications should attempt to reconnect to this server before trying to connect to the backup server. The default value is 1. Important This value must be 1 or higher and it makes sense only if you specify a backup server for this server.
Auto Restart	From the drop-down list, select `true` (the recommended setting) or `false` (the default setting). ◦ Select `True` (recommended) to specify that the User Interaction Layer automatically restarts the application after it fails. ◦ Select `False` to disable the User Interaction Layer from automatically restarting the application after it fails.

Tip

Although the Configuration Database does not use the parameters in Table: Application Object Properties when Speech Resource Application objects are created, the required fields must be populated before you can save the configuration. If you are creating Speech Resource Application objects, retain the default values for the StartUp Timeout, Shutdown Timeout, Redundancy Type, Timeout, Attempts, and Auto Restart fields.

7. Click **Save**.
8. Install the GVP components. See Manually Installing GVP on Windows.

Manually Installing GVP on Windows

These instructions install each GVP component separately, one per tab.

Procedures

This section describes how to install GVP on a Windows OS in a new deployment, or add GVP components to an existing deployment. The prerequisites for each component include preparing the hosts for GVP and completing the preinstallation activities.

- Before you begin to install the components, copy the GVP installation packages to a directory on the Windows hosts or to a network drive from which they can be downloaded.
- To deploy multiple instances of the Media Control Platforms on a single host, follow the instructions in Deploying Multiple MCP Instances.

Squid Caching Proxy

Procedure: Installing the Squid Caching Proxy (Windows)

This procedure installs and starts the Squid caching proxy on the Media Control Platform and Call Control Platform hosts.

Installation is optional; do it only if you have specific needs such as a persistent cache or non-standard security.

1. Verify that the requirements for Windows software are met. See Table: Software Requirements—Windows.
2. On the Windows host, execute the setup.exe setup file:
   If you are using the GVP software DVD, browse to the `<GVPDVD>\gvp\windows\Squid\` folder,
   where `<GVPDVD>` is the DVD drive letter.
   If the DVD image is on a network drive, copy the `<DVDImage>\gvp\windows\Squid\` folder to the local computer.
3. When the Genesys Deployment Wizard appears, click **Next**.

> Tip
>
> To avoid having to make manual configuration changes after installation, install the Squid caching proxy in the default directory, `C:\Squid`.

4. After the Deployment Wizard is complete, click **Finish**.
5. Restart the host computer.

   Start the Service
6. At the host computer, from the Windows Start menu, click Programs **> Administrative Tools >** Services.
7. In the Component Services window, click **Services (Local)**.
8. In the Services list, ensure that the service is running.
9. If the service is not running, right-click **SquidNT**, and then select **Start**.
10. Click **OK**.
11. Install the Media Control Platform. See the tab **Media Control Platform** in this topic.

Media Control Platform

Procedure: Installing the Media Control Platform (Windows)

This procedure installs the Media Control Platform component, so that Session Initiation Protocol (SIP) applications which use Voice Extensible Markup Language (VoiceXML) can access the Media Control Platform media services.

- The Squid Caching proxy is installed on the Media Control Platform host, and the service is started (a prerequisite for GVP 8.1.1 and earlier 8.x releases only; optional in GVP 8.1.2 and later releases). See the tab **Squid Caching Proxy** in this topic.
- The Media Control Platform host is prepared for installation. See Preparing the Hosts for GVP.
- The Media Control Platform Application object template is imported, and an Application object is created. See Preinstallation Activities. Microsoft Internet Information Services (IIS) is installed. See Prerequisites.

1. Execute the setup.exe setup file:
    - If you are using the GVP software DVDs, browse to the `<GMS_Installation_DVD>\solution_specific\windows\mcp\` folder.
    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\mcp\` folder to the local computer.
2. When the Genesys Deployment Wizard appears, click **Next**.
3. Select one of two audio formats for your region:
    - `Mulaw` (North America)
    - `Alaw` (Europe).
4. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.
   These are the connection parameters for the Configuration Server.

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

5. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
6. On the Select Application page, select the Media Control Platform Application object that you want to install.
7. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.

8. Enter a check mark in one or both of the following check boxes, if required:
    - **Use HTTP Proxy** enables the use of an HTTP Proxy.
    - **Enable Voice XML application on this server** enables the use of GVP VoiceXML applications or Genesys Media Server with Play Application treatments.
9. Your action depends on your choice in the previous step (8):
    - Skip this step if you did *not* check the **Use HTTP Proxy** option.
    - If you checked the **Use HTTP Proxy** option:
        - In the Proxy Server Host Name field, enter the host name of the proxy server.
        - In the Proxy Server IP Address field, enter the IP address of the proxy server.
10. Create a connection to the Reporting Server. See Procedure: Creating a Connection to a Server.
11. On the Ready to Install page, click **Install**.
12. When the installation is complete, click **Finish**.
13. Configure the Media Control Platform Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
14. Install the Call Control Platform. See the tab **Call Control Platform** in this topic.

Call Control Platform

Procedure: Installing the Call Control Platform (Windows)

This procedure installs the Call Control Platform component so that applications using Call Control Extensible Markup Language (CCXML) can access the Call Control Platform call-processing services.

1. Verify that:
    - The Squid caching proxy is installed on the Call Control Platform host, and the service is started (a prerequisite for GVP 8.1.1 and earlier 8.x releases only; optional in GVP 8.1.2). See the tab **Squid Caching Proxy** in this topic.
    - The Call Control Platform host is prepared for installation. See Preparing the Hosts for GVP.
    - The Call Control Platform Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. Execute the **setup.exe** setup file:
    - If you are using the GVP software DVDs, browse to the `<GVP_Installation_DVD>\solution_specific\windows\ccp\` folder.

    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\ccp\` folder to the local computer.
3. When the Genesys Deployment Wizard appears, click **Next**.
4. On the Connection Parameters page, enter the information in the Host and User sections as shown in the table below.
   These are the connection parameters for the Configuration Server.

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

5. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
6. On the Select Application page, select the Call Control Platform Application object you want to install.
7. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.
8. In the VP Reporting Server section, enter the Host (name of the Reporting Server) and accept the default value `61616` for the Reporting Server port number.
9. On the Ready to Install page, click **Install**.
10. When the installation is complete, click **Finish**.
11. Configure the Call Control Platform Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
12. Install the Resource Manager. See the tab **Resource Manager** in this topic.

Resource Manager

Procedure: Installing the Resource Manager (Windows)

1. Verify that:

- The Resource Manager host is prepared for installation. See Preparing the Hosts for GVP.
- The Resource Manager Application object template is imported and an Application object is created. See Preinstallation Activities.

2. Execute the `setup.exe` setup file:
    - If you are using the GVP software DVDs, browse to the `<GMS_Installation_DVD>\solution_specific\windows\rm\` folder.
    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\rm\` folder to the local computer.
3. When the Genesys Deployment Wizard appears, click **Next**.
4. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

5. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
   These are the connection parameters for the Configuration Server.
6. On the Select Application page, select the Resource Manager Application object.
7. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.
8. In the VP Reporting Server section, enter the Host (name of the Reporting Server) and accept the default value `61616` for the Reporting Server port number.
9. On the Ready to Install page, click **Install**.
10. When the installation is complete, click **Finish**.
11. Configure the Resource Manager Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
12. Install the Reporting Server. See the tab **Reporting Server** in this topic.

Reporting Server

Procedure: Installing the Reporting Server (Windows)

- Microsoft SQL and Oracle are the only supported databases for Windows. In this procedure, when you select the database, you can choose the Standard or Enterprise edition of the database. If you select the Enterprise edition, partitioning of the database is enabled automatically during installation.
- When database partitioning is enabled, Genesys recommends that you not change the partitioning mode of operation or the number of partitions (even after the Reporting Server is started), because of issues that might arise if the database schema or stored data is changed.

1. Verify that:
    - The Sun Java Runtime Environment (JRE) 6.0, Update 19 or later is installed. See Prerequisites. JRE 7.0 or later is required for IPv6 communications.
    - The Reporting Server host is prepared for the installation. See Preparing the Hosts for GVP.
    - The Reporting Server Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. Execute the setup.exe setup file:
    - If you are using the GVP software DVDs, browse to the `<GMS_Installation_DVD>\solution_specific\windows\rs\` folder.
    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\rs\` folder to the local computer.
3. When the Genesys Deployment Wizard appears, click **Next**.
4. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.
   These are the connection parameters for the Configuration Server.

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

5. On the Select Application page, select the Reporting Server Application object.

6. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.
7. On the Select the Installed Sun's Java Runtime Environment (JRE) page, select the `runtime` environment for your deployment.
8. In the Database Engine Option section, select one of the following:
    - MS SQL Server 2005 or MS SQL Server 2008 Standard Edition<
    - MS SQL Server 2008 Enterprise Edition<
    - Oracle 10g/11g Standard Edition
    - Oracle 10g/11g Enterprise Edition
9. On the VP Reporting Server Parameters page, enter the parameters, as shown in the table below.
   In that table, the terms DB Server and database server refer to the server that hosts the database software. For example, Oracle or SQL Server not to the Management Framework Configuration DB Server.
   In addition, if you are installing an Oracle database, enter the SID or global database name in the Database Name field.

Table: VP Reporting Server Parameters

Section	Field	Description
Database Server	DB Server Host	Enter the host name or IP address, and the instance (if defined), on which the SQL Server or Oracle is installed.
	DB Server Port	Enter the port number of the database server host typically, 1433 for MSSQL and 1521 for Oracle.
Database	Database Name	Enter the name of the Reporting Server database. For example, db_rs.
User	User Name	Enter the user name that you want to use to connect to the database.
	Password	Enter the password that you want to use to connect to the database.

10. In the VP Reporting Server section, accept the default port number `61616`.
11. On the Ready to Install page, click **Install**.
12. When the installation is complete, click **Finish**.
13. Configure the Reporting Server Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically

Supplementary Services Gateway

Procedure: Installing the Supplementary Services Gateway (Windows)

1. Verify that the Supplementary Services Gateway host is prepared for installation. See Preparing the Hosts for GVP.
2. Verify that the Supplementary Services Gateway Application object template is imported, and an Application object is created. See Preinstallation Activities.
3. Execute the setup.exe setup file:
   - If you are using the GVP software DVDs, browse to the `<GVP_Installation_DVD>\solution_specific\windows\SSG\` folder.
   - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\SSG\` folder to the local computer.
4. When the Genesys Deployment Wizard appears, click **Next**.
5. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

6. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
7. On the Select Application page, select the Supplementary Services Gateway Application object.
8. Select the destination folder in one of two ways:
   - Click **Next** to accept the default directory
   - Click **Browse** to select the destination folder, and then click **Next**.
9. On the Ready to Install page, click **Install**.
10. When the installation is complete, click **Finish**.

11. Configure the Supplementary Services Gateway Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
12. If you intend to use the CTI Connector functionality in your environment, install the CTI Connector. See the tab **CTI Connector** in this topic.

CTI Connector

Procedure: Installing the CTI Connector (Windows)

Installation is optional. The CTI Connector acts as a SIP Back-to-Back User Agent (B2BUA) to provide a standard SIP interface to the internal GVP components. Furthermore, the CTI Connector communicates with CTI by using the Interactive Voice Response (IVR) Server XML interface to connect to the Genesys Framework.

1. Verify that the CTI Connector host is prepared for installation. See Preparing the Hosts for GVP.
2. Verify that the CTI Connector Application object template is imported, and an Application object is created. See Preinstallation Activities.
3. Execute the setup.exe setup file:
    - If you are using the GVP software DVDs, browse to the `<GVP_Installation_DVD>\solution_specific\windows\ctic\` folder.
    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\ctic\` folder to the local computer.
4. When the Genesys Deployment Wizard appears, click **Next**.
5. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

6. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
7. On the Select Application page, select the CTI Connector Application object.
8. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.
9. On the Ready to Install page, click **Install**.
10. When the installation is complete, click **Finish**.
11. Configure the CTI Connector Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically
12. If you intend to use the PSTN Connector in your environment, see the tab **PSTN Connector** in this topic.

PSTN Connector

Procedure: Installing the PSTN Connector (Windows)

Tip

Install and use the PSTN Connector (PSTNC) only after careful consideration, because the Dialogic boards that it requires are no longer sold.

Although Dialogic supports this hardware until 2018, support may have limitations and there is no assurance that future versions of GVP will preserve backward support for PSTNC.
The latest PSTNC release is 8.1.4, and it requires MCP 8.1.4 to be compatible with GVP 8.1.5 and above. Install MCP 8.1.4 into a GVP logical resource group, and it will be able to talk to PSTNC 8.1.4.

Installation of the PSTN Connector is required to integrate TDM networks with GVP and facilitate migration to GVP 8.x. TDM integration is supported through Dialogic telephony technology only. </toggledisplay>

1. Verify that the PSTN Connector host is prepared for installation. See Preparing the Hosts for GVP.
2. Verify that the PSTN Connector Application object template is imported, and an Application object is created. See Preinstallation Activities.
3. Execute the setup.exe setup file:
    - If you are using the GVP software DVDs, browse to the <GVP_Installation_DVD>\solution_specific\windows\pstnc\ folder.
    - If the DVD image is on a network drive, copy the <DVDImage>\solution_specific\windows\pstnc\ folder to the local computer.
4. When the Genesys Deployment Wizard appears, click **Next**.
5. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.
6. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
7. On the Select Application page, select the PSTN Connector Application object.
8. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.
9. On the Ready to Install page, click **Install**.
10. When the installation is complete, click **Finish**.
11. Configure the PSTN Connector Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
12. If you intend to use the Policy Server in your environment, see the tab **Policy Server** in this topic.

Policy Server

Procedure: Installing the Policy Server (Windows)

1. Verify that the Policy Server host is prepared for installation. See Preparing the Hosts for GVP.
2. Verify that the Policy Server Application object template is imported, and an Application object is created. Preinstallation Activities.
3. Execute the setup.exe setup file:
    - If you are using the GVP software DVDs, browse to the <GVP_Installation_DVD>\solution_specific\windows\ps\ folder.

    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\ps\` folder to the local computer.
4. When the Genesys Deployment Wizard appears, click **Next**.
5. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.
6. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
7. On the Select Application page, select the Policy Server Application object.
8. Select the destination folder in one of two ways:
    - Click **Next** to accept the default directory
    - Click **Browse** to select the destination folder, and then click **Next**.
9. On the Ready to Install page, click **Install**.
10. When the installation is complete, click **Finish**.
11. Configure the Policy Server Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
12. If you intend to use the MRCP Proxy in your environment, see the tab **MRCP Proxy** in this topic.

MRCP Proxy

Procedure: Installing the MRCP Proxy (Windows)

1. Verify that the MRCP Proxy host is prepared for installation. See Preparing the Hosts for GVP.
    - The MRCP Proxy Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. Execute the setup.exe setup file:
    - If you are using the GVP software DVDs, browse to the `<GVP_Installation_DVD>\solution_specific\windows\mrcpp\` folder.
    - If the DVD image is on a network drive, copy the `<DVDImage>\solution_specific\windows\mrcpp\` folder to the local computer.
3. When the Genesys Deployment Wizard appears, click **Next**.
4. On the Connection Parameters page, enter the information in the Host and User sections, as shown in the table below.

Table: Connection Parameters for Configuration Server

Section	Field	Description

Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

5. On the Client Side Port Configuration page, select **Use Client Side Port** (if required). Enter the Port and IP Address.
6. On the Select Application page, select the MRCP Proxy Application object.
7. Select the destination folder in one of two ways:
   - Click **Next** to accept the default directory
   - Click **Browse** to select the destination folder, and then click **Next**.
8. On the Ready to Install page, click **Install**.
9. When the installation is complete, click **Finish**.
10. Configure the MRCP Proxy Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

Table: Connection Parameters for Configuration Server

Table: Connection Parameters for Configuration Server

Section	Field	Description
Host	Host name	Enter the host name or IP address of the Configuration Server.
	Port	Enter the port number of the Configuration Server.
User	User name	Enter the user name that is used to log in to the Configuration Server.
	Password	Enter the password that is used to log in to the Configuration Server.

Preparing to Install GVP Manually on Linux

This topic describes how to install Genesys Voice Platform (GVP) manually on the Linux operating system (OS) by using the executable files in the GVP Installation Packages. It contains the following sections:

- Task Summaries
- Preparations
- Installing and Configuring the PSTN Connector

Task Summaries

Task Summary: Preparing Your Environment for GVP (Linux)

Objective	Related Procedures and Actions
Plan the deployment	For specific restrictions and recommendations to consider, see Preparing the Hosts for GVP.
Prepare your environment: **Install common Genesys Framework**	1. Management Framework. Install the latest Installation Package (IP) for the Genesys Management Framework, and ensure that it is fully operational and running. See the Framework Deployment Guide. Management Framework is the centralized element management system for all Genesys software.
Prepare your environment: **Install common Genesys Framework components**	2. Genesys Administrator. Install Genesys Administrator, and ensure that it is fully operational. See the Framework Deployment Guide. Genesys Administrator is the centralized management GUI for all Genesys software.
	3. Genesys SNMP Master Agent. Install and configure the SNMP Master Agent on the same host(s) as the Resource Manager, Media Control Platform, Call Control Platform, and Supplementary Services Gateway components. (After the SNMP Master Agent is installed on the GVP hosts, you will assign the SNMP Master Agent to each component for which you want to capture alarm and trap information. This is a post-installation activity; see Procedure: Creating a Connection to a Server.

	The Genesys Voice Platform 8.5 DVD includes an MIB Installation Package that can be loaded on the SNMP management console (for example, HP Open View) in your environment. To install the MIBs, run the executable file and select the default installation path: `sh install.sh /opt/genesys/gvp/VP_MIB_8.5` **Tip** The SNMP Master Agent is required only if you are capturing alarm and trap information. For more information about installing the Management Framework and the SNMP Master Agent, see the Framework Deployment Guide. For more information about the MIBs, see the GVP 8.5 SNMP MIB Reference.
Prepare your environment: **Install third party software**	4. Third-party hardware and software. If you are using automatic speech recognition (ASR) and/or text-to-speech (TTS), install the third-party Media Resource Control Protocol (MRCP) speech server, and ensure that it is operational. For more information about this software, see your MRCP vendor s documentation. For information about prerequisite software, see Prerequisites.
Prepare the host(s)	5. Stop antivirus software that might be running on systems that will host GVP components. Check the vendor documentation for your antivirus software configuration.
	6. Install the Local Control Agent on the GVP hosts.

	See Procedure: Installing the Local Control Agent (Linux).

Complete the prerequisites	7. On Reporting Server and Policy Server hosts, install the Sun JRE 7.0 or later, platform:
	0. Install the Java Runtime Environment (JRE). Obtain the latest Red Hat Package Manager (RPM) or the self-extracting files for Linux from the vendor's website, and follow the instructions for installing it.
	a. When the installation is complete, change to the root user:
	Enter `su`.
	b. Modify the default Java configuration. Enter:
	`/usr/sbin/alternatives --install /usr/bin/java java /usr/java/jre1.7.0_x/bin/java 2000`
	`/usr/sbin/alternatives --config java`
	c. Set up the JRE variables. Modify `./etc/profile`:
	Enter `export JAVA_HOME= /usr/ java/ jre1.7.0_x`.
	d. Exit the current console session, and log in again.
	e. Verify that the configuration is correct. Enter:
	`<tt>echo $JAVA_HOME`. You should see the following output: `/usr/java/JRE1.7.0_06`
	f. Setup the JRE variables in the Genesys LCA (Local Control Agent) and GDA (Genesys Deployment Agent) startup scripts. Modify the `/etc/init.d/gctilca` and `/etc/init.d/gctigda` files by adding the line
	`export JAVA_HOME= /usr/java/jre1.7.0_x`
	at the beginning of these files.
	g. Restart the LCA and the GDA by terminating the LCA and GDA processes, and executing the `/etc/init.d/gctigda start` and `/etc/init.d/gctilca start` commands. For more information about the prerequisite software, see Prerequisites or visit the vendor's website.

	8. On the Media Control Platform, ensure that the user account that is used to perform the installation is the root account with

	the necessary privileges to create directories under the /var/ www directory. The `/var/www/gvp/mcp` folder can then be created during installation, and other folders and files created in this directory. (If this directory does not exist, create it manually prior to installation.)
	9. On the Media Control Platform, if the Apache Web Server is not already installed: 0. Obtain the latest Red Hat Package Manager (RPM) for the Apache Web Server from the vendor s website, and follow the instructions for installing it. a. On the Apache Web Server, modify the `/etc/mime.types` file. Enter `/application/srgs+xml` b. Configure the Apache Web Server to start automatically at startup. Modify the `/etc/rc.d/rc.local` file. Enter `/etc/init.d/httpd start`. For more information about the prerequisite software, see Prerequisites or visit the vendor's website. **Tip** In GVP 8.1.3 and earlier 8.x releases, Apache HTTP Server was required to host inline and universal hotkey grammar files that were fetched by ASR. In GVP 8.1.4, Apache is no longer required. The Media Control Platform now transmits these grammars by default in the MRCP requests.

	10. If you are including the PSTN Connector in your environment, visit the vendor's website for information about how to install and configure Dialogic hardware and software.

Task Summary: Deploying GVP Manually (Linux)

Objective	Related Procedures and Actions
Configure the host(s)	1. Configure a new host in the Configuration Database for each computer that is hosting GVP components. See Configuring a Host in Genesys Administrator.
Complete the Application objects	2. Create the GVP Application objects: 0. Import the templates. See Procedure: Importing Application Object Templates Manually. a. Create the Application objects. See Procedure: Creating Application Objects Manually.
Install GVP	3. Install the GVP components, as described: a. Procedure: Installing the Media Control Platform (Linux). b. Procedure: Installing the Call Control Platform (Linux). c. Procedure: Installing the Resource Manager (Linux). d. Optional Procedure: Installing the Supplementary Services Gateway (Linux). e. Procedure: Installing the Policy Server (Linux). f. Procedure: Installing the MRCP Proxy (Linux). g. Procedure: Installing the Reporting Server (Linux). h. Procedure: Installing the PSTN Connector (Linux). 1. Procedure: Installing LiS. 2. Procedure: Installing Dialogic, on page 358]]].
Start the components	4. Start the components manually (or configure the components to start automatically). See Startup Sequence for the VPS and Starting and Stopping the Components.
Complete the post-installation activities	5. Configure the GVP components for the functionality you want use in your deployment. See Task Summary: Post-Installation Configuration of GVP.

Preparations

You can install GVP on a Linux OS in a new deployment or add GVP components to an existing deployment.

Before you begin to install the components, complete the procedures in the following sections; Preparing the Hosts for GVP and Preinstallation Activities.

In addition, copy the GVP installation packages to a directory on the Linux hosts or to a network drive from which they can be downloaded.

Pre-installation Notes

Multiple Instances of the same component

If you are installing multiple instances of the same component, Genesys recommends that you install each instance on a different host.

Starting in 8.1.2, GVP supports multiple instances of the Media Control Platforms on a single host. For more information about installing multiple instances of the Media Control Platform, see Deploying Multiple Media Control Platforms.

/etc/hosts and the local IP Address

For Linux systems, /etc/hosts must correctly reflect the non-loopback IP address for the local hostname, so that GVP can automatically determine the local IP address for use in various network related operations. Example:

```
[pw@HOMER etc]$ hostname
HOMER
[pw@HOMER etc]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
x.x.x.x HOMER
```

Squid Caching Proxy and Fetching Module

In GVP 8.1.2 and higher, the Fetching Module is integrated with the Media and Call Control Platforms and is no longer a separate installation package. In addition, the Squid proxy is an optional component.

Installation Directory

Genesys recommends that you use `/opt/genesys/gvp/` for the installation directory, where `VP_Component_8.5.000.xx` is the name and release number of the component you are installing.

Root User

Notes:

- GVP release 8.1.4 and above allows a non-root user to perform the Linux installation, but that user must have write permission to the `/var/www/` directory.
- If a root user performs the installation, the system expects that installed files will have user ID `234` assigned.

Manually Installing GVP on Linux

These instructions install each GVP module separately, one per tab.

Media Control Platform

Procedure: Installing the Media Control Platform (Linux)

Install the Media Control Platform component to enable Session Initiation Protocol (SIP) applications that use Voice Extensible Markup Language (VoiceXML) to access the Media Control Platform media services.

Before you start the Media Control Platform and Call Control Platform Application objects, you must start the Apache Hypertext Transfer Protocol (HTTP) Server and (if you use the Squid) the Squid caching proxy.

1. Verify that:
    - The Apache HTTP Server is installed, and the application is started. (Apache is no longer a prerequisite in 8.1.2.) See "Complete the prerequisites" in the Task Summary: Preparing Your Environment for GVP.
    - The Media Control Platform host is prepared for the installation. See Preparing the Hosts for GVP.
    - The Media Control Platform Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. At the Linux host, log in as `root`, and then enter `su`.
3. Navigate to the directory that contains the Media Control Platform installation package.
4. Enter `chmod a+x install.sh`.
5. Run the `./install.sh` command.
   The installation script is initiated.
6. At the prompt, enter the hostname of the Media Control Platform server. For example:

```
Please enter the host name or press enter for "<local_host>"
=> <local_host>.
```

7. At the prompt, enter the information that is required for the Configuration Server for example:

```
Configuration Server hostname
=><config_serv>
Network port =>2020
User name =>default
Password =>password
```

8. At the prompt, enter the information, if required, for the Client Side Port Definitions for example:

```
Do you want to use Client Side Port option (y/n)?y
Client Side Port port =>1234
Client Side IP Address (optional), the following values can
be used
10.0.0.222
10.0.0.254
=>10.0.0.222
```

9. At the prompt, choose the application that you want to install. For example:

```
1 : MCP-Host
2 : MCP_8.5.000.09
3 : MCP_8.5.000.19
=>3
```

10. At the prompt, choose the audio format for your region for example:
    - `Mulaw` (North America)
    - `Alaw` (Europe)
11. Select one of three options for HTTP Proxy mode:
    - `use HTTP proxy "localhost"`
    - `disable HTTP proxy`
    - `specify HTTP proxy`
12. If you selected `specify HTTP proxy` in the previous step:
    - Enter the HTTP proxy host.
    - Press **Enter** to confirm `3128` as the HTTP proxy port or enter a new one.
13. After the following output is displayed, enter `y` or `n` at the prompt:

```
If you are using GVP VoiceXML applications on this server,
you need to enable VoiceXML applications.

If you are using the Genesys Media Server with Play
Application treatments (also VoiceXML),
e.g., from routing strategies, then you need to purchase GVP
ports and enable VoiceXML applications.

Otherwise, you do not need to enable VoiceXML applications.

Do you wish to enable VoiceXML applications (y/n)? =>y
```

14. At the Next prompt, enter the path to the directory in which the application files will reside for example:

```
Press ENTER to confirm /<Install_Dir>/gvp81/MCP_8.5.000.xx as
the destination directory
or enter a new one => /opt/genesys/gvp/
VP_Media_Control_Platform_8.5.000.xx
```

A message appears that indicates that the installation files are being extracted and copied to the directory. Then, a final message appears which indicates that the installation was completed successfully.

15. Configure the Media Control Platform Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
    To start any Application object manually on a Linux host, Enter `<Install_Dir>/bin/run.sh` where `<Install_Dir>` is the directory in which the application is installed.
16. If required, install the Call Control Platform. See the tab **Call Control Platform** in this topic.

Call Control Platform

Procedure: Installing the Call Control Platform (Linux)

Install the Call Control Platform component, so that applications that use Call Control Extensible Markup Language (CCXML) can access the Call Control Platform call-processing services.

Before you start the Media Control Platform and Call Control Platform Application objects, you must start the Apache Hypertext Transfer Protocol (HTTP) Server and (if you use the Squid) the Squid caching proxy.

1. The Call Control Platform host is prepared for the installation. See Preparing the Hosts for GVP.
2. The Call Control Platform Application object template is imported and an Application object is created. See Preinstallation Activities.
3. At the Linux host, log in as root and enter `su`.
4. Navigate to the directory that contains the Call Control Platform installation package.
5. Complete Steps 3 to 13 under the tab **Media Control Platform** in this topic, substituting information for the Call Control Platform, where necessary.
6. Configure the [ems]log_sinks parameter, see Task Summary: Preparing Your Environment for GVP.
7. Configure the Call Control Platform Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
8. Install the Resource Manager. See the tab **Resource Manager** in this topic.

Resource Manager

Procedure: Installing the Resource Manager (Linux)

1. Verify that:

   - The Resource Manager host is prepared for the installation. See Preparing the Hosts for GVP.
   - The Resource Manager Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. At the Linux host, log in as root, and then type `su`.
3. Navigate to the directory that contains the Resource Manager installation package.
4. Complete Steps 3 to 13 under the tab **Media Control Platform** in this topic, substituting information for the Resource Manager, where necessary.
5. Configure the Resource Manager Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

Supplementary Services Gateway

Procedure: Installing the Supplementary Services Gateway (Linux)

1. Verify that:
   - The Supplementary Services Gateway host is prepared for the installation. See Preparing the Hosts for GVP.
   - The Supplementary Services Gateway Application object template is imported, and an Application object created. See Preinstallation Activities.
2. At the Linux host, log in as root, and enter `su`.
3. Navigate to the directory that contains the Supplementary Services Gateway installation package.
4. Complete Steps 3 to 13 under the tab **Media Control Platform** in this topic, substituting information for the Supplementary Services Gateway, where necessary.
5. Configure the Supplementary Services Gateway Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

CTI Connector

Procedure: Installing the CTI Connector (Linux)

1. Verify that:
   - The CTI Connector host is prepared for installation. See Preparing the Hosts for GVP.
   - The CTI Connector Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. At the Linux host, log in as root and enter `su`.
3. Navigate to the directory that contains the CTI Connector installation package.

4. Enter `chmod a+x install.sh`.
5. Run the `./install.sh` command.
   The installation script is initiated.
6. At the prompt, enter the hostname of the CTI Connector server. For example:

```
Please enter the host name or press enter for "<local_host>"
=> <local_host>
```

7. At the prompt, enter the information that is required for the Configuration Server. For example:

```
Configuration Server hostname =><config_serv>
Network port =>2020
User name =>default
Password =>password
```

8. At the prompt, enter the information, if required, for the Client Side Port Definitions. For example:

```
Do you want to use Client Side Port option (y/n)?y
Client Side Port port =>1234
Client Side IP Address (optional), the following values can
be used
10.0.0.222
10.0.0.254
=>10.0.0.222
```

9. At the prompt, choose the application that you want to install. For example:

```
1 : CTI_Connector_ICM
2 : CTI_Connector_IVRSC
=>2
```

10. At the prompt, choose the CTI FrameWork. For example:

```
1: Genesys CTI
2: Cisco ICM
=>1
```

11. At the Next prompt, enter the path to the directory where the application files will reside. For example:

```
Press ENTER to confirm /opt/genesys/gvp/VP_CTIC_Connector_8.5
as the destination directory or enter a new one =>
/opt/genesys/gvp/CTI_Connector_IVRSC
```

A message indicates that the installation files are being extracted and copied to the directory.

Then, a final message indicates that the installation was completed successfully.

12. Configure the CTI Connector Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

PSTN Connector

Installing the PSTN Connector is similar to installing other GVP 8.x components. You can accomplish it with the deployment option in Genesys Administrator or by executing the setup.exe on the PSTN Connector host.

JCT-specific Configuration

For JCT boards, the PSTN Connector parameter MediaVoxResourceBoard must be configured with route number information for the different board used for CSP. Please refer to parameter help for more details.

Configuring Dialogic

- Use Dialogic Service update `229`

Path to these services updates: `\\inccisss003\PlatformTeam\VCS\Dialogic\SystemReleases\SR 6.0\ServiceUpdate229`

For Dialogic configuration for different TDM protocols, please refer to this document.

Specific Configuration for Windows 2008

The complete path to the Dialogic package for Windows 2008 Server is:
`\\inccisss0003\genesys\PlatformTeam\VCS\Dialogic\SystemReleases\SR 6.0\ServiceUpdate239\win2008\vista_red.zip`

You *must* disable Physical Address Extension (PAE) on Windows 2008; otherwise Dialogic may not function properly. Please refer to this Dialogic web page for details.

To disable PAE, run these commands from the command line:

```
C:\bcdedit /set nx OptOut
C:\bcdedit /set pae ForceDisable
```

Then restart the server.

Interworking with SIP-Server

To enable the PSTN Connector to interwork with SIP-Server, configure the PSTN Connector as a Trunk DN with the following parameters:

```
[TServer]contact=<PSTNIPAddr:port>
[TServer]prefix=<xyz>
[TServer]replace-prefix= (Empty String)
```

This configuration specifies that the outbound call lands on the same PSTN Connector instance from where the inbound call is received.

Notes:

- The trunk DN, `[TServer]contact=<`*`same as PSTN Connector contact`*`>`, must exist before PSTN Connector starts.
- In order to find the trunk DN, the SIP-Server application must be attached to the PSTNC application in the connections tab.
- The change in SIP-Server application/Switch is not considered at run time.
- Once the trunk DN is deleted, the prefix value is set to empty value and is not modified until the restart of PSTNC.
- Though SIP-Server application is attached in the connections tab, the options `UserAgentAddr` and `UserAgentPort` should be configured with SIP Server IP address and listening SIP port.

Execute this procedure: Installing and Configuring the PSTN Connector.

Policy Server

Procedure: Installing the Policy Server (Linux)

1. Verify that:
   - Management Framework components (Configuration Server and Genesys Administrator) have been upgraded. See Table: Versions Compatible With GVP.
   - The Policy Server host is prepared for installation. See Preparing the Hosts for GVP.
   - The Policy Server Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. At the Linux host, log in as root and enter `su`.
3. Navigate to the directory that contains the Policy Server installation package.
4. Complete Steps 3 to 13 under the tab **Media Control Platform** in this topic, substituting information for the Policy Server, where necessary.
5. Configure the Policy Server Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

MRCP Proxy

Procedure: Installing the MRCP Proxy (Linux)

1. Verify that:
   - Management Framework components (Configuration Server and Genesys Administrator) have been upgraded. See Table: Versions Compatible With GVP.
   - The MRCP Proxy host is prepared for installation. See Preparing the Hosts for GVP.
   - The MRCP Proxy Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. At the Linux host, log in as root and enter `su`.
3. Navigate to the directory that contains the MRCP Proxy installation package.
4. Complete Steps 3 to 13 under the tab **Media Control Platform** in this topic, substituting information for the MRCP Proxy, where necessary.
5. Configure the MRCP Proxy Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
6. Complete the prerequisites for the Reporting Server. See "Complete the prerequisites" in the Task Summary: Preparing Your Environment for GVP.
7. Install the Reporting Server. See the tab **Reporting Server** in this topic.

Reporting Server

Procedure: Installing the Reporting Server (Linux)

1. Verify that:
    - Oracle is the only supported database for Linux. In this procedure, when you select the database, you can choose the Standard or Enterprise edition of the database. If you select the Enterprise edition, partitioning of the database is enabled automatically during installation.
    - When database partitioning is enabled, Genesys recommends that you not change the partitioning mode of operation or the number of partitions (even after the Reporting Server is started) because of issues that might arise if the database schema or stored data is changed.
    - Database partitioning is supported in GVP 8.1.2 only. If you are installing GVP 8.1.1 or earlier 8.x versions, the option to select the Enterprise edition is not available.
    - The Sun Java Runtime Environment (JRE) 6.0, Update 19 or later is installed. Sun JRE 7.0 or later is recommended. See "Complete the prerequisites" in the Task Summary: Preparing Your Environment for GVP.

    > Tip
    >
    > JRE 7.0 or later is required if you are using IPv6 communications.

    - The Reporting Server host is prepared for installation. See Preparing the Hosts for GVP.
    - The Reporting Server Application object template is imported, and an Application object is created. See Preinstallation Activities.
2. At the Linux host, log in as root, and then type `su`.
3. Navigate to the directory that contains the Reporting Server installation package.
4. Complete Steps 3 to 6 under the tab **Media Control Platform** in this topic, substituting information for the Reporting Server, where necessary.
5. At the prompt, choose the application that you want to install. For example:

    ```
    1 : RS-Host
    2 : RS_8.5.000.09
    3 : RS_8.5.000.19
    =>3
    ```

6. At the prompt, enter the number associated with the database server you want to selec. For example:

```
Please specify the type of Database Server used:
1) Oracle 10g/11g Standard Edition
2) Oracle 10g/11g Enterprise Edition
3) MS SQL Server 2005 or MS SQL Server 2008 Standard Edition
4) MS SQL Server 2008 Enterprise Edition
=>1
```

> **Tip**
>
> GVP supports only Oracle 10g or 11g Database Servers on Linux.

7. At the prompt, confirm (or enter) the database host name or IP address. For example:

   ```
   Press ENTER to confirm "10.10.15.152" as the Database Server
   hostname or IP address or enter a new one =>
   ```

8. At the prompt, press **Enter** to confirm the database-server port number. For example:

   ```
   Press ENTER to confirm "1433" as the Database Server port or
   enter a new one =>
   ```

9. At the prompt, confirm or enter the name of the database server. For example:

   ```
   Press ENTER to confirm "RS" as
   the Database name or enter a new one =>
   ```

10. At the prompt, press Enter to confirm the user name of the database server. For example:

    ```
    Press ENTER to confirm "sa" as the Database Server user name
    or enter a new one =>
    ```

11. At the prompt, enter the password. For example:

    ```
    Please specify the Database Server user password =>
    ''password''
    ```

12. At the prompt, press **Enter** to confirm the Reporting Server port number. For example:

```
Press ENTER to confirm "61616" as the VP Reporting Server
port or enter a new one =>
```

13. At the prompt, press Enter to confirm the Web Server port number. For example:

```
Press ENTER to confirm "8080" as the VP Reporting Server Web
Service port or enter a new one =>
```

14. At the prompt, enter the path to the directory in which the application files will reside. For example:

```
Press ENTER to confirm /opt/genesys/gvp/RS_8.5.000.xx as
the destination directory or enter a new one =>
/opt/genesys/gvp/VP_Reporting_Server_8.5.000.xx
```

> Tip
>
> Genesys recommends you use /opt/genesys/gvp/ for that the installation directory, where VP_Component_8.5.000.xx is the name and version number of the component that you are installing.
>
> A message appears that indicates that the installation files are being extracted and copied to the directory. Then, a final message appears that indicates that the installation was completed successfully.

15. Configure the Reporting Server Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

Installing and Configuring the PSTN Connector

- Procedure: Installing the PSTN Connector (Linux)
- Dialogic Installation

Procedure: Installing the PSTN Connector (Linux)

Install and use the PSTN Connector on the host only after careful consideration—because the Dialogic boards used in PSTNC are no longer sold. Although Dialogic supports this hardware until 2018, that support may have limitations and there is no assurance that future versions of GVP will preserve backward support for PSTNC.

1. Verify that:
    - The PSTN Connector host was prepared for the installation. See Preparing the Hosts for GVP.
    - The PSTN Connector Application object template was imported, and an Application object was created. See Preinstallation Activities.
2. At the Linux host, log in as the root user and enter `su`.
3. Navigate to the directory that contains the PSTN Connector installation package.
4. Complete Steps 3 to 13 of the procedure Manually Installing GVP on Linux, substituting PSTN Connector for Media Control Platform.
5. Configure the PSTN Connector Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

Dialogic and gcc support

A Dialogic installation automatically compiles its drivers. You must install gcc to enable this functionality.

- The official versions of gcc supported by Dialogic are gcc 3.2, gcc 3.4.4, and gcc 3.4.6. There is no official support extended to the latest versions of gcc viz or gcc 4.x. However, Dialogic confirms that even if their drivers are built with gcc 4.x compiler, it is acceptable if you have also installed hcc 3.4 backwards-compatibility libraries.
- With RHEL5 installation, gcc 4.x is the default version installed. This creates a conflict because Dialogic drivers are either compiled with gcc 3.4 or gcc 3.2. To avoid any discrepancies in the functionalities of the drivers, Dialogic suggests installing the compatibility-gcc 3.4.x libraries during the installation of the Linux OS. This is performed during the OS-installation steps. Thus, no additional steps are required for RHEL5 for installing Dialogic.

Dialogic Installation

This installation contains two procedures and two configurations:

- Installing LiS
- Installing Dialogic

- Configuring DMV Boards
- Configuring JCT Boards

Procedure: Installing LiS

LiS is a mandatory component; Dialogic will not install without it. For this, LiS drivers have to be made into Kernel modules. This step requires availability of Linux source code on the system. This was already installed as part of the OS installation.

The LiS package is provided with the Dialogic package; there is no need to download LiS separately.

1. To create a tar file, unzip the .gz file in the Dialogic package:
   `gunzip lnxdlgcsu317.tar.gz`
2. Untar the tar file:
   `tar xf lnxdlgcsu317.tar`
   Untarring creates the directories needed by the Dialogic installation.
3. Unzip and untar the LiS package file.
   Under the `redistributable-sources/` directory, you can find the `LiS/` directory where the LiS package is located, because a `.gz` file will be present.
   Go to the directory `LiS2.x/` and issue the command:
   `# make` This step prompts for various options but the first prompt asking whether to run LiS as Kernel module or User module alone is the most important. Choose the `Kernel module` option in this step. All subsequent options can have the default values. Once this is complete without errors, enter this command:
   `# make install`
4. Reboot the system.
5. Install Dialogic.

Procedure: Installing Dialogic

1. Go to the top directory of the untarred Dialogic package and start the installation:
   `install.sh`
2. Subsequent steps offer various Dialogic packages for installation. Only these two are important to install:
   - Drivers for DMV/JCT boards
   - Global call library packages
3. If the installer asks Please install LiS, then reboot the system.

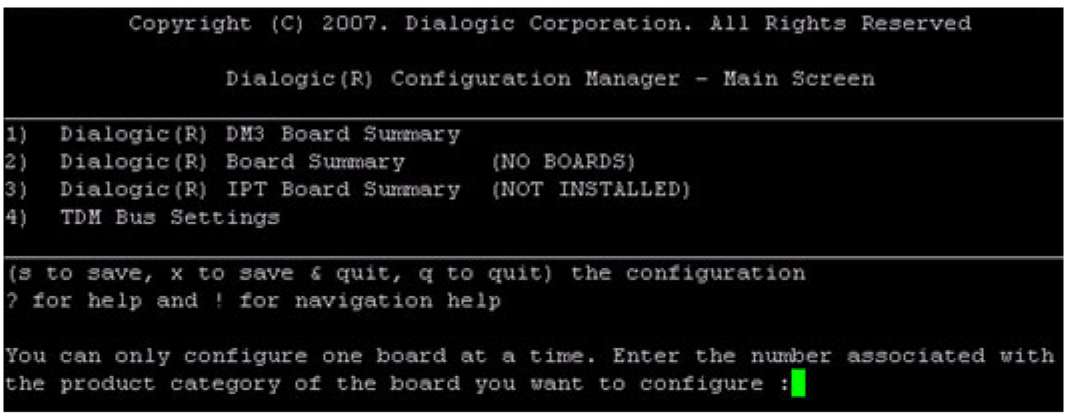
4. When the installer asks to install redistributable-sources package, select `Yes` and press **Enter**. For further installation/configuration instructions, if needed, please refer to the Dialogic installation documents.
5. Configure your Dialogic boards. See Configuring Dialogic Boards, below.

Configuring Dialogic Boards

When the installation is complete, the Dialogic installer prompts you to run config.sh. Select `Y` to proceed with the configuration. There are two other ways to begin the configuration:

- Run `config.sh` in the `/redistributable-runtime/` directory.
- Run the `CFG` utility in the Dialogic installation `/bin/` directory.

Both methods begin the installation and display the same first screen:



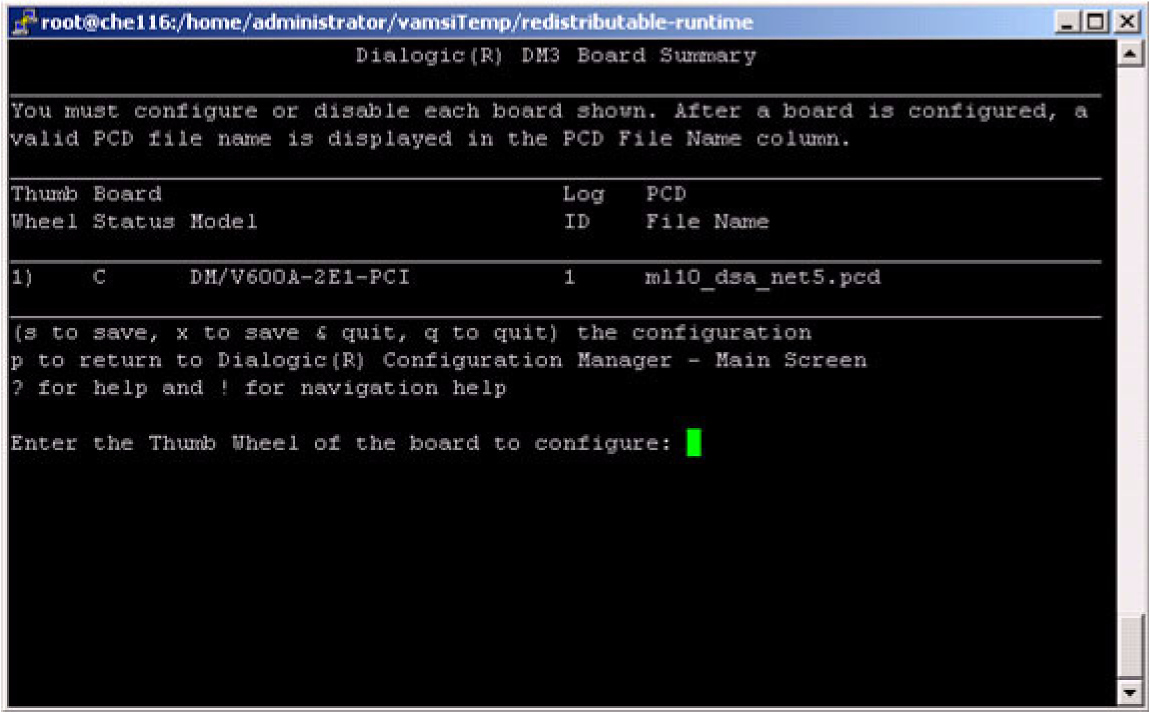
Dialogic Board Configuration Opening Screen

From here, perform the appropriate procedure:

[+] Configuring DMV Boards

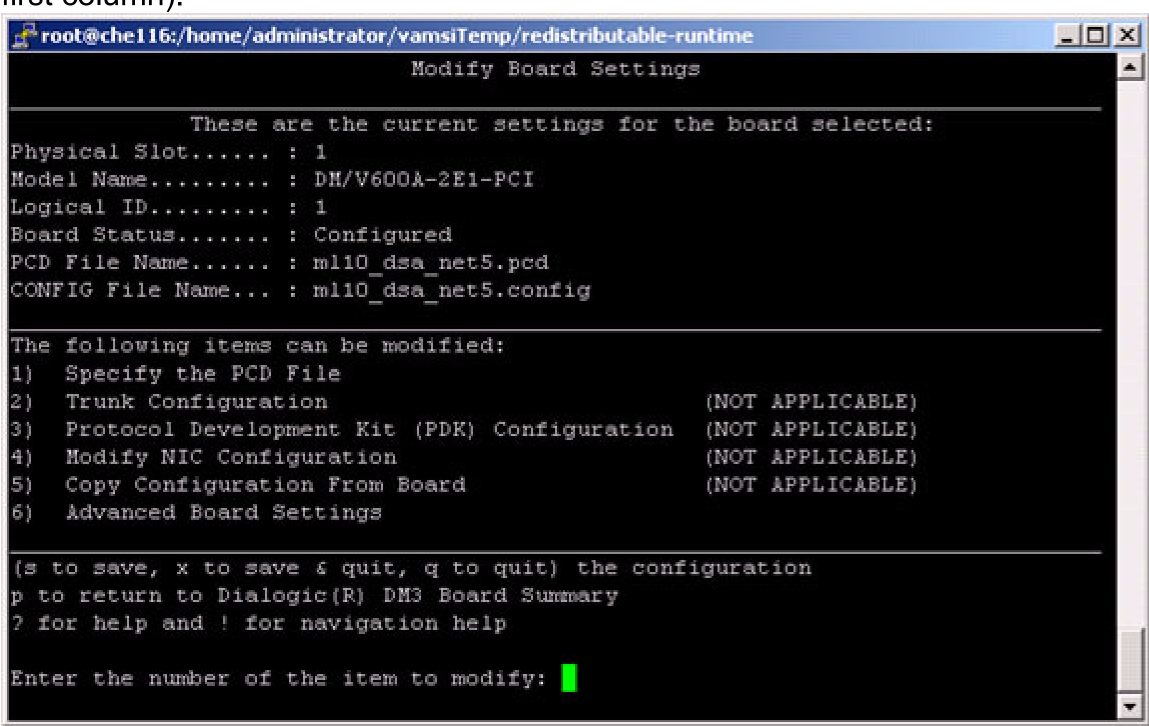
Use the screen shots to guide yourself through the DMV board configuration. Your screens will vary slightly.

1. Enter `1` to see the DMV board summary.



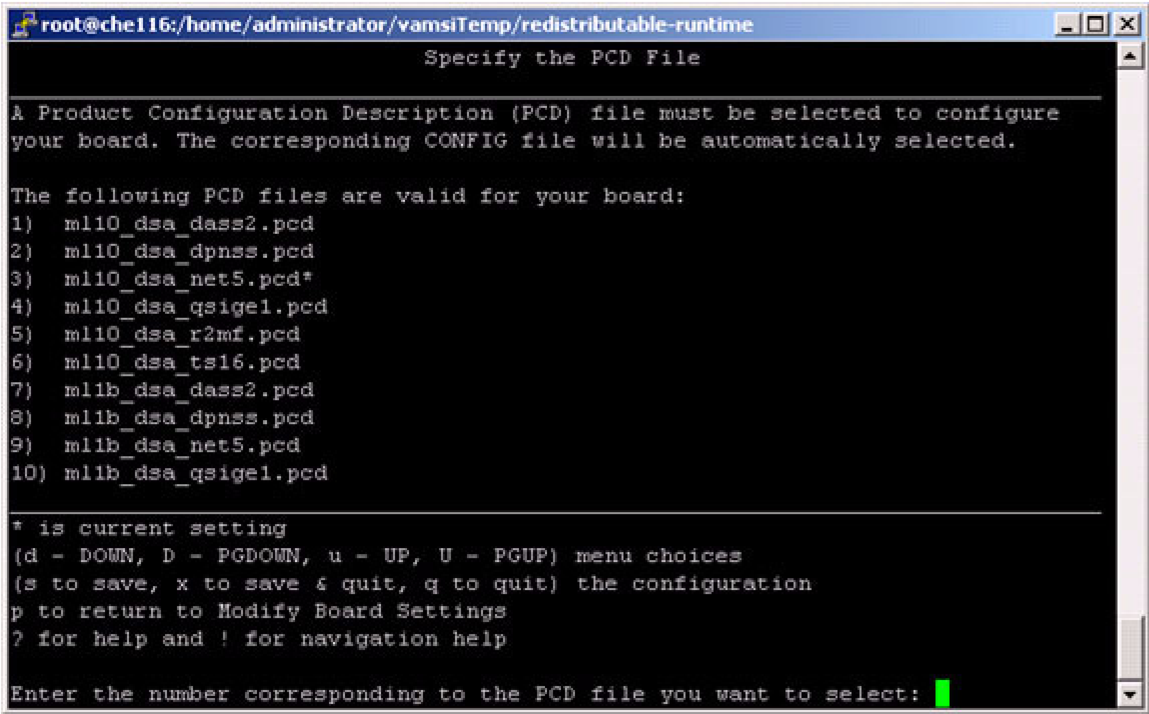
DM3 Board Summary

2. In the screen shot, there is one DMVA card installed on the computer. To configure it: at the prompt, enter `1` (the number of the Thumb Wheel of the board, listed in the first column).



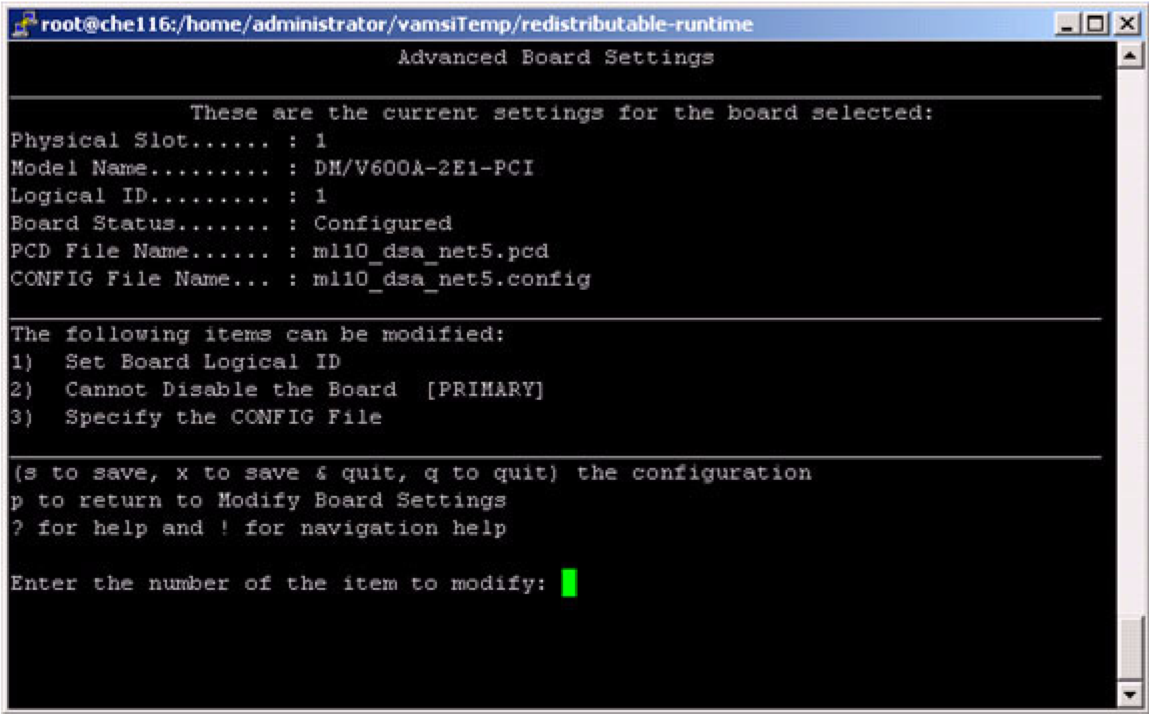
Modify Board Settings

3. Enter `1` to set the PCD file.



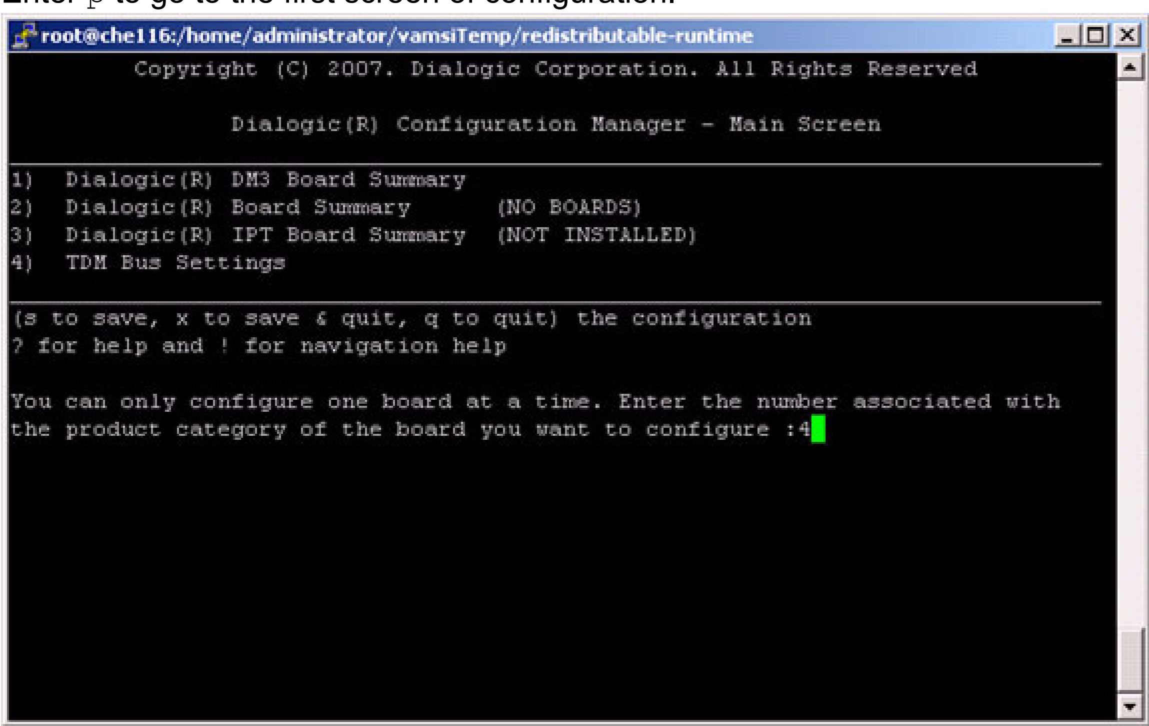
Specify the PCD File

4. Specify the appropriate pcd file.
   Once configured, the configuration returns to the previous screen.
5. Enter `6` to go to the advanced board settings step.



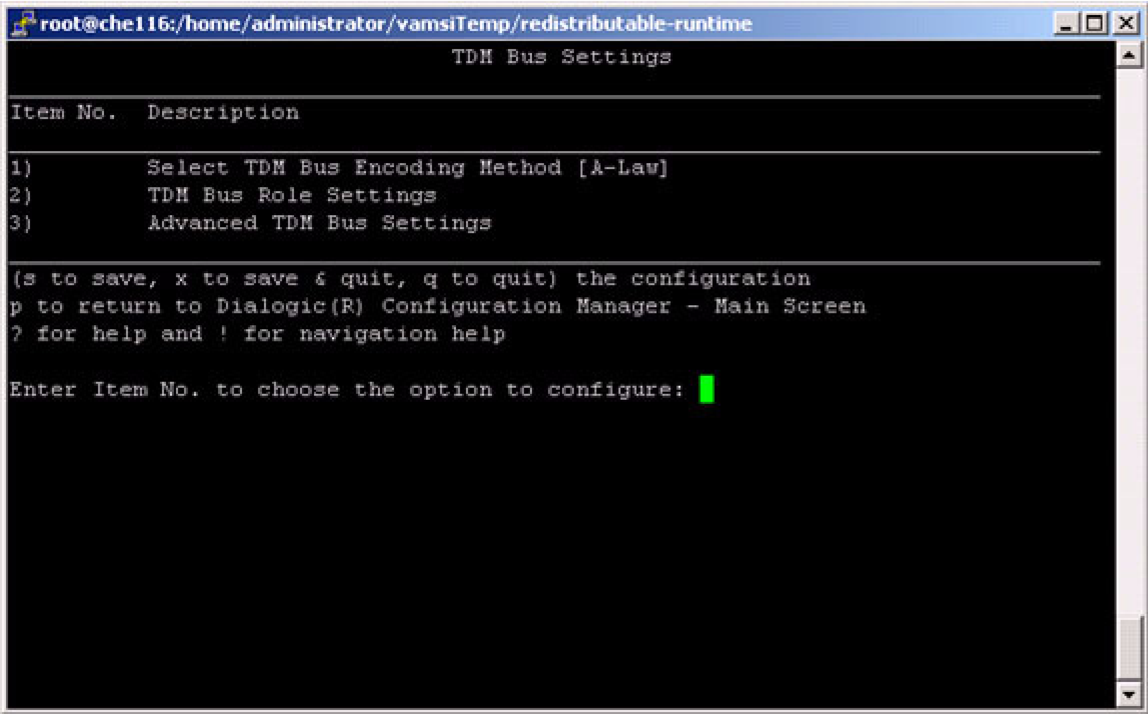
Advanced Board Settings

6. Set the parameters appropriately and save the configuration.
7. Enter `p` to go to the first screen of configuration.



Configuration Manager - Main Screen

8. For any TDM bus settings, enter `4`.



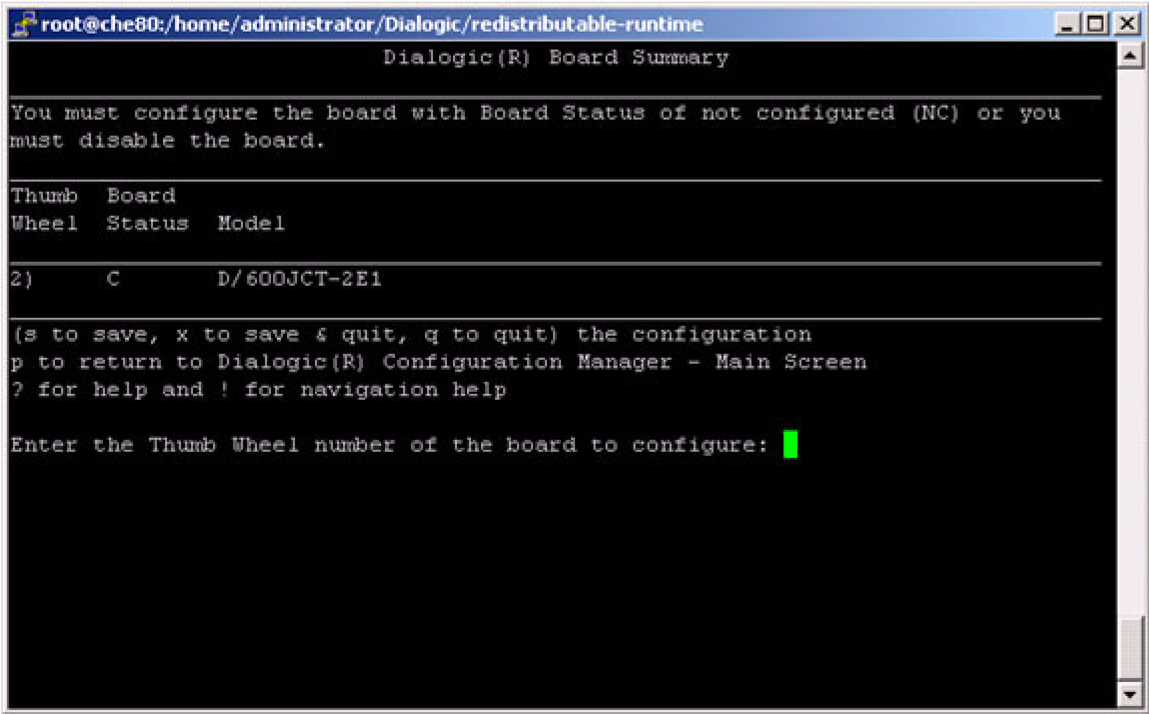
TDM Bus Settings

9. Choose the next configuration step and set the parameters appropriately such as the encoding method, TDM bus settings of primary and secondary masters (this is mostly not required).
10. Save the configuration and exit.
11. Start Dialogic one of these two ways:
    `/etc/init.d/ct_intel start`
    OR
    `<Dialogic bin dir>/dlstart`

[+] Configuring JCT Boards

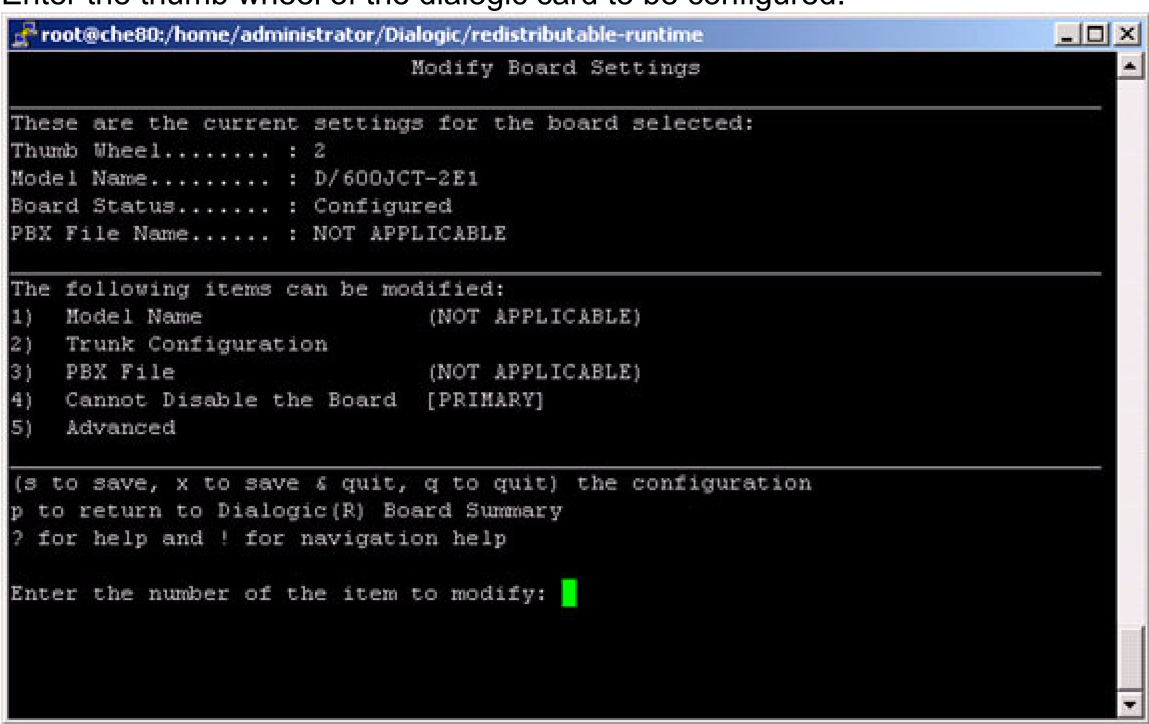
Use the screen shots to guide yourself through the JCT board configuration. Your screens will vary slightly.

1. Enter `2` to display the Dialogic Board Summary.



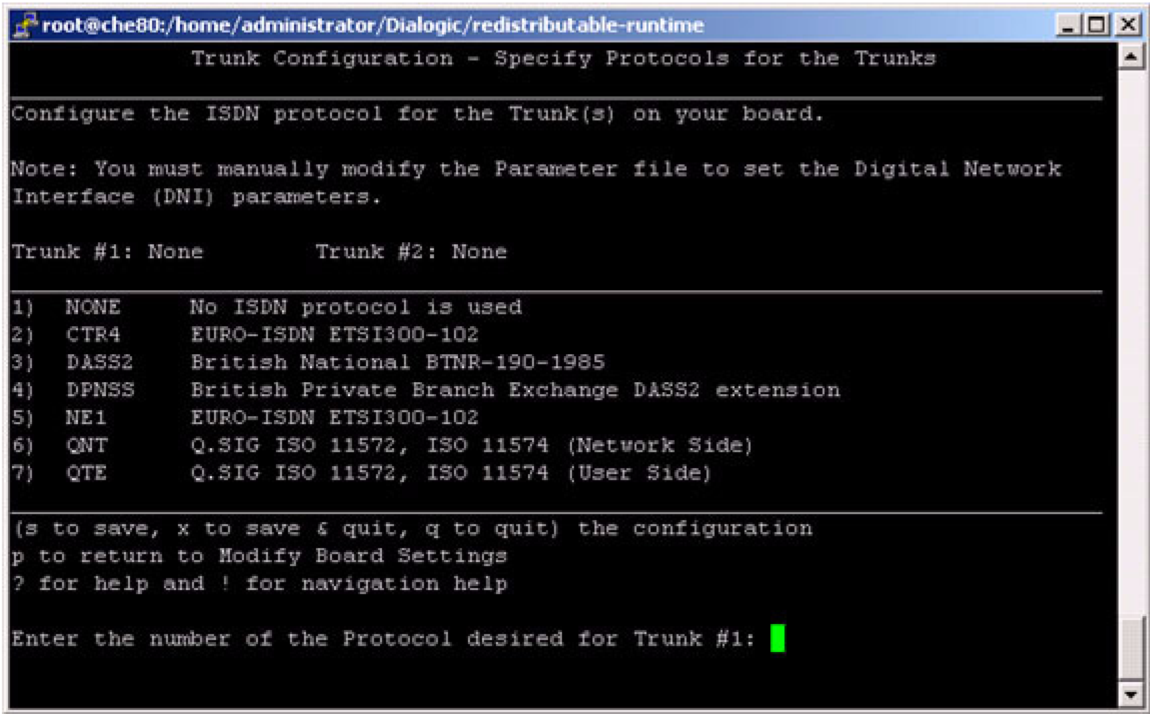
Dialogic Board Summary

2. Enter the thumb wheel of the dialogic card to be configured.



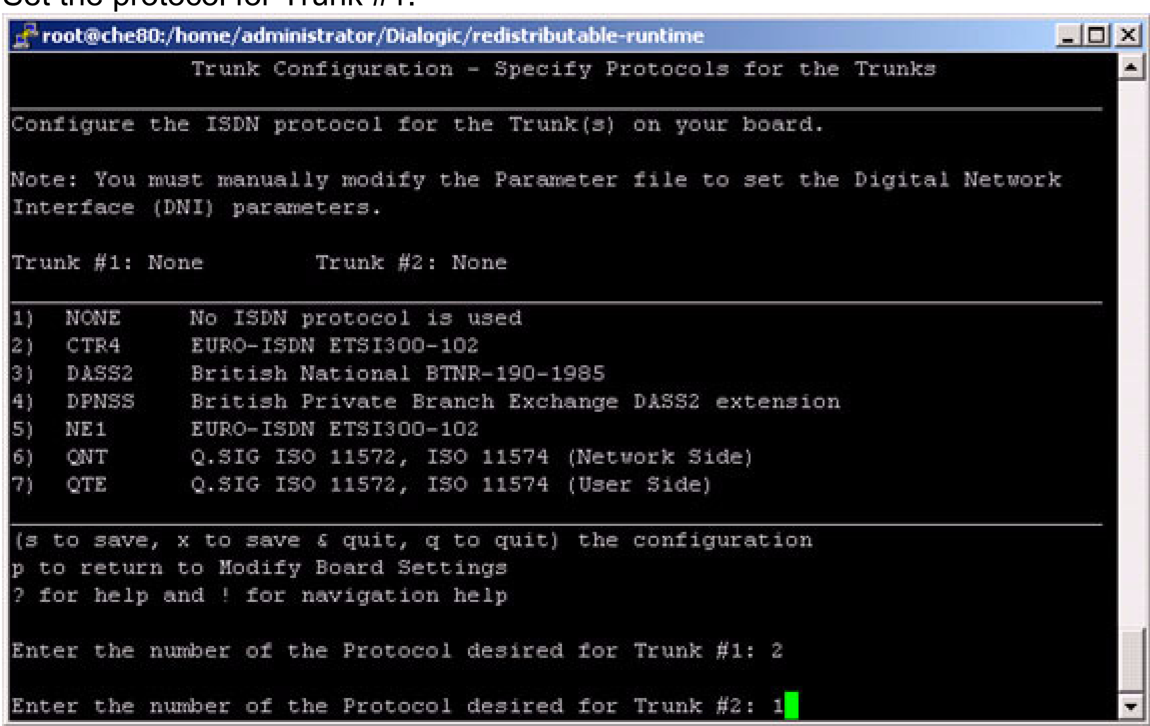
Modify Board Settings

3. Enter `2` to choose Trunk Configuration.



Trunk Configuration - Specify Protocols for the Trunks

4. Set the protocol for Trunk #1.



Trunk Configuration - Specify Protocols for the Trunks

5. Set the Protocol for Trunk #2.

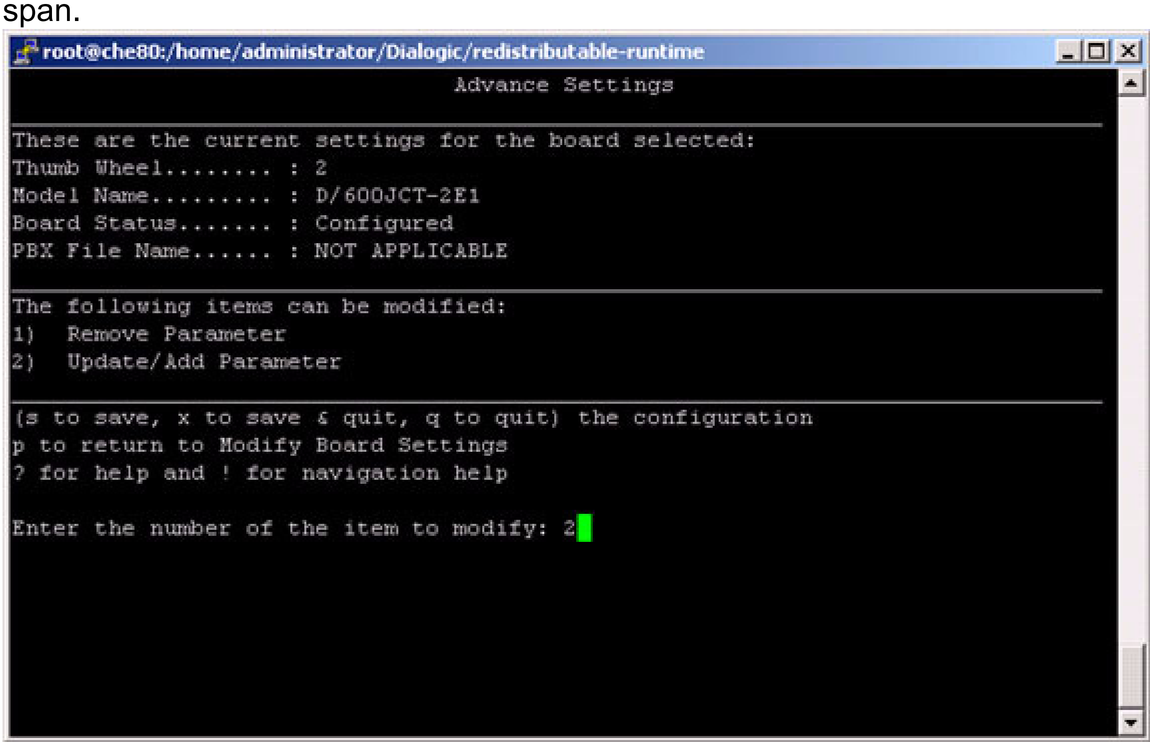
Tip

If the two Trunks are intended for call-handling and no ecstream support is required by ASR applications, select the appropriate protocol from the numbers `2-7`.

If the two Trunks will be required to support ASR applications for which ecstream support by PSTNC is mandatory, set the protocol NONE (by entering `1`).

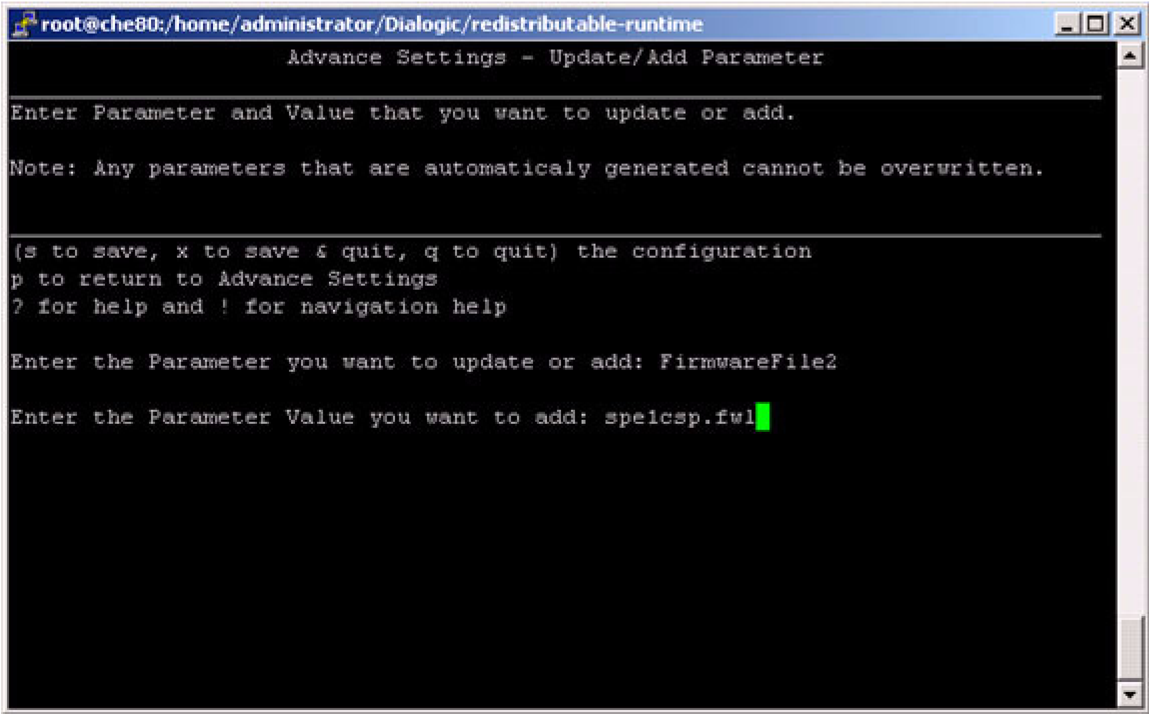
The configuration wizard returns to the previous screen.

6. Select `5` for Advanced settings, if the ecstream support is desired on the second span.



Advanced Settings

7. Enter `2`, to add a parameter for setting ecstream support.



Advanced Settings - Update/Add Parameter

8. Set the parameter `FirmwareFile2` and its value.
9. Enter `s` to save the configuration changes.
10. (Optional) Press `p` to return to the previous screen and set any other parameters, as necessary.
11. Save the changes and exit the configuration.
12. Start Dialogic one of these two ways: `/etc/init.d/ct_intel start` OR `<Dialogic bin dir>/dlstart`

Deploying Multiple Media Control Platforms

This topic describes how to deploy multiple instances of the Media Control Platforms on a single server by using Genesys Administrator and by using a manual procedure. It contains the following sections:

- Deployment Using Genesys Administrator
- Manual Deployment

Deployment Using Genesys Administrator

Before you begin to provision the Media Control Platform Applications, see the Task Summary: Preparing Your Environment for GVP.

The following Task Summary contains a list of tasks that are required to deploy and provision multiple instances of the Media Control Platform on a single server by using Genesys Administrator, and includes links to detailed information that is required to complete these tasks.

Task Summary: Deploying Multiple MCP Applications by Using Genesys Administrator

Objective	Related Procedures and Actions
Configure the host(s)	1. Configure a new host in the Configuration Database for the computer that is hosting the Media Control Platform instances. See Procedure: Configuring a Host in Genesys Administrator.
Install GVP	2. Import the Installation Package for the Media Control Platform into the Genesys Administrator Repository by using the Single Installation Package method. See Procedure: Importing the Installation Packages into the Repository. 3. Use the Genesys Deployment Wizard to install the Media Control Platform Applications with basic configuration, and repeat this procedure for each Application you are installing. For each Application: ◦ Enter a different Application Name. ◦ Select a different Working Directory. See Procedure: Using the Deployment Wizard to Install GVP.
Configure the Applications	4. Use the Genesys Media Control Platform Configuration Wizard to resolve any port conflicts and maximize the performance of each Media Control Platform Application. See Procedure: Using the Media Control Platform Configuration Wizard.
Start the Applications	5. Configure the Media Control Platform Application objects to start automatically.

	See Starting and Stopping the Components.
Complete the post-installation activities	6. Configure the Media Control Platform Application objects for the functionality that you want use in your deployment. See Task Summary: Post-Installation Configuration of GVP.

Procedure: Using the Media Control Platform Configuration Wizard

Provision multiple Media Control Platform Applications on a single server by modifying default TCP/IP configuration to resolve any conflicts.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, click **Environment > Hosts**.
3. In the Hosts pane, select the host that has the Media Control Platform Applications that you want to configure.
4. In the Tasks pane, click **Configure MCP network parameters**. The Media Control Platform Configuration Wizard Welcome page appears including a configuration summary indicating the validation status of the Media Control Platform Applications on the selected host.

> **Tip**
>
> Alternatively, you can select a Media Control Platform Application in the Environment > Applications pane. The wizard does not appear in the Tasks pane until either a Host or Application is selected.

5. On the Local SIP Ports page:
    - In the SIP Port field, enter the SIP port numbers.
    - In the Secure SIP Port field, enter the SIPS port numbers.
6. On the MRCPv2 Ports page:
    - In the Client SIP Port field, enter the SIP port number for the client.
    - In the Lower Media Port field, enter the lower port range boundary for the dynamically-allocated media ports.

    - In the Upper Media Port field, enter the upper port range boundary for the dynamically-allocated media ports.
7. On the MRCPv1 Ports page:
    - In the Lower Media Port field, enter the lower port range boundary for the dynamically-allocated media ports.
    - In the Upper Media Port field, enter the upper port range boundary for the dynamically-allocated media ports.
8. On the Remdial Ports page, enter the Remdial port number.
9. On the Local RTSP Ports page:
    - In the Lower RTSP Port field, enter the lower port range boundary for the dynamically-allocated media-streaming RTSP ports.
    - In the Upper RTSP Port field, enter the upper port range boundary for the dynamically-allocated media-streaming RTSP ports.
10. On the Local RTP Ports page:
    - In the Lower RTP Port field, enter the lower port range boundary for the dynamically-allocated RTP ports.
    - In the Upper RTP Port field, enter the upper port range boundary for the dynamically-allocated RTP ports.
11. On the Debugger Ports page:
    - In the Local Debugger Port field, enter the local debugger port number.
    - In the Public Server Host field, enter the host name for the public server.
    - In the Public Debugger Port field, enter the public debugger port number.
12. Click **Finish**.
13. Configure the Media Control Platform Application objects to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
14. Complete the Post-Installation activities on the Media Control Platform Application objects. See Configuring the GVP Components.

Manual Deployment

Before you begin to provision the Media Control Platform Applications, see the Task Summary: Preparing Your Environment for GVP (Windows) or Task Summary: Preparing Your Environment for GVP (Linux).

The Task Summary below contains a list of tasks that are required to manually deploy multiple instances of the Media Control Platform on a single server (Windows or Linux) and includes links to detailed information that is required to complete these tasks.

Task Summary: Manually Deploying Multiple Media Control Platforms

Objective	Related Procedures and Actions
Configure the host(s)	1. Configure a new host in the Configuration Database for the computer that is hosting the Media Control Platform instances.

	See Procedure: Configuring a Host in Genesys Administrator.
Complete the Application objects	2. Create the GVP Application objects: a. Import the template. Use the same template for each of the Media Control Platform Applications you are installing. See Procedure: Importing Application Object Templates Manually. b. Create the Application objects. Create a separate Application object with a different name for each instance of the Media Control Platform you are installing. See Procedure: Creating Application Objects Manually.
Install GVP	3. Install the Media Control Platform Applications by repeating the installation procedure for each Application that you are installing. For each Application: ◦ Enter a different Application Name. ◦ Select a different Working Directory. See Manually Installing GVP on Windows or Manually Installing GVP on Linux.
Configure the Applications	4. Use the Media Control Platform Configuration Wizard to resolve any port conflicts and maximize the performance of each Media Control Platform Application. See Procedure: Using the Media Control Platform Configuration Wizard. Tip To avoid user error and configure all of the Applications at once, Genesys recommends that you

	use the Media Control Platform Configuration Wizard to configure the Applications after the installation is completed. A manual configuration procedure is also available. See Procedure: Manually Configuring Multiple Applications on a Single Server.
Start the components	5. Start the components manually (or configure the components to start automatically). See Startup Sequence for the VPS and Starting and Stopping the Components.
Complete the post-installation activities	6. Configure the GVP components for the functionality that you want use in your deployment. Task Summary: Post-Installation Configuration of GVP.

Procedure: Manually Configuring Multiple Applications on a Single Server

Repeat this procedure for each Application object that you have installed on the server, to manually configure the Media Control Platform Application objects to resolve port conflicts and maximize performance.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, click **Environment > Applications**.
3. Select the Media Control Platform Application that you want to configure.
   The Configuration tab appears.
4. Click the Options tab and change the values of the options in the table below incrementally in each Application object.

Table: Options Media Control Platform

Section	Option	Value
`mpc`	`rtp.portrange`	Default: `10000` start/mins

		Offset each Application by at least `10000`.
`mrcpv2client`	`sip.transport.0` `sip.localport`	Default: `7080` Offset the value for each Application by at least `1`.
`remdial`	`port`	Default: `6999` Offset the value for each Application by at least `1`.

`sip`	`localport`
	`localsecureport`
	`transport.0`
	`transport.1`
	`transport.2`
		Default: `5070/5071`
		
		Offset the value for each Application by at least `2`.

`stack`	`connection.portrange`	Default: `12345-13345` Offset the value for each Application by at least `1000`.
`vrm`	`client.mrcpv2.portrange`	Default: `12345-13345` Offset the value for each Application by at least `1000`.
`vxmli`	`debug.server.port` `debug.server.port.public`	Default: `27666` Offset the value for each Application by at least `1`.

5. Save the configuration.

6. Configure the Media Control Platform Application objects to start automatically. See Procedure: Configuring Application Objects to Start Automatically.
7. Complete the Post-Installation activities on the Media Control Platform Application objects. See Configuring the GVP Components.

Deploying GVP Multi-Site Environments

This topic describes deployment solutions and configurations for Genesys Voice Platform (GVP) in multi-site or large scale environments. It contains the following sections:

- Overview
- Sites and Segments
- Synchronization Across Sites
- GVP Multi-Site Reporting
- Deploying Disaster Recovery Sites

Overview

To ensure your Genesys Voice Platform multi-site environment is secure and functioning efficiently, consider the following key factors:

- A solution that requires the use of a virtual IP address (VIP) across WAN is likely unacceptable, since the use of virtual IPs is better suited to LAN environments.
- In a WAN environment, sites are interconnected by individual links between each site. These individual links can fail and create islands of sites even though locally the site could still be operational.
- Policy enforcement must be consistent across all sites.
- Reporting functions must be consistent across all sites and must be customized to provide individual site reports and multi-site or overall environment reports.
- Resource sharing must be enabled between sites to mitigate failures or if spill-over traffic occurs.
- SIP Server instances within the same site can use all of GVP's available resources within the site.
- A Resource Manager HA configuration is required for multi-site, when integrated with SIP-S.
- In an environment with multi-site deployments, each Resource Manager pair requires a matching pair of Report Servers.

Consider the key factors described in this section when you are planning a multi-site deployment.

Scalability

A single GVP site typically includes a Resource Manager instance (or a High Availability [HA] pair), a Reporting Server instance (or an HA pair), and a pool of Media Control Platform instances. Within a single site, scalability is typically limited by the number of call attempts per second (CAPS) that the Reporting Server can support.

However, scalability can occur across multiple physical sites, enabling policy enforcement, resource sharing, and reporting across multiple sites. In addition, historical and real-time reports can be filtered to generate site reports (by using site identification) or system-wide reports (no site identification).

Multi-Tenancy

In hosted environments where multiple tenants are deployed, GVP can be deployed across multiple sites and media services for different tenants can be serviced by any one of the sites. There are two requirements for this type of deployment:

- Enforcement of tenant policies must be applied consistently across all the sites. Usage limits for a tenant must be applied globally across all the sites at all times. For example, if a tenant has a usage limit of 100, the maximum number of concurrent calls that can be serviced across all the sites at all times is 100.
- Collection of operational data, such as peak and summary usage, must be aggregated correctly across all the sites and can be filtered on a single-site basis. This applies to both historical and real-time reporting.

Disaster Recovery

Implementation of a disaster recovery (DR) plan is critical in multi-site deployments. It means that one site in the multi-site deployment is designated as the DR site and is enabled and ready to be fully functional in the event that any given site is out-of-service, even if it is out for an extended period of time. In addition, operational data must be replicated to the disaster recovery site.

A DR site deployment enables:

- Servicing of incoming requests at full capacity.
- Access to and reporting of data from the failed site.

For more information about DR sites, see Deploying GVP Multi-Site Environments.

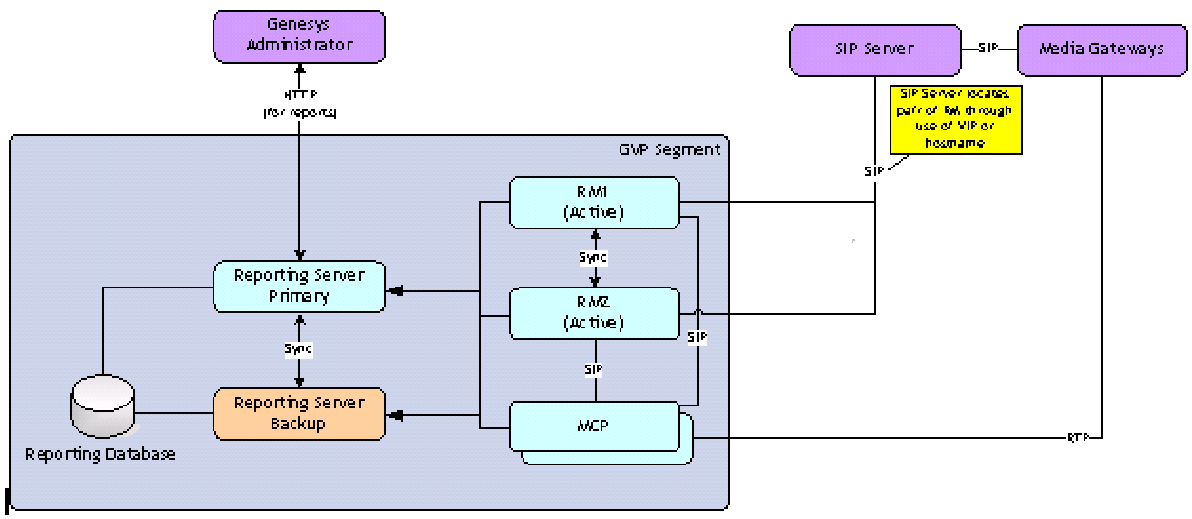
Sites and Segments

The Reporting Server writes reporting data to the database, which provides single-site and overall system reports, and can prove to be a bottleneck, based on known performance metrics. Therefore, the core design of the solution that is proposed in this section is scaled upward by the deployment of multiple sets of the Reporting Server and multiple sets of the database storage units.

GVP Segment Defined

A GVP segment can be defined as a logical grouping of core components, such as Resource Manager, Reporting Server, and logical resource groups (LRG) that include Media Control Platforms, Call Control Platforms, or CTI Connectors. The figure below depicts the high-level relationship between the core GVP components.

The components in a segment must be deployed locally in the same site. However, one or more segments can be deployed within a site. Genesys Administrator provides a mechanism to identify and display the segments and sites. In addition, when resource groups are created, the user can choose the segment to which the resource group will belong, rather than the Resource Manager to which it will belong.

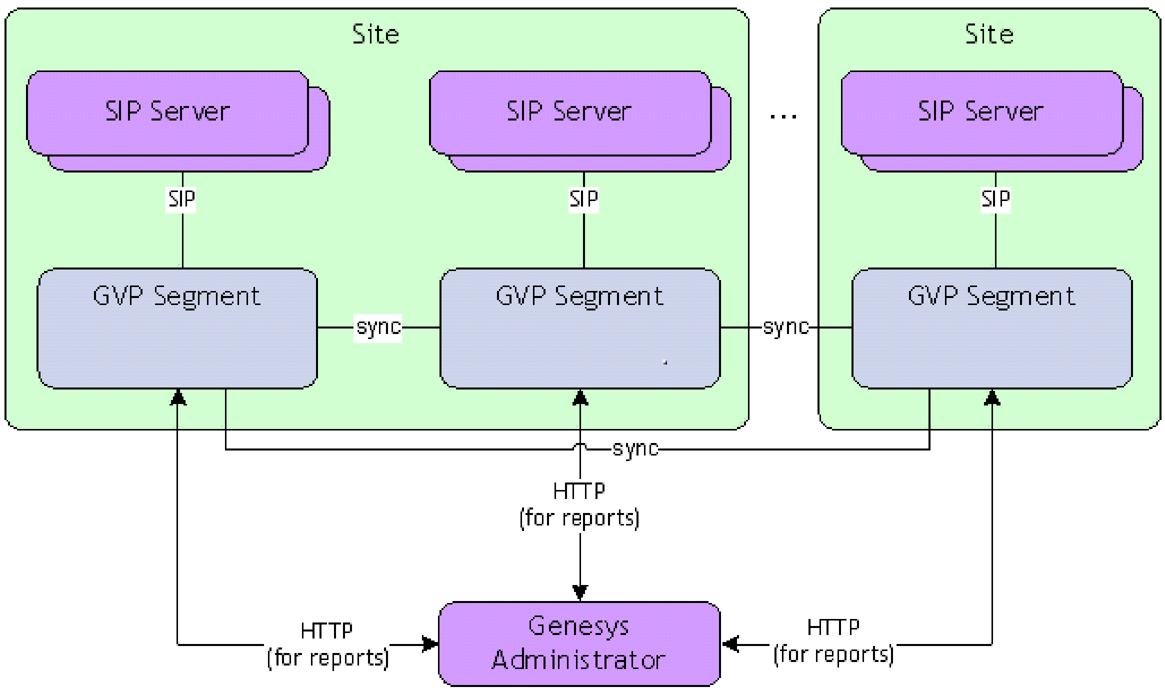


Relationship Between GVP Core Components

Within the segment, one or more SIP Servers can connect to a pair of Resource Manager instances to access media services or through SIP communication, IVR Server. In addition, Genesys Administrator can retrieve historical and real-time reports through the Reporting Server.

The Resource Manager and the Media Control Platform contribute to the events that are logged to the reporting database by the Reporting Server. The Resource Manager and Media Control Platform generate operational data, such as summary and peak information and send it to the Reporting Server.

To make a single site scalable, multiple segments can be deployed within the site. To make multiple sites scalable, segments can be deployed across multiple sites. In this solution, all the segments are considered to be working together as a single large deployment, regardless of the boundaries between them. The illustration below shows an example of how the multiple segments can apply to both a single site or multiple sites.



Segmentation of Single or Multiple Sites

In this solution, three elements enable the GVP segments to work together as a large deployment.

1. Certain components synchronize with all other segments. This is a higher-level synchronization than the synchronization that occurs locally, such as the HA synchronization between two active Resource Manager instances or two Reporting Server instances. In other words, local HA synchronization is designed to ensure continuity of operations for the same component, while synchronization across sites is for elements or counters that are globally shared. In this solution, only the Resource Manager (for policy enforcement) and the Reporting Server (for historical and real-time reporting) require synchronization across sites.

2. The Resource Manager monitors the local segment to determine if the media servers are able to handle all incoming requests or if their capacity is exhausted. If this happens and resource sharing is enabled, the Resource Manager can forward the SIP requests to another segment.
3. Genesys Administrator is the GUI from which aggregated real-time and historical reports from all the Reporting Server instances across all segments can be extracted. All segments in the deployment have a site identifier, which Genesys Administrator uses to aggregate specific reports. The site identifier enables users to generate reports on a per-site basis. In addition, multiple segments can have the same site identifier.

Synchronization Across Sites

This section provides information about how site synchronization, policy enforcement, and resource sharing occurs across GVP multi-site environments. It also describes segment and network recovery after a failure.

Site Policy Enforcement

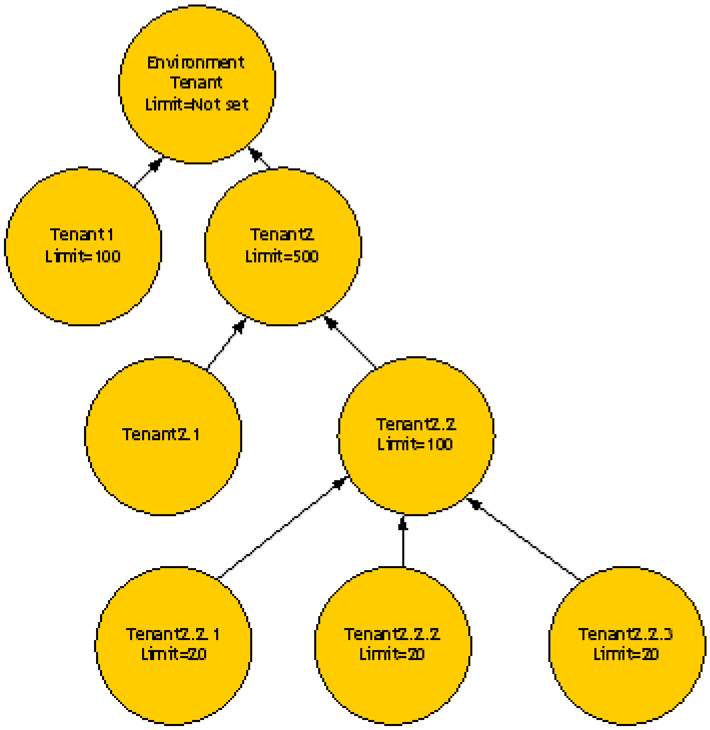
In a multi-site environment, policies are categorized as static or dynamic to help the user understand which policies are dynamically enforced. The Resource Manager enforces mostly static policies, which means it reads the value from Configuration Server to determine how the policy is enforced.

The Resource Manager tracks the current usage limits in memory for dynamic policies. To accurately enforce the policy usage limit for each tenant and IVR Profile and ensure all calls are accounted for, the usage limit is routed through the Resource Manager.

Usage Limit Counters

After the call is established, the Resource Manager stays in the SIP messaging path, so that it can track the time of call termination. The Resource Manager keeps a counter for each tenant and IVR Profile, based on the current usage.

In distributed environments, where the same counter is tracked system-wide, each usage limit counter is subdivided into smaller units and shared with other segments so that each segment can locally track each counter. This process ensures the accuracy of each local counter. For example, in a deployment that has 3 segments and a tenant with a usage limit of 100, the counter for the segments can be subdivided into 40, 40, and 20. `INSERT_TEXT` The figure below provides an example of a tenant hierarchy and its usage limits.



Tenant Hierarchy and Usage Limits

Role of the Coordinator

The Resource Manager maintains a synchronized connection with other Resource Manager peer instances across multiple segments and uses an election algorithm to select a coordinator. The coordinator's role is to assign usage limit values to each segment for dynamic policies, by dividing each usage limit counter for each tenant. The values are assigned, based on a weight parameter, which defaults to 100 if it is not set.

Election Algorithm

To simplify the election algorithm, each segment is assigned a segment identifier or number. When the Resource Managers (including the active pairs) connect to each other through the synchronization port, they assign a coordinator to divide the counters, based on the segment identifier (together with the local Resource Manager identifier). The sequence of events occurs as follows:

1. A Resource Manager instance (RM) broadcasts an election message to all other connections, which includes the segment identifier.

2. If RM does not hear from other connections within a certain amount of time (for example, 5 seconds), then RM declares victory and broadcasts itself as the coordinator.
3. If RM hears from another Resource Manager instance with a lower identifier, RM waits for the timeout to expire and then listens for the broadcast result. If RM does not hear the result within the timeout, then RM sends the election message again (see Step 1).
4. At the end of the election algorithm, the coordinator broadcasts another message with the division of the counters as the result. By default, the counters are divided evenly among the segments. See the example in the table below.
5. After the counters are divided, the Resource Manager pairs for each segment enforce the usage limits locally.

Table: Division of Counters Among Segments

Tenant	Segment 1 (coordinator)	Segment 2	Segment 3
Environment	Not set	Not set	Not set
T1	34	33	33
T2	167	167	166
T2.1	Not set	Not set	Not set
T2.2	34	33	33
T2.2.1	7	6	6
T2.2.2	7	6	6
T2.2.3	7	6	6

Customizing Division of Counters

The division of counters can be customized by adding a weight value to each segment. Each segment has a weight of 100 by default. The coordinator reads the weight values and uses them to divide the counters. The coordinator adds up all the weight values for all segments, giving counters to each segment in proportion to its own weight. For example, if segments 1, 2, and 3 have weights 300, 100, 50, respectively, segment 1 is assigned 300/450 (or 2/3) of the counters. the table below contains the segments from the table above with adjusted weights factored into the equation.

Table: Customized Division of Counters With Weights

Tenant	Segment 1 coordinator (weight 300)	Segment 2 (weight 100)	Segment 3 (weight 50)

Environment	Not set	Not set	Not set
T1	67	22	11
T2	334	111	55
T2.1	Not set	Not set	Not set
T2.2	67	22	11
T2.2.1	14	4	2
T2.2.2	14	4	2
T2.2.3	14	4	2

Segment Failure

If a Resource Manager instance detects a peer segment failure (both Resource Manager instances within the segment fail), the surviving segments re-issue the election process to elect a new coordinator. The new coordinator then distributes the counters, recalculated with the remaining weights, among the remaining segments after the election algorithm is complete. In the table below, see how the counters are divided when segment 3 fails and segment 1 is elected as the coordinator.

Table: Division of Counters With Weights After Re-Election

Tenant	Segment 1 coordinator (weight = 325)	Segment 2 (weight =125)	Segment 3 (down)
Environment	Not set	Not set	--
T1	75	25	--
T2	375	125	--
T2.1	Not set	Not set	--
T2.2	75	25	--
T2.2.1	15	5	--
T2.2.2	15	5	--
T2.2.3	15	5	--

Segment Recovery

When a Resource Manager instance recovers and re-joins the system, no action is taken if the instance belongs to an active segment. When a Resource Manager instance from a failed segment recovers and re-joins the system, the existing segments re-issue the election process to elect a new coordinator. The new coordinator distributes the counters among the remaining segments, recalculating them with the remaining weights.

New Segments Joining

If a new segment joins the system, the counters are further sub-divided among the segments. It is possible for some segments to have more existing calls than the new usage limit. The Resource Manager does not attempt to drop calls because of the new (and lower) limit, but allows over-usage temporarily.

> Tip
>
> When the Resource Manager allows over-usage temporarily, any new incoming calls are rejected until a sufficient number of existing calls drop, and bring the current usage lower than the new usage limit.

Network Disconnections

> Tip
>
> When the Resource Manager allows over-usage temporarily, any new incoming calls are rejected until a sufficient number of existing calls drop, and bring the current usage lower than the new usage limit.

A WAN link failure can create islands of segments that can operate independently. A disconnection from the WAN link is treated as a disconnected segment by the surviving segments. The separate islands independently issue the election process to find a new coordinator. When this happens, the system has two full sets of usage limits because the islands do not see each other. The table below provides results when Site A and B are disconnected from each other.

Table 44: Island Segments After a WAN Link Failure

	Site A (Island 1)		Site B (Island 2)
Tenant	Segment 1 coordinator (weight = 300)	Segment 2 (weight = 100)	Segment 3 (weight = 50)
Environment	Not set	Not set	Not set
T1	75	25	100
T2	375	125	500
T2.1	Not set	Not set	Not set
T2.2	75	25	100
T2.2.1	15	5	20
T2.2.2	15	5	20
T2.2.3	15	5	20

Network Recovery

When the network is recovered, the segments in the system are re-connected. The segments issue another election process to find a new coordinator. Similar to segment recovery, some segments might end up with more existing calls than the new usage limit. The Resource Manager does not attempt to drop calls because of the new (and lower) limit, but allows over-usage temporarily.`INSERT_TEXT`

> **Tip**
>
> When the Resource Manager allows over-usage temporarily, any new incoming calls are rejected until a sufficient number of existing calls drop, and bring the current usage lower than the new usage limit.

GVP Multi-Site Reporting

In GVP multi-site environments, Reporting Server instances in each GVP segment collect data. Genesys Administrator queries the Reporting Server instances to generate historical and real-time reports on a per-site or system-wide basis. Genesys Administrator can access site information from Configuration Server by checking the gvp.site folder in the Annex section of the `Provisioning > Environment > Applications` folder. It reads the site information at the user session logon and retains it throughout the session.

The following reports can be aggregated to provide system-wide data:

- IVR Profile Call Arrival
- Component Call Arrival
- Tenant Call Arrival
- VAR IVR Action Usage
- ASR/TTS Usage
- All VAR Summary reports

Genesys Administrator Reporting Interface

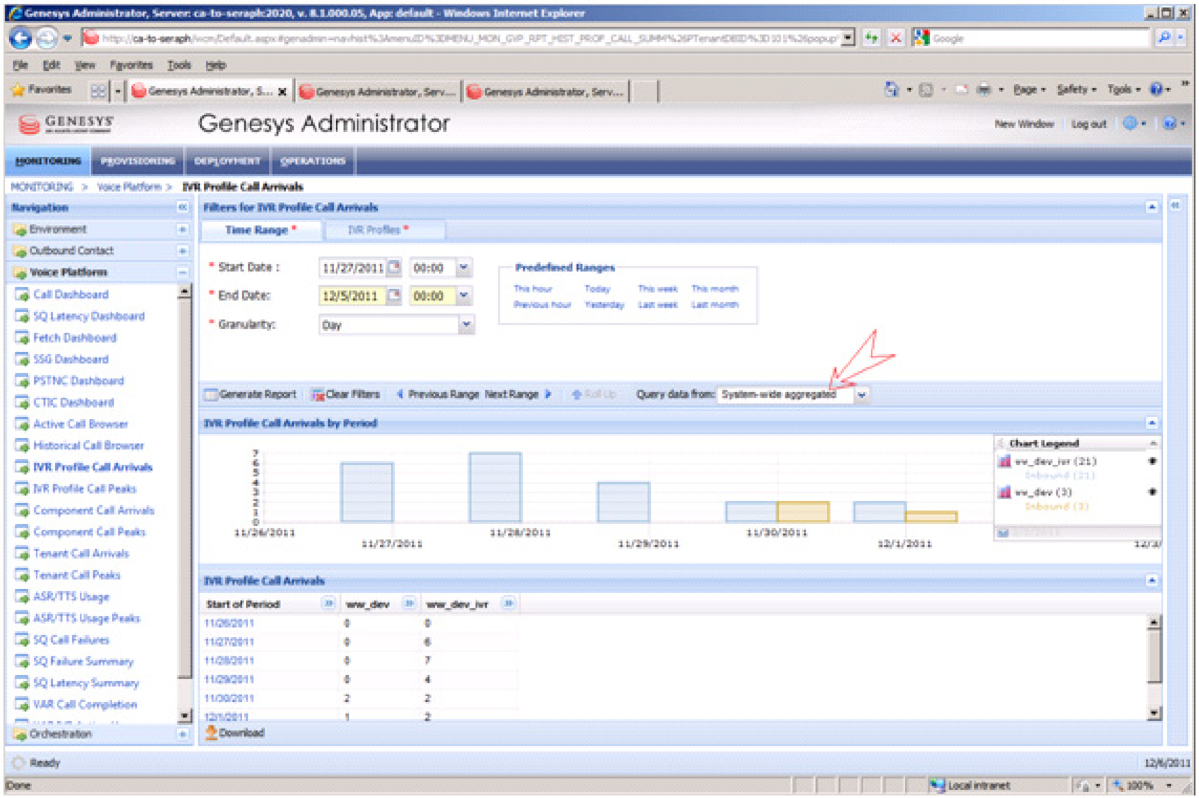
This section describes the additional menu selections that are displayed on Genesys Administrator's Monitoring tab when GVP is deployed in a multi-site configuration.

Operational Reports

IVR Profile Call Arrivals, Tenant Call Arrivals, and ASR/TTS Usage Reports

On the **Generate Report** bar of these report pages of the GUI, you can choose a site from a drop-down list. This drop-down list is displayed when there are multiple sites configured in Configuration Server, whether a Reporting Server is present or not in Genesys Administrator's **Application Connections** settings.

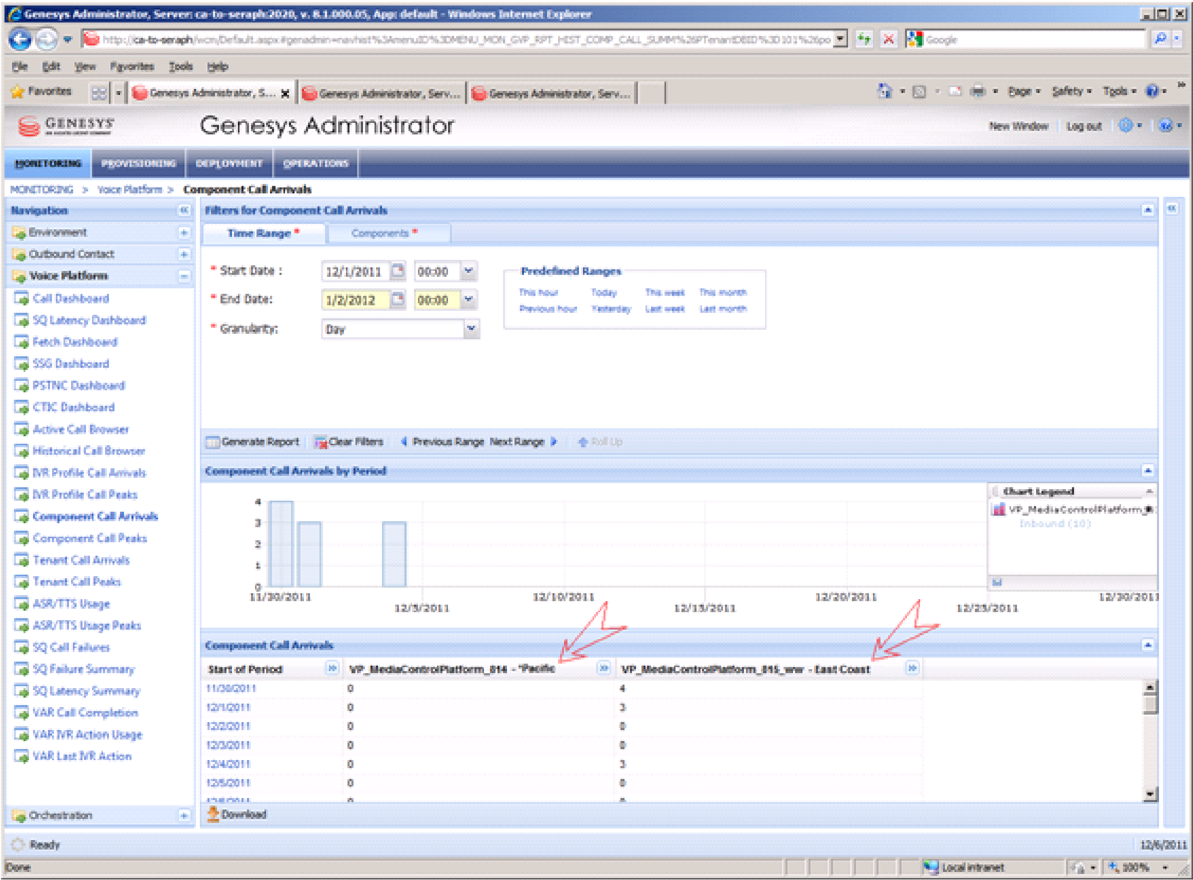
The default selection is `All-Sites`. When this option is selected, the arrivals data must be summarized across all sites on the selected IVR Profiles.



Query Date From Field Genesys Administrator

Component Call Arrivals and ASR/TTS Usage Reports

On the Component Call Arrivals section of these report pages in the GUI, the site name is appended to the component name to indicate the site to which the component belongs. See the figure below.



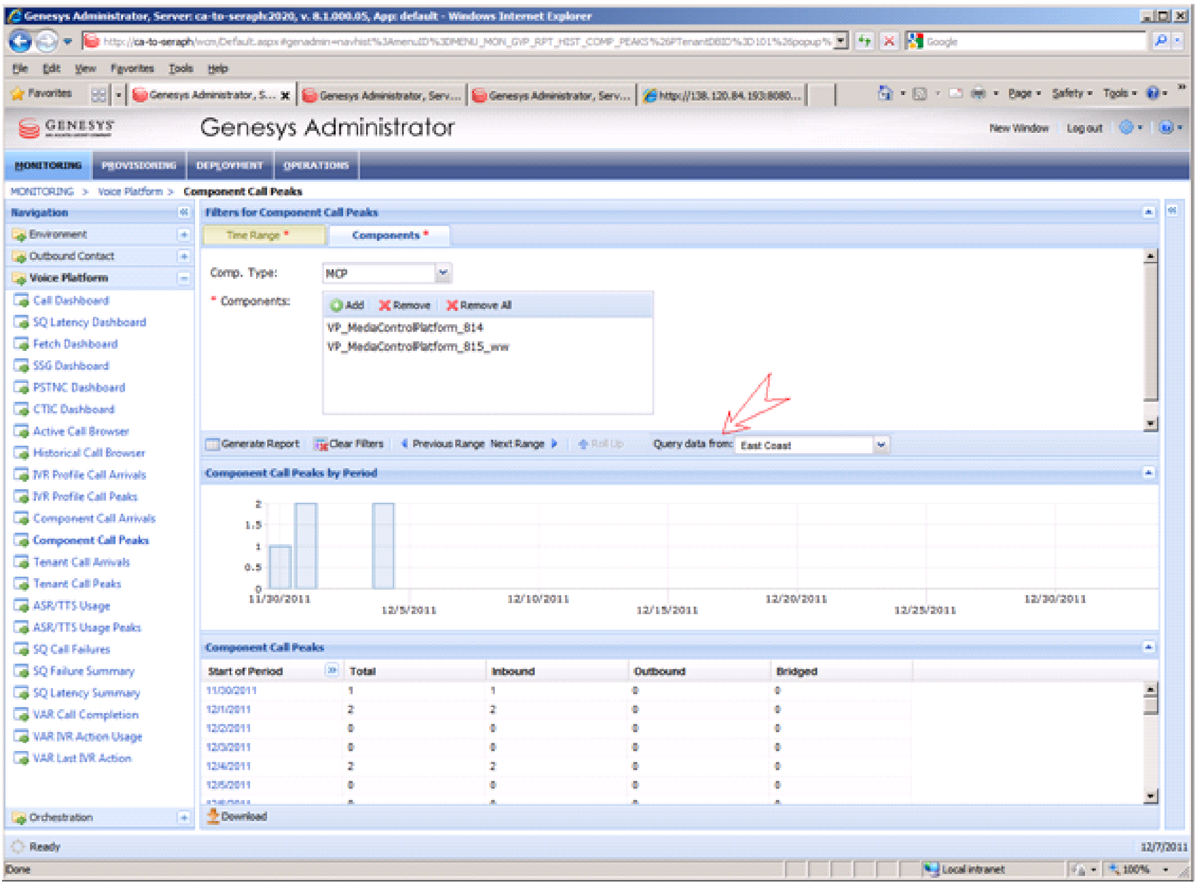
Component Call Arrival Genesys Administrator

IVR Profile Call Peaks, Component Call Peaks, Tenant Call Peaks, and ASR/TTS Usage Peaks Reports

On the **Generate Report** bar of this report GUI, you can choose a site from a drop-down list. The drop-down list does not include the `All-Sites` option.

The default selection is the first site that appears at the top of the drop-down list. See the figure below.

> Tip
>
> Currently, Peaks reports display only one selection.



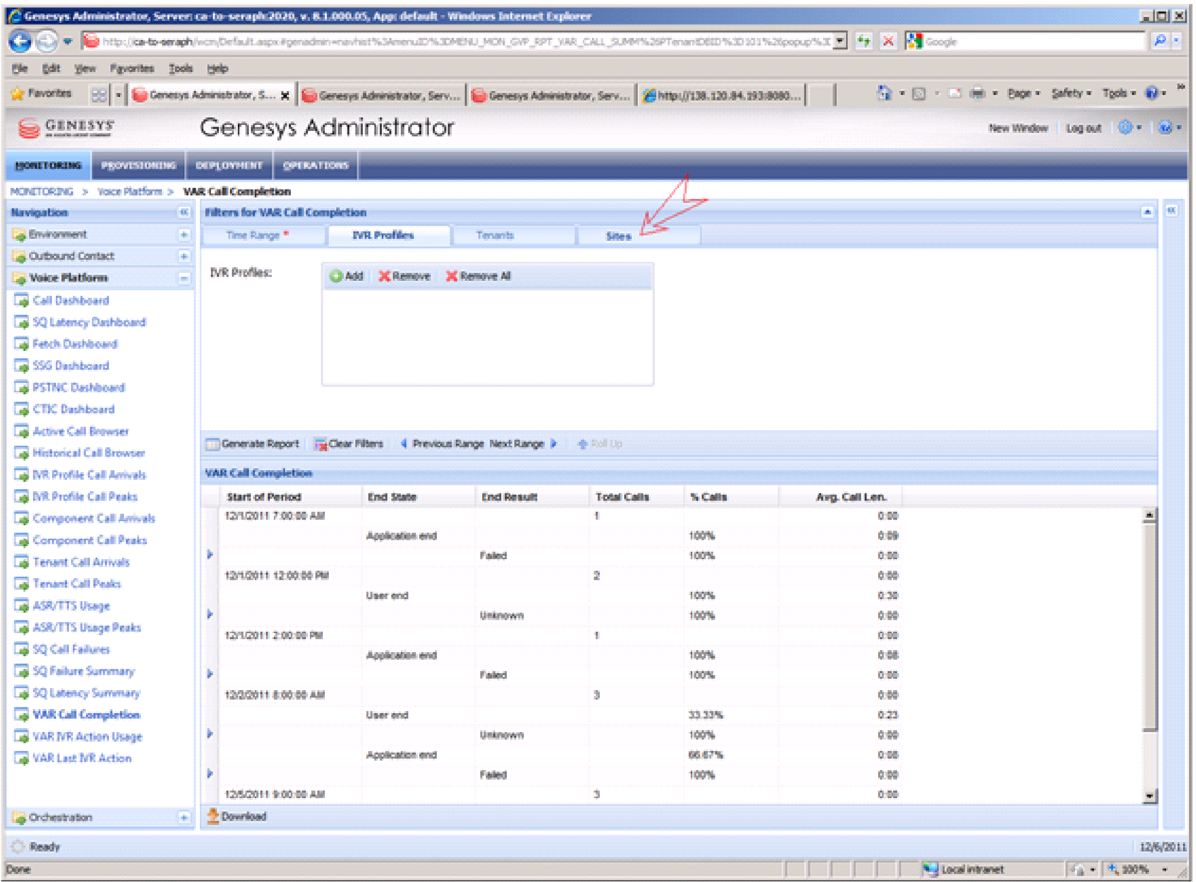
Component Call Peaks Genesys Administrator

VAR Reports

VAR Call Completion Report

On the **Filters for VAR Call Completion** section of this report page of the GUI, you can choose from a selection of sites from a drop-down list. See the figure below.

The default selection is `All-Sites`. When this option is selected, data is merged from the values that are returned from multiple queries.

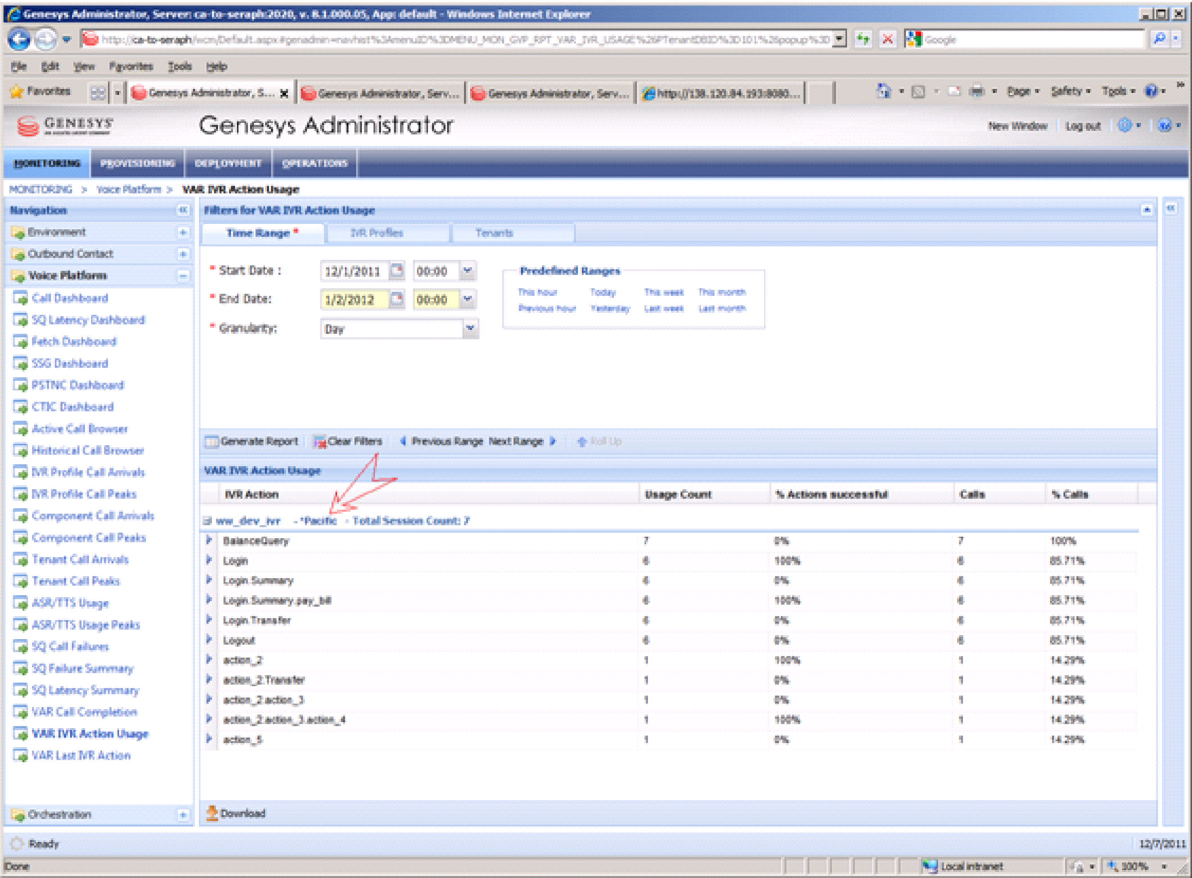


VAR Call Completion Genesys Administrator

VAR IVR Action Usage Report

On the **VAR IVR Action Usage** section of this report page in the GUI, you can choose from a selection of sites from a drop-down list. See the figure below.

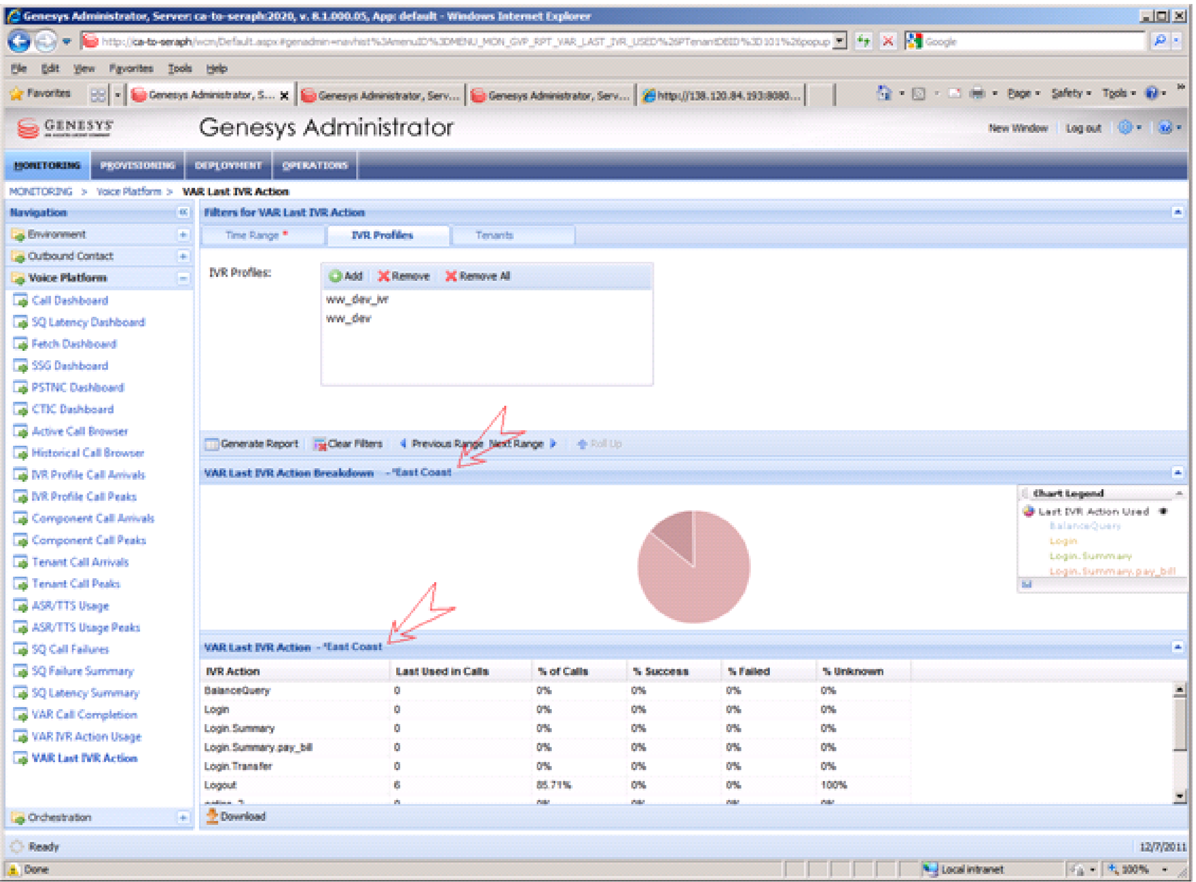
The default selection is `All-Sites`.



VAR IVR Action Usage Genesys Administrator

On the **VAR Last IVR Action Breakdown** and **VAR Last IVR Action** sections of this report page in the GUI, you can choose from a selection of sites from a drop-down list. See the figure below.

The default selection is `All-Sites`.



VAR Last IVR Action Genesys Administrator

Deploying Disaster Recovery Sites

A Disaster Recovery (DR) deployment is basically organized as two regular segments, with the capacity to replicate reporting data between them to ensure operational reports are available when a segment fails. **Structure of a Disaster Recovery Segment** (below) depicts the structure of a DR segment.

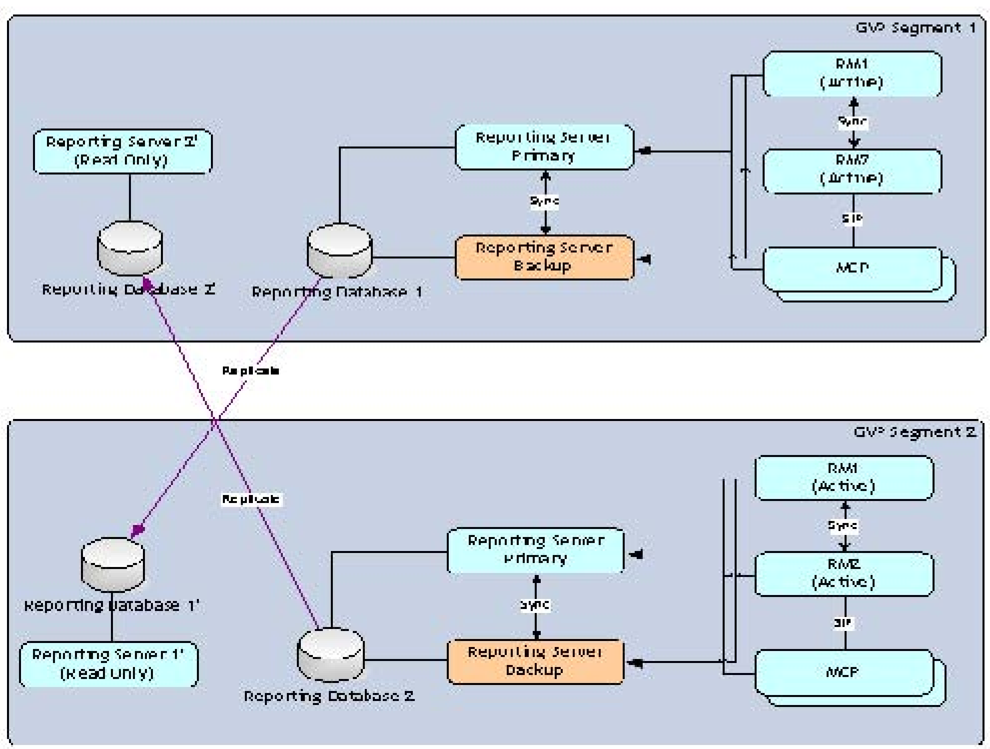
The Disaster Recovery deployment consists of a Primary (or hot) site and a DR (or cold) site. The Reporting Server instance at the Primary site is configured to accept incoming calls. Its `rs.histonly.enabled` configuration option is set to `false`.

- The Reporting Server instance at the DR site is configured so that it does not accept incoming calls until another DR site is no longer reachable. Its `rs.histonly.enabled` configuration option is set to `true`.

To deploy the DR site, create a replicated reporting database and a read-only Reporting Server instance so that the data is replicated during runtime. This ensures that the replicated instance has the most recent data available at all times.

Only the Primary sites are included in the **Site** drop-down list and **All Sites** queries in Genesys Administrator.

> Tip
>
> In this release, the Primary (hot site) and DR (cold site) configuration is supported by the Reporting Server only (not Resource Manager).



Structure of a Disaster Recovery Segment

Role of the Read-Only Reporting Server in Disaster Recovery

The role of the read-only Reporting Server is to provide operational reports only. These servers stay active despite the fact that the segment might be a cold DR site, so that operational reports are available in both sites at all times. In this way, Genesys Administrator can automatically select one of the working Reporting Server instances to retrieve operational reports.

Add the DR Reporting Server Connection

Go to the primary Reporting Server's connections tab and add a connection to the DR Reporting Server.

Reporting Data Queries

In DR deployments, Genesys Administrator queries data from the Reporting Server in the following order:

For the historical data:

- HA Primary Reporting Server
- HA Backup Reporting Server (that synchronizes with the HA Primary Reporting Server)
- Read-only DR Reporting Server (that synchronizes with the HA primary Reporting Server)

For real-time data:

- HA Primary Reporting Server
- HA Backup Reporting Server (that synchronizes with the HA Primary Reporting Server)

For aggregate data:

- The queries are repeated *n* times for *n* Primary sites, where *n* is the number of Primary sites. For example, if there are 10 Primary sites, the query is repeated 10 times.

Disaster Recovery Modes of Operation

A DR site or segment can be running in hot or cold mode. A hot DR segment acts as a normal site and participates in the election process to accept incoming calls. A cold DR segment does not process incoming calls and starts as passive. The Resource Manager instances in a cold DR segment still connect to the other segments and participate in the election algorithm. As long as all of the hot DR sites are running and active, the coordinator does not assign usage limits to the cold DR site. If one of the hot DR sites goes down, the cold DR site becomes active and the coordinator assigns usage limits to the (now active) cold DR site.

After a cold DR site becomes active and begins accepting calls, the site does not become passive again until all hot DR sites are back online and active. The coordinator can assign a zero usage limit to a cold DR site, once it determines all hot DR sites are back online. The cold DR site can then become passive again.

Configuring GVP

Following installation, some Genesys Voice Platform (GVP) components require additional configuration to initiate the advanced features and optimize operation. Starting here: read about post-installation activities for the GVP hosts, as well as how to create the database and schema for the Reporting Server.

- Configuration Task Summary
- Configuring the GVP Components
- Configuring the Reporting Server Database

Task Summary

Task Summary: Post-Installation Configuration of GVP

Objectives	Related Procedures and Actions
Complete the post-installation configuration of GVP components	1. Integrate Application objects. See Procedure: Integrating Application Objects with Resource Manager. 2. Create server connections. See Procedure: Creating a Connection to a Server. 3. Provision and configure Speech Resource objects. See Provisioning Speech Resource Application Objects and Procedure: Assigning the MRCP Server to the Media Control Platform.

Complete the post-installation configuration for the Reporting Server and database	1. **If you are installing GVP 8.1.2 or later on the Media Control Platform host**, use the following steps to configure the grammars:
	a. Change to the root user (`su`).
	b. Add the following lines to the file `/etc/httpd/conf/httpd.conf`:
	`Alias /mcp/ "/var/www/gvp/mcp/"`
	
	`<Directory "/var/www/gvp/mcp/">`
	
	`ExpiresActive On`
	
	`ExpiresDefault "now plus 5 minutes"`
	
	`Options Indexes MultiViews`

```
AllowOverride None

Order allow,deny

Allow from all

</Directory>
```

2. On the Apache Web Server (Linux only), modify the `/etc/mime.types` file. Type: `application/srgs+xml`.
3. Configure the CTI Connector for Cisco ICM Integration. See Procedure: Configuring the CTI Connector for Cisco ICM Integration.
4. Provision the PSTN Connector. See Procedure: Configuring the PSTN Connector, and Procedure: Configuring a Trunk DN for the PSTN Connector.
5. Provision the Supplementary Services Gateway for outbound-call initiation. See Procedure: Provisioning the Supplementary Services Gateway, and Procedure: Configuring DNs for the Supplementary Services Gateway.
6. Create DNs for the Supplementary Services Gateway:
    a. a. Create a Routing Point DN to use for outbound calls with the legacy GVPi (if required). See Procedure: Configuring a Routing Point DN.
    b. Create a VoIP Service DN to initiate MSML dialogs (if required). See Procedure: Configuring a Voice Over IP Service DN.
    c. Create a Voice Treatment Port (VTP) DN to play IVR Profile VoiceXML dialogs (if required). See Procedure: Configuring a Voice Treatment Ports DN.
7. (Optional) Install and configure security certificates to enable the Supplementary Services Gateway to interact with SIP Server over secure TLS ports. See Chapter 3 in the GVP 8.5 User's Guide.
8. Prepare the Call Control Platform to make a call. See Procedure: Configuring the Call Control Platform.
9. If you are deploying GVP 8.1.2 in a multi-tenant environment, create the child tenants in the hierarchy. To create child tenants manually or import multiple tenants from a file, see Genesys Administrator 8.1 Help.

<table>
<tr>
<td></td>
<td>
10. If you are deploying GVP 8.1.3 or higher, and require Resource Manager High Availability, configure SIP static routing for MCP, CCP and CTI Connector to work with an Resource Manager pair in active active HA mode. Set the [sip]transport.staticroutelist parameter to each Resource Managers IP address. For example, `[sip]transport.staticroutelist=138.120.84.101, 138.120.84.102`. Also: configure the Resource Managers in the same group to listen on the same port number.
11. Group and configure the GVP resources, IVR Profiles, and DIDs for ease of management and administration. See Using Resource Groups.

Tip
Before you begin to plan and configure your GVP resources, there is important information you should know about tenant permissions and assigning DID Groups in multi-tenant environments. See HMT Permissions and Access Rights.

12. Assign and configure the default tenant, and create the default profile. See Assigning Default Tenants and Creating Default Profiles.
</td>
</tr>
<tr>
<td>Complete the post-installation configuration of the Reporting Server and database.</td>
<td>
1. Integrate and configure the Reporting Server. See Procedure: Configuring the Reporting Server User Interfaces.
2. Configure the Locale on Reporting Server (if required). See Procedure: Configuring the Reporting Server Locale.
3. Install the database. See Configuring the Reporting Server Database. For information about the supported databases and versions, see Prerequisites.
</td>
</tr>
</table>

Configuring the GVP Components

Perform these advanced configuration procedures after GVP installation and basic configuration.

Integrating Application Objects

After the Media Control Platform and Call Control Platform Application objects are created and the components are installed, they are integrated with the Resource Manager which acts as a proxy server. SIP devices and VoiceXML or CCXML applications can then make use of media-centric services through the proxy, without having to know the actual location of these resources.

This procedure is optional and required only if you want the Resource Manager to act as a proxy server for outbound requests. To integrate these Application objects with the Resource Manager, you configure the Session Initiation Protocol (SIP) settings.

This procedure describes how to integrate Application objects with the Resource Manager by configuring SIP and secure SIP options.

Procedure: Integrating Application Objects with Resource Manager

Integrate an Application object with Resource Manager by configuring the Application parameters.

1. Verify that all GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP.
2. Log in to Genesys Administrator.

3. On the Provisioning tab, select `Environment > Applications`.
4. Click the Application object that you want to configure for example, the Media Control Platform or Call Control Platform Application.
   The Configuration tab appears.
5. Click the Options tab, and use the View drop-down list to select **Show options in groups...**
6. In the sip section, find the `routeset` option.
7. In the Value field of the `routeset` option, enter the following:
   - `sip:<IP_RM>:<SIPPort_RM>;lr`
     to integrate the Media Control Platform with Resource Manager.
   - `sip:<IP_RM>:<SIPPort_RM>;lr`
     to integrate the Call Control Platform with Resource Manager.

     ...where `IP_RM` is the IP address of the Resource Manager, and `SIPPort_RM` is the SIP port of the Resource Manager typically, `5060`.

     For both MCP & CCP the syntax is the same:
     `routeset = sip:`*`<RM1_IP>`*`:`*`<RM1_proxy_port>`*`;lr`
     for half of the MCPs & CCPs, the other half use
     `routeset = sip:<tt><RM2_IP>:<RM2_proxy_port>`;lr</tt>

     (Where anything not in italics is required verbatim.)
     ...where `proxy_port` is the port from `[proxy] sip.transport.n`

     Tip

     You must include the angle brackets in the Value field in the `sip.routeset` and `sip.securerouteset` parameters.

8. In the Value field of the `securerouteset` option, enter the following:
   - `sip:<IP_RM>:<SIPSecurePort_RM>;lr` to integrate the Media Control Platform with Resources Manager.
   - `sip:<IP_RM>:<SIPSecurePort_RM>` to integrate the Call Control Platform with Resource Manager.
     ...where `IP_RM` is the IP address of the Resource Manager, and `SIPSecurePort_RM` is the SIP secure port of the Resource Manager typically, `5061`.

> Tip
>
> The GVP components and the external SIP Server support secure SIP capabilities.

9. To use the Call Recording Solution through third-party recording servers: In the vrmrecorder section, configure the following options (pointing to the Resource Manager's IP address and SIP port, as shown in the two previous steps:
   - `sip.routeset`
   - `sip.securerouteset`
10. Save the configuration.
11. Create the connections to the Message Server. See Procedure: Creating a Connection to a Server.

Creating a Connection to a Server

Use the procedure in this section to create connections to:

- The Message Server—Create a connection in the Media Control Platform, Call Control Platform, Resource Manager, Supplementary Services Gateway, CTI Connector, PSTN Connector, MRCP Proxy, Reporting Server and Policy Server Applications to ensure that component log information reaches the Log database and can be viewed in the Solution Control Interface (SCI).
- The Reporting Server—Create a connection in the Media Control Platform, Call Control Platform, Resource Manager, PSTN Connector, CTI Connector, Supplementary Services Gateway, and MRCP Proxy Applications to ensure that these components detect the Reporting Server to which they are sending reporting data. Genesys Administrator also requires a connection to Reporting Server to monitor GVP components.
- SIP Server—Create a connection in the Resource Manager, Supplementary Services Gateway, and PSTN Connector Applications to manage the initiation of outbound calls.
- MRCP Proxy—Create a connection in the Media Control Platform Application if you are planning to use the proxy to manage MRCPv1 RTSP traffic within the GVP deployment.
- MRCP Server—Create a connection in the MRCP Proxy Application if you are planning to use the proxy to manage MRCPv1 RTSP traffic within the GVP deployment (in the Media Control Platform Application if you are not deploying the MRCP Proxy).

- Cisco T-server—Create a connection in the UCM Connector Application to ensure the tenant DBID of the Cisco T-Server is included in Request URI in any SIP INVITE messages sent to the UCM Connector.
- The SNMP Master Agent—Create a connection in the Media Control Platform, Call Control Platform, Resource Manager, Supplementary Services Gateway, CTI Connector, PSTN Connector, MRCP Proxy, and Reporting Server Applications if you want to capture alarm and trap information.

Procedure: Creating a Connection to a Server

This procedures creates a connection in an Application object to a server or component.

1. Verify that all GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Click the Application object for which you are creating the connection; for example, the Media Control Platform Application object.
   The Configuration tab appears.
5. In the General section, in the Connections field, click **Add**.
   The Connection Info dialog box appears.
6. In the Server field, click the down arrow to open the Browse Application dialog box.
7. Select the server or component to which you want to create a connection for example, Message Server, SIP Server, or SNMP Master Agent. The required fields in the Connection Info section are populated automatically. (Ensure the Connection Protocols field is left blank. It is not required for GVP components.)
8. Click **OK**.
   The server or component you selected in Step 6 appears under Connections.
9. Save the configuration.
10. Complete the remaining post-installation activities for the Media Control Platform. See Procedure: Provisioning Speech Resource Application Objects.

Provisioning the Speech Resources

The Media Resource Control Protocol (MRCP) speech resources are controlled by the Call Manager Application Program Interface (CMAPI), which opens and closes sessions, and provides the speech recognition and speech synthesis commands that the MRCP Server uses to carry out speech requests.

If the MRCP Proxy is deployed, the configurations in this procedure vary slightly. Therefore, the configurations are described with and without the MRCP Proxy. If you have installed the MRCP Proxy, see also Provisioning the MRCP Proxy.

Tip

The procedures in this section are required only if you are using Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) speech resources, and have an MRCP Server or MRCP Proxy in your deployment.

This section contains two procedures that create the Speech Resource Applications and assign the MRCP Server or MRCP Proxy to the Media Control Platform.

Procedure: Provisioning Speech Resource Application Objects

This procedure creates the MRCP Speech Resource Applications for ASR and TTS. After a Speech Resource Application is created with the basic configuration, it must be provisioned with the IP address and port number of the MRCP Server or the MRCP Proxy (if required).

1. Verify that:
    - The ASR and TTS servers are installed and operational.
    - The MRCP Speech Resource object templates are imported. See Procedure: Importing Application Object Templates Manually.
    - The MRCP Speech Resource objects are created. See Procedure: Creating Application Objects Manually.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Select the Speech Resource Application you want to configure.
   The Configuration tab appears.
5. Click the Options tab, and scroll to the provision section.
6. Enter the value for each Option as described in this table:

Table: MRCP Application Properties Options Tab

Option name	Option value
For MRCPv1	
`vrm.client.resource.name`	Enter the identifier used to link the VoiceXML application to a common set of speech resources; for example, ◦ For ASR, enter ATT, IBM, LOQUENDO, LUMENVOX,

	NUANCE, TELISMA, or ONMOBILE. ◦ For TTS, enter ATT, ACAPELA, VOCALIZER, IBM, or VOXYGEN. **Notes:** ◦ A common set of speech resources means that the provisioning data for each speech resource with the same name is identical. A resource with the same name but different provisioning data should not be added to the common set of resources. ◦ GVP supports dynamically removing and adding ASR/TTS servers (resources), but does *not* support dynamically changing a resource's provisioning data. To change a resource's provisioning data, follow these steps: 1. Remove (delete) the resource. 2. Modify the resource s provisioning data. 3. Add back (reconnect) the resource. **Important:** If the provisioning data of the modified resource is different from an existing common set of resources with the same name, then you must use a different name for this resource.
`vrm.client.resource.uri`	The URI must contain the IP address and port number of the MRCP Server or the MRCP Proxy by using the following format:

	`rtsp://servername:<port>/<path>` For the recommended resource Uniform Resource Identifier (URI), check the MRCP vendor documentation. Tip The MRCP Proxy supports MRCPv1 speech resources only.
`vrm.proxy.ping_interval`	Enter a value (or retain the default) to specify the ping interval in milliseconds (used only when the MRCP Proxy is deployed). Default value: 30000
For MRCPv2	
`vrm.client.resource.name`	Enter the identifier used to link the VoiceXML application to a common set of speech resources for example, For ASR, enter NUANCE or RECOGNIZER. For TTS, enter NUANCE or VOCALIZER.

`vrm.client.resource.uri`	The URI must contain the IP address and port number of the MRCP server using one of two formats:
	
	`sip:mresources@<MRCP server IP>:<port>;transport=TLS`
	
	`sips:mresources@<MRCP server`

	`IP>:<port>` (The default SIPS port number for Nuance Speech Servers is `5061`.) For the recommended resource URI, check the MRCP vendor documentation.
`vrm.client.TransportProtocol`	Enter one of two values: MRCPv2 without Security MRCPv2 with secure TLS

7. Save the configuration.

> Tip
>
> Complete Steps 8 to 11 if you are deploying MRCPv2 with Secure RTP (SRTP) only.

Configure the Media Control Platform Application

8. Select the Media Control Platform Application that is associated with this speech resource.
   The Configuration tab appears.
9. Click the Options tab, and scroll to the `mpc` section.
10. Configure the following parameters with the values that are shown here:
    - `asr.srtp.mode=offer`
    - `asr.srtp.sessionparams=none`
    - `tts.srtp.mode=offer`
    - `asr.srtp.sessionparams=none`
11. Save the configuration.
    Configure the ASR Server
12. Configure the following options on the ASR Server:
    - If the ASR Server supports session timeout, configure `600000` (milliseconds, or 10 minutes) for the timeout value to prevent interruption of any active recognition sessions.

- For Nuance SpeechWorks MediaServer and OpenSpeech Recognizer, configure the server.transport.sessionTimeout VXIInteger option with a value of `600000` (10 minutes).
- For Nuance Speech Server and Nuance Recognizer, configure the `server.mrcp2.sip.sessionTimeout` and `server.mrcp1.rtsp.sessionTimeout` options with a value of `600000` (10 minutes).
  For other ASR vendors, check the vendor documentation.

13. To make the ASR service work correctly with GVP, you must edit the Nuance Recognizer file `baseline.xml`. and comment out the third and fourth lines in the code sample below:

```
<param name="swirec_extra_nbest_keys">

<value>SWI_meaning</value>

<style="color: red;"><!--</style> <value>SWI_literal</value>
<style="color: red;">--></style>

<style="color: red;"><!--</style>
<value>SWI_grammarName</value> <style="color:
red;">--></style>

</param>
```

The characters to add to the code are marked in red.
14. Assign the MRCP Server to the Media Control Platform Application object. See Procedure: Assigning the MRCP Server to the Media Control Platform.

Procedure: Assigning the MRCP Server to the Media Control Platform

Use this procedure if you have not deployed the MRCP Proxy, otherwise see Provisioning the MRCP Proxy.

1. Verify that:
   - The MRCP Speech Resource object templates are imported. See Procedure: Integrating Application Objects with Resource Manager.
   - The MRCP Speech Resource objects are created. See Procedure: Creating Application Objects Manually.
   - The MRCP Speech Resource objects are provisioned. See Procedure: Provisioning Speech Resource Application Objects.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.

4. Double-click the Media Control Platform Application object that you want to configure.
   The Configuration tab appears.
5. In the General section, in the Connections field, click **Add**.
   The Connection Info dialog box appears.
6. Enter the information in the required fields, as shown in this table:

Table: Connection Info Dialog Box

Field	Description
Server	Browse to select the MRCP Server.
ID	This field is populated automatically with the value default.
Trace Mode	This field is populated automatically with the value Trace is Turned Off.
Application Parameters	Enter `provisiontype=primary` for a primary MRCP server. Enter `provisiontype=backup` for a backup MRCP server.

7. Click **OK**.
8. Save the configuration.

> Tip
>
> There is no limit to the number of primary or backup MRCP servers that you can assign to the Media Control Platform; however, do not assign the same server as both primary and backup.

9. If required, complete the post-installation activities for the Supplementary Services Gateway. See Provisioning the PSTN Connector.

Provisioning the MRCP Proxy

The MRCP Proxy is an optional component, but must be deployed if ASR and TTS usage reporting is required. You can deploy the MRCP Proxy in stand-alone or warm active standby HA mode. The procedures in this section describe the steps for each configuration.

Tip

By design, the MRCP Proxy supports only the NUANCE speech resource.

Procedure: Configuring the MRCP Proxy

Configure the MRCP Proxy to act as a proxy for all MRCPv1 traffic in the environment.

1. Verify that:
    - The MRCP Speech Resource objects are provisioned. See Procedure: Provisioning Speech Resource Application Objects.
    - The server connections are created. See Procedure: Creating a Connection to a Server.
    - The connections to the ASR and TTS resource access points are created. See Procedure: Provisioning Speech Resource Application Objects.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select Environment > Applications.
4. Double-click the MRCP Proxy Application that you want to configure.
   The Configuration tab appears.
5. Click the Options tab, in the vrmproxy section, configure the host part of the uri configuration option with the actual IP address of the MRCP Proxy.

   Important

   If the Media Control Platform is installed on the same host as the MRCP Proxy, retain the default value for the uri configuration option.

6. Create a connection to the MRCP Server. See the Prerequisites section of this procedure.
7. Save the configuration.

Procedure: Configuring the MRCP Proxy for HA

A configured MRCP Proxy acts as a warm standby in case of failover which means that, like a hot standby, the standby instance becomes active if the active instance fails. However, unlike a hot standby, a warm standby does not handle existing sessions. Application requests are rejected mid-stream during a failover; and applications must be designed to accommodate such a failure.

The failover sequence of events is as follows:

1. The primary MRCP Proxy terminates.
2. The LCA in the primary MRCPP machine informs SCS about this event.
3. SCS checks to see if the terminated MRCPP has a backup instance configured.
4. If there is a backup instance configured, SCS instructs through LCA in the backup computer the other MRCPP to become primary.

In a standard configuration, the MRCP Proxies are configured as backup to each other, and SCS has an HA license to perform a switch-over.

This procedure configures the MRCP Proxy in HA mode to act as a proxy for all MRCPv1 traffic in the environment.

1. Verify that:
    - The latest versions of Management Framework and LCA are installed and the Solution Control Server (SCS) Application is configured to support HA licenses. See the Framework 8.5.1 Deployment Guide and the Framework Management Layer User's Guide.
    - The prerequisites in Procedure: Configuring the MRCP Proxy are fulfilled. The prerequisites for the MRCP Proxy backup server are the same as for the primary in HA mode, and the connections must be the same on both MRCP Proxy Applications in the HA pair.|2}}
2. Complete Procedure: Configuring the MRCP Proxy for both MRCP Proxy Applications in the HA pair.
3. In the primary MRCP Proxy Application, click the Configurations tab.
4. In the Server Info section, in the Backup Server field, browse to the backup MRCP Proxy Application and click to select it.
5. In the Redundancy field, select warm-standby.
6. Save the configuration.
   Connect to the MCP
7. In the Media Control Platform Application, create a connection to the primary MRCP Proxy.
8. Save the configuration.

Procedure: Adding a Speech Server as Primary or Backup

1. Log in to Genesys Administrator.

2. On the Provisioning tab, select **Environment > Applications**.
3. Select the MRCP Proxy Application that you want to configure and click Manage Connections.
   The Manage Connections dialog appears.
4. Click **Next** twice in the Manage Connections dialog.
   The Add Connections dialog appears.
5. Click **Add** and select the speech server to add.
6. Click **Edit** and select the Advanced tab.
7. Enter `provisiontype=primary` in the Application Parameters field, to add the speech resource as primary.
   OR
   Enter `provisiontype=backup` in the Application Parameters field to add the speech resource as backup.
8. Click **Execute** and then **Finish**.

Configuring GVP Components 2

Perform these advanced configuration procedures after installation and basic configuration.

Configuring the CTI Connector for Cisco ICM

When you install the CTI Connector, you can select the CTI Framework that is appropriate for your environment Genesys CTI or Cisco ICM. Use the procedure in this section to configure the CTI Connector if you selected Cisco ICM.

- Procedure: Configuring the CTI Connector for Cisco ICM Integration

Procedure: Configuring the CTI Connector for Cisco ICM Integration

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select `Environment > Applications`.
3. Double-click the CTI Connector Application that you want to configure.
   The Configuration tab appears.
4. Click the Options tab.
5. If you want to use the Call Routing Interface (CRI), in the icmc section:
    - Configure the ICMInterface option with the CRI value.

      > Tip
      >
      > During installation, when you select Cisco ICM, the Service Control Interface is initialize by default.

    - Configure the TrunkGroupID option with an applicable value.

   For Single Tenant Environments:

6. In the `Tenant1` section:
    - Enter the tenant name in the `TenantName` configuration option value field.
    - Change the value of the Ports configuration option, as required. For example, `8000` (or retain the default, `9000`).

   For Multi-Tenant Environments:

7. Copy the `Tenant1` section and rename it for each additional tenant. For example: `Tenant2`, `Tenant3`, `Tenant4`.
8. For each newly created tenant:
    - Enter the tenant name in the TenantName configuration option value field.
    - Change the value of the Ports configuration option, as required. For example, `8000` (or retain the default, `9000`). Ensure that there are no duplicate ports configured across all tenants.

      > Tip
      >
      > You can specify a comma-separated list of listener ports for a single tenant, one for

> each VRU-PG. For example, 8000, 9000, 10000.

9. Save the changes.
10. Configure an IVR Profile to support ICM call flows, see *Chapter 6 in the Genesys Voice Platform 8.5 User s Guide*.

Provisioning the PSTN Connector

The procedures in this section describe how to configure the mandatory parameters for the Public Switched Telephone Network (PSTN) Connector Application object and the how to integrate the PSTN Connector with SIP Server. There are many more configurable parameters for the PSTN Connector, all of which are optional. For a complete list and description of configuration options, see the *Genesys Voice Platform 8.5 User's Guide*.

The PSTN Connector component is required if you are planning to migrate from GVP 7.x Voice Communication Server (VCS), or a VoiceGenie (VG) TDM interface to GVP 8.1.2 or later.

- Procedure: Configuring the PSTN Connector
- Procedure: Configuring a Trunk DN for the PSTN Connector

Procedure: Configuring the PSTN Connector

Prepare the PSTN Connector to manage inbound and outbound calls for GVP.

1. Verify that:
    - All of the GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP, on page 221.
    - The connections to Message Server, SIP Server, and the SNMP Master Agent are configured in the PSTN Connector Application object. See *Procedure: Creating a Connection to a Server, on page 243*.
    - SIP Server is installed. See the Voice Platform Solution 8.1 Integration Guide.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select Environment > Applications.
4. Select the PSTN Connector Application object that you want to configure.
   The Configuration tab appears.
5. Click the Options tab, and from the View drop-down list, select the Mandatory Options view.

6. In the DialogicManager_Route1 and GatewayManager sections, enter the values for the mandatory options as shown in the table below.

Table: PSTN Connector Mandatory Parameters

Section	Option	Value
DialogicManager_Route1	RouteType	Specify one of three route types or call directions for this route, enter: ◦ 0 for Inbound ◦ 1 for outbound ◦ 2 for In/Out (See Note below, in this table.)
	Signaling Type	Specify one of five signaling types, enter: ◦ 0 for T1-ISDN (PRI) ◦ 1 for Analog ◦ 2 for E1-ISDN (PRI) ◦ 3 for T1-RobbedBit ◦ 4 for E1-CAS
	Channels	Specify the ports for this route by using the format, [<Card>:<PortRange>,<Card>:<PortRange>]. You can provision more than one board in a route and a partial range of ports in a board for example: ◦ 1:1-23 ◦ 1:1-23,2:1-23 ◦ 1:1-30,2:1-30 ◦ 1:1-12,2:1-15
GatewayManager	SIP Destination IP Address	Enter the SIP endpoint IP address that will receive SIP calls from the PSTN Connector. (This is the IP address for SIP Server or the Resource Manager, depending on your configuration.)

<table>
<tr><td></td><td>SIP Destination Port Number</td><td>Enter the SIP endpoint port number of the server that is configured in SIP Destination IP Address.</td></tr>
<tr><td>MediaManager</td><td>Supported Local Codec Type</td><td>Enter the audio format that is in use on the TDM trunk:
0 - Mulaw
8 - ALaw</td></tr>
<tr><td colspan="3">Notes
The Inbound & Outbound route type is supported only on ISDN (PRI) lines. If you select one of these route types, ensure that you use a compatible signaling type.
If you are using a T1-Robbed Bit or E1-CAS interface, options in the T1rb options group must be configured, specifically the T1rbProtocolFile option, see the component metadata.
For JCT boards only, a separate span is required to support ASR or recorded VoiceXML applications in T1-ISDN, E1-ISDN, or E1-CAS environments. The MediaVoxResourceBoard option must be configured with route number that is used for CSP.
When JCT boards are used with the PSTN Connector, and the VoiceXML application uses ASR or recording media, not all the spans can be used for call handling. For each span that is configured to take calls, there must be another dedicated span for streaming echo cancelled audio to the Media Control Platform. Therefore, if Route1 is configured to handle calls on span1 (for example, ports=1:1-23), the <code>MediaVoxResourceBoard</code> option under the <code>DialogicManagerRoute1</code> section should be set to <code>2</code>. Repeat the same steps if you have configured a DialogicManager_Route2 section.
This restriction does not apply in the following scenarios:
The VoiceXML application is a pure DTMF application (does not use ASR or recording media).
The VoiceXML application uses ASR but the JCT board is configured with the T1-Robbed Bit protocol.</td></tr>
</table>

7. Create additional `DialogicManager_Route<N>` sections as required (for example, you may want to create a section for inbound ports and one for outbound ports):
    a. From the View drop-down list, select either Advanced View (Options) or Advanced View (Annex).
    b. Right-click on the Section column heading and select New.

c. Enter `DialogicManager_Route2` for the Section name, Description for the Option name, and Route2 Information for the Value.
d. Click **OK**.
e. Copy and paste all of the options from the `DialogicManager_Route1` and `DialogicManager_Route2` sections, modifying the mandatory values as required.
f. Repeat these steps as required.

> Tip
>
> **Dialogic Circular Buffer Size**
>
> When you configure the PSTN Connector application, set the value for [DialogicManager] DialogicTransferBufferSize to 2048. This specifies the size of the Dialogic Circular buffer which is used for transferring data to the Dialogic firmware. The default value of this parameter on Windows is different, but must be set to 2048 for Linux. If not, the Dialogic may return LOW_WATER_MARK warnings during initial play of the media, which interferes with normal audio play and may result in the prompt being cut off.

8. Save the configuration.
9. Configure the Trunk and Trunk Group DNs for the PSTN Connector. See Procedure: Configuring a Trunk DN for the PSTN Connector.

Procedure: Configuring a Trunk DN for the PSTN Connector

Configure SIP Server with a Trunk DN that points to the PSTN Connector Application object, to ensure that outbound calls can be routed to a specific PSTN Connector instance.

You can deploy multiple PSTN Connectors, however, you must ensure that SIP Server routes the outbound call to the same PSTN Connector instance as the inbound call. This procedure includes configuration options to enable this functionality.

1. Verify that the PSTN Connector is installed and configured. See Procedure: Configuring the PSTN Connector.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Switching > Switches`.
4. Double-click the switch that you want to configure.
   The Configuration tab appears.
5. On the DNs tab, select **New**.
6. In the General section, enter values for the mandatory fields, selecting **Trunk** from the Type drop-down list.
7. In the Switch pane, double-click the Trunk DN you created in Step 4.
8. On the Options tab, select Advanced View (Annex) from the View drop-down list.
9. Right-click on the Section column, and select **New**.
10. In the New Option dialog:
    - In the Section field, enter `TServer`.
    - In the Name field, enter `contact`.
    - In the Value field, enter the IP address and port number of the PSTN Connector separated by a colon for example, `10.10.10.101:5060`
11. In the DNs pane, click **New**.
12. In the TServer section, add the following options and values:
    - `prefix = <xyzz>` where `<xyzz>` represents a number which, if present in the dial string of the outbound call, enables the SIP Server to route the call to the PSTN Connector instance that is configured with this prefix.
    - `replace-prefix = <empty String)` where `<empty String)` represents an empty string to ensure that the prefix added by Resource Manager to the destination number string is removed by the SIP Server before the call is forwarded to the same PSTN Connector instance.
13. Save the configuration. For information about how the PSTN Connector fits into a common VPS deployment architecture, see the Supported Architecture chapter in the Voice Platform Solution 8.1 Integration Guide.
14. If required, complete the post-installation activities for the Supplementary Services Gateway. See Provisioning the Supplementary Services Gateway.

Provisioning the Supplementary Services Gateway

The following procedures describe how to configure the mandatory parameters for the Supplementary Services Gateway Application object and the how to integrate the Supplementary Services Gateway with SIP Server. There are many more configurable parameters for the Supplementary Services Gateway, all of which are optional. For a complete list and description of configuration options, see the GVP 8.5 User's Guide.

The Supplementary Services Gateway is an optional component and is required only if you intend to support outbound call campaigns in your deployment.

Tip

Multiple instances of the Supplementary Services Gateway can be installed on the same host; however, the HTTPPort and HTTPSPort parameters must have unique values and cannot be the same for more than one instance.

This section contains the following procedures:

- Configuring the Supplementary Services Gateway
- Configuring DNs for the Supplementary Services Gateway
- Configuring a Routing Point DN
- Configuring a Voice Over IP Service DN
- Configuring a Voice Treatment Ports DN

Procedure: Configuring the Supplementary Services Gateway

Prepare the Supplementary Services Gateway to receive and respond to outbound call initiation requests from TAs.

1. Verify that:
    - All other GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP, on page 221.
    - The connections to Message Server, SIP Server, and the SNMP Master Agent are configured in the Supplementary Services Gateway Application object. See Procedure: Creating a Connection to a Server.
    - SIP Server is installed. See the Voice Platform Solution 8.1 Integration Guide.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Select the Supplementary Services Gateway Application object that you want to configure.
   The Configuration tab appears.
5. Click the Options tab, and select Advanced View (Options) from the View drop-down list.
6. In the Tenant1 section, enter Environment in the Value field of the TGDN option. (The value that is configured in the TGDN parameter is used as the tenant name.)

7. Create additional `Tenant<n>` sections, as required&mdahs;for example: Tenant2, Tenant3, and so on. To create new sections by copying and pasting options, see Step 6 in Procedure: Configuring the PSTN Connector.
8. Save the configuration.
9. Configure the Trunk and Trunk Group DNs to route for Resource Manager and external numbers. See Procedure: Configuring DNs for the Supplementary Services Gateway.

Procedure: Configuring DNs for the Supplementary Services Gateway

Configure SIP Server with a Trunk Group DN that points to the Resource Manager Application object, and a Trunk DN that is used to route calls to external numbers.

The Trunk Group DN must have the same name as the tenant for which the Supplementary Services Gateway will make calls. This configuration enables the SUBSCRIBE and NOTIFY messages between SIP Server and the Resource Manager.

1. Verify that the Supplementary Services Gateway is installed and configured. See Procedure: Provisioning the Supplementary Services Gateway.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select Environment > Applications.
4. Select the SIP Server Application object you want to configure. The Configuration tab appears.
5. On the Options tab, in the View drop-down list, select Advanced View (Options).
6. In the TServer section, change the value of the following Options:
    - `am-detected = connect`
    - `fax-detected = connect`
    - `cpd-info-timeout = 7`
    - `sip-invite-treatment-timeout = 30`
7. Configure a DN on the SIP switch of type Trunk Group, and, for the DN name, enter the name of the tenant for example, `Tenant1`.
8. Configure a DN on the SIP switch of type Trunk as the endpoint. The endpoint is the destination of the outbound call for example, a softphone.
Configure Trunk Group DN
9. After the `<TenantName>` Trunk Group DN is configured, enter the values for the following parameters in the TServer section of the Annex, as shown in the figure below.
    - `subscription-id = <TenantName>` Must be the name of the tenant that is associated with the Resource Manager. (The DN, tenant, and subscription-id must all have the same name: <TenantName>.

> **Tip**
>
> Tenant names are case sensitive and must be used consistently, especially in outbound scenarios. For example, if you create the tenant names, tenant1 and Tenant1, they are treated as two separate tenants. The value that is configured in the TGDN parameter is used as the tenant name.

- `contact = sip:172.24.133.50:5060` The IP address and port number of the Resource Manager. (The value shown here for contact is only an example.)
- `cpd-capability = mediaserver` This configuration designates the Media Server module of the Media Control Platform as the CPD provider.
- `request-uri = sip:msml@172.24.133.50:5060;gvp-tenant-id=TenantName` The value of this parameter should point to the Resource Manager, and the user part must contain msml.

Configure Trunk DN

10. After the Trunk DN is configured, enter the value for the following parameter in the TServer section of the of the Advanced View (Annex):
    - `contact = sip:172.21.193.61:10000` The IP address and port number of the external party. (The value shown here for contact is only an example).

> **Tip**
>
> If CPD is enabled at Media Gateway, configure CPD at the Trunk. If CPD is enabled at both Trunk and Trunk Group, CPD enabled on the media gateway takes precedence by default.

For information about how to configure the Supplementary Service Gateway within the VPS, see the VP Solution 8.1 Integration Guide.

11. (Optional) Configure a Routing Point DN. See Configuring a Routing Point DN.
12. Complete the post-installation activities for the Call Control Platform if you intend to use it for outbound calling. See Procedure: Configuring the Call Control Platform.

Procedure: Configuring a Routing Point DN

Create a Routing Point DN to use for outbound calls with the legacy GVPi.

Use this procedure if you want to support your outbound solution by making outbound calls through a Routing Point DN (instead of a Trunk Group DN) by using the legacy GVPi and CTI functionality. Create a Routing Point DN for each tenant in your environment.

1. Verifythat a Trunk DN exists on the SIP switch for outbound calls. See the VP Solution 8.1 Integration Guide.

Configure the Routing Point DN

2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Switching > Switches`.
4. Double-click the SIP Server Switch object you want to configure.
   The Configuration tab appears.
5. Click the DNs tab and select **New**.
6. On the DN Configuration tab:
   a. In the Number field, enter valid Route Point DN number. (Ensure there is no conflict with the Trunk Group DN that was created for receiving port details.)
   b. In the Type field, select **Routing Point** from the drop-down list.
7. On the Options tab, click **New**.
8. In the New Option dialog box:
   a. In the Section field, enter `TServer`.
   b. In the Name field, enter `partition-id`.
   c. In the Value field, enter the name of the tenant (for which this Route Point DN is configured). This configuration enables SIP Server to select an MSML service with the same partition-id.

Configure the SSG Application

9. On the Provisioning tab, select `Environment > Applications`.
10. Select the Supplementary Services Gateway Application object that you want to configure.
    The Configuration tab appears.
11. Click the Options tab, and select **Advanced View (Options)** from the View drop-down list.

12. In the Tenant1 section:
    a. For the RPDN option, enter a value that matches the RPDN. For example, `RP 2000`.
    b. For the TGDN option in the Value field, enter the tenant name. The value that is configured in the TGDN parameter is used as the tenant name.
13. If you want to send outbound calls for multiple tenants, create additional `Tenant <N>` sections configured with the RPDN option, as required for example, Tenant2, Tenant3, and so on. To create new sections by copying and pasting options, see Step 6 in Procedure: Configuring the PSTN Connector.
14. Save the configuration.
15. Configure a VoIP Service DN. See Procedure: Configuring a Voice Over IP Service DN, below:

Procedure: Configuring a Voice Over IP Service DN

Create and configure a VoIP Service DN that is used by SIP Server to initiate MSML dialogs (to obtain CPD) when a request is received on the Routing Point DN.

The VoIP Service DN works in conjunction with the Routing Point DN. For outbound calls, SIP Server selects the MSML service type with the value that is configured in the partition-id option (either no partition-id or the partition-id that is configured in the SIP Server application).

1. Verify that a Routing Point DN has been configured. See Configuring a Routing Point DN.

   Configure the Voice Over IP Service DN
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Switching > Switches`.
4. Double-click the SIP Server Switch object you want to configure.
   The Configuration tab appears.
5. Click the DNs tab and select **New**.
6. On the DN Configuration tab:
    a. In the Number field, provide a unique, valid number for the DN.
    b. In the Type field, select Voice Over IP Service from the drop-down list.
7. On the Options tab, click **New**.
8. In the New Option dialog box:
    a. In the Section field, enter `TServer`.
    b. In the Name field, enter `contact`.
    c. In the Value field, enter the IP address and port number of the Resource Manager, for example, `sip:172.24.133.50:5060`.

9. Repeat the two previous steps to add the following options and values:
    - `cpd-capability = mediaserver`
    - `partition-id = <tenant_name>`
    - `service-type = msml`
    - `subscription-id = <tenant_name>`
10. Click **Save** and **Close**.
11. Repeat Steps 4 to 9 to create a second VoIP Service DN to play Treatments (`service-type = treatment`).
12. (Optional) Configure VTP Ports. See Procedure: Configuring a Voice Treatment Port's DN, below:

Procedure: Configuring a Voice Treatment Ports DN

Create and configure a Voice Treatment Ports (VTP) DN to play IVR Profile VoiceXML dialogs. You can create any number of VTP DNs to control the number of simultaneous outbound requests that can be placed on the route point.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select `Switching > Switches`.
3. Select the SIP Server Switch object you want to configure.
   The Configuration tab appears.
4. Click the DNs tab and select **New**.
5. On the DN Configuration tab:
    a. In the Number field, enter a valid Voice Treatment Port as the number.
    b. In the Type field, select **Voice Treatment Port** from the drop-down list.
6. On the Options tab, click **New**.
7. In the New Option dialog box:
    a. In the Section field, enter `TServer`.
    b. In the Name field, enter `contact`.
    c. In the Value field, enter the IP address and port number of the Resource Manager, for example, `sip:172.24.133.50:5060`.
8. Repeat Steps 6 and 7 to add the following options and values:
    - `prefix = mediaserver`
    - `request-uri = sip:<vtp DN name>@<RM contact>gvp-tenant-id=<tenantName>`
    - `event-ringing-on-100trying = true` (when CTI Connector is used on the call, otherwise, false)
    - `cpd-capability = mediaserver` (the Media Control Platform performs CPD)

- `userdata-map-filter = constant string "gsw-ivr-profile-name,gsw-session-dbid,OutboundData, AnswerClass,outbound-ivr-call"`

  Where:
- `OutboundData` is the filter that is provided to pass user data from the Supplementary Services Gateway to the IVR Application. For information about attaching user data to outbound calls, see the Voice Platform Solution 8.1 Integration Guide.
- `outbound-ivr-call` must be passed to the Resource Manager to ensure the media service call is not dropped until the route is used when a temporary double-counting of IVR Profile usage can happen when the IVR VoiceXML call leg is placed.
- `AnswerClass` is required if the CPD result must be passed to VoiceXML application. Add the `GVP-IVRPort` parameter if the Supplementary Services Gateway is integrated with the PSTN Connector and the `GVP-PSTNC-DBID` parameter if the Supplementary Services Gateway is integrated with the CTI Connector, for example: `userdata-map-filter = constant string "gsw-ivr-profile-name,GSW_SESSION_DBID,gsw-session-dbid, OutboundData,AnswerClass,GVP-IVRPort,GVP-PSTNC-DBID"`

> Tip
>
> Genesys recommends that you create a strategy to select a available VTP DN (from the set of VTP DNs that you created) to load on the Routing Point DN. For more information about routing strategies, see Chapter 11 of the Voice Platform Solution 8.1 Integration Guide.

Preparing the Call Control Platform for Outbound Calling

This section describes how to prepare the Call Control Platform to make outbound calls. The Call Control Platform is an optional component.

Procedure: Configuring the Call Control Platform

Configure the Call Control Platform Application object to make outbound calls.

1. Verify that:
   - All of the GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP.
   - The Call Control Platform is integrated with the Resource Manager. See Procedure: Integrating Application Objects with Resource Manager.
   - A Call Control Platform connection to the Message Server is created. See Procedure: Creating a Connection to a Server.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Select the Call Control Platform Application.
5. Click the Options tab, and scroll to the `mediacontroller` section.
6. Click the Value field of the `sipproxy` option, and then enter
   <IP_RM>:<SIPPort_RM>
   Where `IP_RM` is the IP address of the Resource Manager, and `SIPPort_RM` is the SIP port of the Resource Manager.
7. Click the Value field of the `bridge_server`, and then enter
   <IP_RM>:<SIPPort_RM>
   Where `IP_RM` is the IP address of the Resource Manager, and `SIPPort_RM` is the SIP port of the Resource Manager (`5060` is the default).
8. Click **Apply**.
9. Save the configuration.
10. Complete the post-installation activities for the Resource Manager. See the next section, *Using Resource Groups*.

Configuring GVP Components 3

Perform these advanced configuration procedures after installation and basic configuration.

- Provisioning the MRCP Proxy

Using Resource Groups

To enable ease of management, common GVP resources can be grouped into Logical Resource Groups (LRGs). When multiple instances of a resource, such as, the Media Control Platform, Call Control Platform, or CTI Connector, are assigned to a resource group, the Resource Manager can easily manage and provide load balancing for the resources within the group. In addition, connections are created to enable the physical resources to communicate with the Resource Manager so that they can be assigned to fulfill requests for services. To create the connections, see Procedure: Creating a Connection to a Server.

> **Important**
>
> Genesys strongly recommends that you configure a gateway resource group for SIP Server; not doing so may cause Resource Manager to generate errors for certain call scenarios.

Procedure: Creating a Resource Group

Group resources that use common services and provide load balancing.

The MCPGroup created in this procedure provides load balancing for the resources within it that are using VoiceXML services. If you have one or more Call Control Platforms installed in your deployment, create a resource group that includes resources that use CCXML services. You can also create a group to manage resources that use the gateway, CTI, conference services, or recording servers.

1. Verify that all of the GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, click **Voice Platform > Resource Groups**.
4. On the Details pane toolbar, click **New**.
   The Resource Group Wizard opens to the Welcome page.
5. On the Resource Manager Selection page, select the Resource Manager Application object for which you want to create the group. On the Group Name and Type page:
6. Enter a group name; for example, `MCPGroup`.

7. Select one of five group types:
    - Media Control Platform
    - Call Control Platform
    - Gateway
    - CTI Connector
    - Recording Server

> Tip
>
> When creating resource groups in multi-tenant environments, ensure that only one CTI Connector resource group is configure for the entire hierarchy.

8. On the Tenant Assignments page, select the child tenant to which the resource group will be assigned.
9. On the Group Properties page, enter the information from the table below for each resource group that you are configuring.

> Tip
>
> For the Media Control Platform group, the Max.Conference Size and Max.Conference Count, and Geo-location options are optional; therefore, they are not included in thus table. For a complete list of resource-group options and their descriptions, see the GVP 8.5 User's Guide.

Table: Group Properties Resource Groups Wizard

Field name	Value
Media Control Platform	
Monitoring Method	Retain the default value: SIP OPTIONS.
Load Balance Scheme	Select least percent.
Call Control Platform	
Monitoring Method	Retain the default value: none.
Load Balance Scheme	Select least percent.
Gateway	

Monitoring-Method	Retain the default value: none.
Load Balance Scheme	Select least percent.
CTI Usage	Select Always off, Always on, or Based on DN lookup.
CTI Connector	
Monitoring-Method	Retain the default value: none.
Load Balance Scheme	Retain the default value: round-robin.
Recording Server	
Monitoring-Method	Retain the default value: SIP Options.
Load Balance Scheme	Retain the default value: round-robin.

10. On the Resource Assignment page:
    a. Select the checkbox beside each resource you want to assign to this group.
    b. In the SIP Port column, click in the column to select a port number from the drop-down list.
    c. In the SIPS Port column, click in the column to select a port number from the drop-down list.

    Tip

    When you are creating Gateway resource groups: there is only the SIP Port column, and you must enter the port number (there is no drop-down list of ports).

    d. In the Max Ports column, enter a number that represents the maximum number of requests this resource is capable of handling.
    e. In the Redundancy column, click in the column to choose active or passive from the drop-down list. The Resource Assignment list is compiled depending on the type of group that you are creating; for example, if you are creating a Media Control Platform group, only Media Control Platform servers appear in the list.
11. On the Confirmation page, click **Finish**.
12. Continue with the post installation activities for the Resource Manager. See Procedure: Creating IVR Profiles.

Creating IVR Profiles and DID Groups

GVP uses IVR Profiles, which are VoiceXML, CCXML, Announcement, and Conference applications, to control interactions that require the use of Direct Inward Dialing (DID) numbers and provides service for the resources that use them.

Tip

IDs were formerly referred to as Dialed Numbers (DN) in GVP releases prior to GVP 8.1.2.

You can create as many IVR Profiles as you need and any number of DIDs or DID ranges. DIDs are grouped into DID Groups for ease of assignment and administration. DIDs are obtained from the Dialed Number Identification Service (DNIS). The Resource Manager can be configured to obtain DNIS information from SIP Server.

If GVP is configured to map DIDs to IVR Profiles or a tenant, the Resource Manager uses DNIS to determine which IVR Profile to invoke for the session. If GVP is not configured in this way, the Resource Manager uses a default IVR Profile that is specified for the Environment (or default) tenant.

This section contains the following procedures:

- Procedure: Creating IVR Profiles
- Procedure: Adding a Context Services base URL to an IVR Profile
- Procedure: Creating DID Groups

Procedure: Creating IVR Profiles

Create IVR Profiles that use DIDs to provide service for the resources that use them.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select **Voice Platform > IVR Profiles**.
3. In the Tasks panel, click **Define New IVR Profile**.
   The IVR Profile Wizard opens to the Welcome page.
4. On the Service Type page:
   a. Enter a name for the IVR Profile for example, `VPS_IVRProfile`.

Tip

> The IVR Profile name is case-sensitive and can be up to 255 characters in length. For information about naming IVR Profiles, see Genesys Administrator 8.1 Help.

b. From the drop-down list, select one of four service types:
    - VoiceXML
    - CCXML
    - Conference
    - Announcement

5. On the Service Properties page, enter the mandatory values from Table: IVR Profile Wizard Service Properties for the service type that you selected in the previous step. This table includes only those options that are mandatory to create an IVR Profile. For a complete list of the options used to configure IVR Profiles and their descriptions, see the GVP 8.5 User's Guide.

Table: IVR Profile Wizard Service Properties

Service Type	Field	Value
VoiceXML	Initial Page URL	Enter the Universal Resource Locator (URL) to your VoiceXML page for example, `http://samples/hello.vxml` or `file:///C:/GVP/VP_MCP/samples/helloaudio.vxml`
CCXML	Initial-Page-URL	Enter the URL to your CCXML page for example, `http://samples/hello.vxml` or `file:///C:/GVP/VP_CCP/samples/helloaudio.ccxml`
Conference	Conference-ID	Enter a value that starts with a letter, number, or underscore (cannot exceed 255 characters), for example: `3332`.
Announcement	Play	Enter the URL that points to the announcement you want to play for example, `http://samples/hello.vxml` or `C:/GVP/VP_CCP/samples/announcements`.

Note that the URLs in this table are examples. When you create your IVR Profiles, enter the URLs that point to the actual VoiceXML, CCXML, Conference, or Announcement applications in your environment. The small icon to the right of the URL field in the wizard, is used to load the URL into a pop-up web page, verify the accuracy, and confirm that an application actually exists at that location.

Tip

After the Service Properties are entered, you have the option of clicking Finish and a basic IVR Profile is created. However, if you want to customize the profile, you can continue on through the Usage Limits, IVR Capabilities, CTI Parameters, and Dialing Rules pages which contain optional configuration parameters. For more information about these configuration parameters, see the GVP 8.5 User's Guide.

6. On the Usage Limits page, in the Maximum Concurrent Sessions field, enter a number to define the maximum number of concurrent sessions that can be used by the IVR Profile.
7. On the IVR Capabilities page, configure the parameters in the table below as required for your IVR Profiles. The IVR Capabilities page appears only if you have selected the VoiceXML or CCXML service types in Step 4.

IVR Capabilities Page IVR Profile Wizard

Option	Description
Allow Outbound Calls	Insert a check mark to enable (or leave blank to disable). Sets the value of the outbound-call-allowed parameter (for bridge or consultation transfers, as well as for outbound calls), in the gvp.policy configuration section. By default, INVITE transfers are enabled
Allows Transfers	Insert a check mark to enable (or leave blank to disable). Sets the value of the transfer-allowed parameter (for blind or consultation transfers), in the gvp.policy configuration section. By default, REFER transfers are enabled.

Gateway Selection	Select one of three options: ◦ Always use the same gateway ◦ Use same gateway if possible ◦ Use any available gateway

8. On the CTI Parameters pane, configure the parameters as described in the table below. The CTI Parameters page appears only if you have selected the VoiceXML service type earlier in this procedure.

Table: CTI Parameters Pane IVR Profile Wizard

Option	Description
Require CTI Interaction	Insert a check mark to enable (or leave blank to disable). Sets the value of the cti-allowed parameter, in the gvp.policy configuration section. By default, the CTIC is not required.
Transfer on CTI	Insert a check mark to enable (or leave blank to disable). Sets the values of the cti.transferoncti and cti.defaultagent, respectively, in the gvp.service-parameters configuration section.
Default Agent	The default agent to whom transfers will fall back if the original transfer fails.

9. On the Dialing Rules page:
   - In the Action field, retain the default value `Accept`.
   - In the Regular Expression field, enter the expression in the form of a URL. The Dialing Rules page appears only if you have selected the VoiceXML or CCXML service type in step 4 of Procedure: Creating IVR Profiles.
10. On the Policies page, in the SQ Notification Threshold (%) field, enter a number between 1 and 100.
11. On the Confirmation page, if the configuration is correct, click **Next**.
12. Click **Finish**.
13. (Optional) Manually add a Context Services base URL to an IVR Profile. See Procedure: Adding a Context Services base URL to an IVR Profile.
14. Create the DID Groups. See Procedure: Creating DID Groups.

Procedure: Adding a Context Services base URL to an IVR Profile

Universal Contact Server (UCS) interfaces use a database that stores contact (customer) data. Classic UCS works with Genesys eServices (Multimedia). By using Context Services, which is an optional set of additional capabilities, UCS can work with other Genesys products and solutions, such as Genesys Voice Platform and Conversation Manager. This procedure is optional.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select **Voice Platform > IVR Profiles**.
3. Select the newly created IVR Profile.
4. To add a new option, click **New**:
    a. In the Section field, enter gvp.service-parameters.
    b. In the Name field, enter voicexml.cs_base_url.
    c. In the Value field, enter fixed, *<the base HTTP URL of the context services>*.

> Important
>
> The voicexml.cs_base_url value can also contain the username and password if it is required by the context services. If the username and password is required, use this syntax:
>
> http://<username>:<password>@<host>:<port>
> If the username and password is not required, use this syntax:
> http://<host>:<port>

5. Click **OK**.

Procedure: Creating DID Groups

Create DID Groups that contain DIDs to assign to IVR Profiles and tenants. DID Groups enable ease of management and assignment. The groups can contain a single DID, a range of DIDs, or no DIDs. Empty DID Groups can be created initially as placeholders until you are ready to populate them.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select **Voice Platform > DID Groups**.
3. Select **New**.
4. In the Name field, enter the name of the DID Group.
5. In the IVR Profile field, click the browse icon to find the IVR Profile or tenant that you want to associate with this DID Group.
6. In the DIDs field, click **Add**.
7. In the DID dialog box, enter a DID, a range of DIDs or a number prefix for example:
   ```
   1234

   4567-8901

   456*
   ```
8. In the DID Group Property panel, click **Save** or click **Save & New** to create another DID Group.
9. Configure the Environment (default) Tenant and default IVR Profile. See Assigning Default Tenants and Creating Default Profiles

Assigning Default Tenants and Creating Default Profiles

In multi-tenant environments, the default tenant and IVR Profile are used for those calls that are not validated or cannot be associated with a specific tenant or profile. To properly configure the default objects, a specific Resource Manager instance must be configured to manage the default tenant, the tenant data must be configured, and a default IVR Profile must be created.

> **Important**
>
> Mandatory for any tenant that is managed by Resource Manager:
> 1. You must create a default IVR-Profile object under that tenant. For simplicity, in the IVR-Profile's Annex tab, the parameter service-type under gvp.general section may point to voicexml.
> 2. In that tenant's Annex tab, the parameter default-application under gvp.general section must point to that profile object.

Use the following procedures to complete the task in this section:

- Procedure: Adding the Environment Tenant to the Resource Manager
- Procedure: Creating a Default Profile for the Default Tenant
- Procedure: Updating the Tenant Data

Procedure: Adding the Environment Tenant to the Resource Manager

Add the Environment Tenant to the Resource Manager Application that is used to create a default IVR application.

This procedure describes the steps to add the Environment tenant to the Resource Manager Application when GVP is deployed in a multi-tenant environment. If your environment is single-tenant, the default tenant is named Resources and not Environment.

1. Verify that all of the GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select **Environment > Applications**.
4. Click the Resource Manager Application object you want to configure.
   The Configuration tab appears.
5. In the Server Info section, in the Tenants field, click **Add**.
   A Browse dialog box appears.
6. Select Environment, and then click **OK**.
   The Environment Tenant object appears in the Tenants field.
7. Save the configuration.
8. Create a default IVR Profile for the Environment Tenant. See Procedure: Creating a Default Profile for the Default Tenant

Procedure: Creating a Default Profile for the Default Tenant

Create a default IVR Profile that can be used to accept calls other than those specified in the dialing plans.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select **Voice Platform > IVR Profiles**.
3. In the Tasks panel, click **Define New IVR Profile**.
   The IVR Profile Wizard opens to the Welcome page.
4. On the Service Type page:
   a. Enter the name of the default IVR Profile, `IVRAppDefault`.
   b. Select VoiceXML from the drop-down list.
5. On the Service Properties page, enter `http://samples/hello.vxml`.

6. Click **Finish**.
7. Update the Environment tenant data. See Procedure: Updating the Tenant Data.

Procedure: Updating the Tenant Data

Configure the tenant to look for the default IVR Profile application, so that calls other than those specified in the dialing plans are accepted.

1. All of the GVP components are installed. See Procedure: Using the Deployment Wizard to Install GVP.
2. A default IVR Profile has been created, named, IVRAppDefault. See Procedure: Creating a Default Profile for the Default Tenant.
3. Log in to Genesys Administrator.
4. On the Provisioning tab, select `Environment > Tenants`.
5. Click the Environment tenant or, if you are configuring a single-tenant environment, click **Resources**.
6. On the Options tab, create a new section named `gvp.general`.
7. In the `gvp.general` section, create a new option named `default-application`.
8. For the `default-application` option, enter the value `IVRAppDefault`.
9. Enter the values for the remaining options in the `gvp.general`, `gvp.policy` section, and `gvp.dnis-range` sections as shown in the table below.
   The default-application option is mandatory for a tenant.

   Sections, Names, and Values GVP Options

Section	Name	Value
`gvp.general`	`default-application`	`IVRAppDefault`
	`sip.sessiontimer`	`1800`
`gvp.policy`	`usage-limits`	`100`

   The values for the gvp.dnis-range configuration option are added automatically by the DID wizard.

   The IVR Profile and Environment tenant configuration sections, gvp.log, gvp.log.policy, and gvp.policy.dialing-rules can be further defined with many more supported options. For a complete list of these options, go to the Options tab of the Application object template.

10. Save the configuration.

11. Complete the post-installation activities for the Reporting Server. See Integrating the Reporting Server User Interface with GVP, below:

Integrating the Reporting Server User Interface with GVP

The Reporting Server User Interface (RPTUI) is installed when Genesys Administrator is installed, however, you can customize your environment by using the default Application object (or Configuration Server object) to configure port numbers, authentication, and HTTP settings.

In addition, you must create a connection to Reporting Server in the default Application object to ensure that the RPTUI functions properly. The RPTUI discovers the Reporting Server host based on this connection. Furthermore, the RPTUI reads and enforces the data reporting limits that are configured in the Reporting Server Application object (in the reporting section).

Finally, you must configure the Reporting Server logging and messaging parameters so that the monitoring and reporting functionality perform as intended.

Tip

During the installation of the Reporting Server, the RPTUI and the logging and messaging parameters are configured with default values. Therefore, unless your environment is better served by manually changing the configuration, the only requirement is to create the connection to Reporting Server in the default Application object. See Procedure: Creating a Connection to a Server.

Procedure: Configuring the Reporting Server User Interfaces

Configure the default Application object to ensure that the Reporting Server user interfaces are exposed and to create the connection to the Reporting Server.

1. Verify that:
    - Genesys Administrator is installed and fully functional. See the Framework 8.1 Deployment Guide.
    - All of the GVP components are installed and started. See Procedure: Using the Deployment Wizard to Install GVP.

2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Click the default Application object.
5. In the Connections section, click **Add**.
   The Connection Info dialog box appears.
6. In the Server field, click the Browse icon.
7. Select the Reporting Server to which you want to create a connection.
8. Click **OK**.
   The Reporting Server you selected appears in the Connection section.
9. On the Options tab, select GVP Reporting from the View drop-down list.
   The Options list is filtered, and all of the `rptui` section options appear.

> Tip
>
> If you do not see GVP Reporting, select Show options in groups from the View drop-down list. The list changes, and GVP Reporting is available for selection.

10. Retain or modify the values for the options in the `rptui` section, as shown in the table below.

Table: default Application Object Options Tab

Option	Value
`enablehttps`	Retain the default value, `false`.
`httpport`	Retain the default port value, `8080`, or enter a port number from `1030` to `65535`.
`httptimeout`	Retain the timeout value, `30`, or enter any value greater than `0`.
`username`	Enter a user name to enable the web server for authentication. (Must match the password that is configured in the Reporting Server Application.)
`password`	Enter a password. (Must match the password that is configured in the Reporting Server Application.)
`tzoffset`	Retain the default time zone offset value, -08:00, or enter a value in the format **shh:mm**, where **s** is either a plus (+) or

	minus (-), **hh** represents hours, and **mm** represents minutes.
`dsthours`	Retain the default value `01:00`, or enter a value in the format shh;mm, where `s` is either a plus (+) or minus sign (-), `hh` represents hour, and `mm` represents minutes.
`localtimeformat`	Retain the default value true to display the datetime fields in local time format, or enter false to display the datetime fields in Universal Time Coordinated (UTC).

Tip

Click on any option on the Options tab for a detailed description and the default value.

11. Save the configuration.
12. In the default Application object, create a connection to the Reporting Server. See Procedure: Creating a Connection to a Server.
13. Create a database for the Reporting Server. See Reporting Server Database.

Configuring the Reporting Server Locale

If the Reporting Server is installed on a host that is configured with a locale other than English (default), you must complete the procedure in this section to achieve full functionality of the Reporting Server.

Procedure: Configuring the Reporting Server Locale

Configure the Reporting Server with a locale that is not the default (English).

1. In the Reporting Server installation directory, locate the `JavaServerStarter.ini` file.
2. In the [JavaArgs] section, add the line `Duser.language=en`, for example:
```
[JavaArgs]

-Xmx1536M

-Duser.language=en
```
3. Open the Java Control Panel and on the Java tab, click View.

4. On the User tab, in the Runtime Parameters field, change the language setting. This configuration affects all JVM default locales.

Configuring the GVP Reporting Server Database

The GVP 8.5 Reporting Server requires one of two supported relational database systems is installed Microsoft SQL Server or Oracle. The database and the Reporting Server can share a host or you can install the database on a separate host. This section describes how to create the GVP Reporting database schema and partitioning the database in the following topics:

- Before You Begin
- Setting Up the Database
- Partitioning CDR and Event Log Tables
- Recovery Model for Microsoft SQL Server

Before You Begin

- Genesys recommends that the VP Reporting Server is installed before you or your database administrator create a database in your Database Management System (DBMS).
- Ensure that a fully functional instance of the Microsoft SQL Server or Oracle exists in your deployment.
- Create the Reporting Server database.

> Important
>
> This section describes the creation of the database schema only. The setup of the Microsoft SQL Server or Oracle 10g instances is outside the scope of this document. For information about setting up these instances, see the vendor documentation.

- For installations using an Oracle database, the database administrator should grant the following system privileges to the user(s) who will own the Reporting Server schema:
  - CREATE TRIGGER
  - CREATE SEQUENCE
  - CREATE TABLE

- CREATE PROCEDURE
- FORCE TRANSACTION
- CREATE VIEW
- CREATE SESSION
- UNLIMITED TABLESPACE

Setting Up the Database

This section includes a description of a generic procedure to create the database schemas.

> **Important**
>
> Genesys recommends that the user setting up an Oracle database be a different user than the one used to create the Configuration Server database.
>
> The table below contains the paths to the scripts that are used to create the database schema. Select the path to the script that matches the edition (standard or enterprise) of the database that you selected during installation of the Reporting Server component.
>
> A minor change is required when Reporting Server is deployed with an Oracle RAC database. When the `hibernate.remote.database` configuration option is used, the Reporting Server internally appends some parameters to the value of the `hibernate.remote.url` option, including the value of the `hibernate.remote.database` option. Therefore, ensure the `hibernate.remote.url` option is properly configured for use with Oracle RAC by configuring the `hibernate.remote.database` option with the value blank.

Table 30: Database Script Files

Script file	Default path
mssql_schema.sql	For Windows: <Installation_Directory>\Program Files\GCTI\gvp\VP Reporting Server 8.5\RS_App_Name\scripts\standard <Installation_Directory>\Program Files\GCTI\gvp\VP Reporting Server 8.5\RS_App_Name\scripts\enterprise
oracle_schema.sql	For Windows: <Install_Dir>\Program Files\GCTI\gvp\VP Reporting Server 8.5\RS_App_Name\scripts\standard <Install_Dir>\Program Files\GCTI\gvp\VP Reporting Server 8.5\RS_App_Name\scripts\enterprise
	For Linux: <Install_root>/opt/genesys/gvp/VP_Reporting_Server_8.5/scripts\standard <Install_root>/opt/genesys/gvp/VP_Reporting_Server_8.5/scripts\enterprise

Important

The Reporting Server supports the partitioning option for Oracle 10g or 11g enterprise version, and SQL Server 2008 enterprise versions. For all other databases, including SQL Server 2005 (enterprise or standard), use the \scripts\standard\ path to find the appropriate scripts.

Reporting Server in Partition Mode on Oracle

When the Reporting Server is installed in partitioned mode (Enterprise) on Oracle by and the default GATHER_STATS_JOB option is used to automatically gather statistics, server performance can be severely affected. In partitioned mode, the Reporting Server database uses rotating staging tables that are flushed throughout the day. These volatile tables are not suited for automatic statistics gathering.

Genesys recommends the following:

- Disable the `GATHER_STATS_JOB` option before you install the Reporting Server database to ensure that inaccurate statistics are not associated with the staging tables.
- Use the provided lock-stats script to lock the staging tables, for example:
  <RS_INSTALLATION_DIRECTORY>/scripts/enterprise/ReportingService/sql/oracle-lock-stats.sql
  This script uses the following Oracle SQL procedure:
  ```
  DBMS_STATS.LOCK_TABLE_STATS (

  ownname VARCHAR2,

  tabname VARCHAR2);
  ```
  …where the `ownname` option must be the name of the schema that is associated with the Reporting Server tables. The script assumes the schema name is REPORTING. You must replace all instances of REPORTING with the proper schema name.

Procedure: Setting Up a Database for the Reporting Server

There are many available query tools that SQL Server and Oracle 10 g can use to execute Structured Query Language (SQL) scripts. For example, SQL Server Management Studio is included in Microsoft SQL Enterprise edition, and Oracle SQL Developer is available on the Oracle website.

Oracle is the only supported database when the Reporting Server DBMS is installed on Linux operating systems.

Microsoft cumulative update packages for SQL Server contain the most recent hot fixes and security fixes. Ensure that the hot fix that is listed as a prerequisite for this procedure is included in the Service Pack that you have installed.

1. Verify that:
   - Microsoft SQL hot-fix build `3175` must be installed.
   - The Microsoft SQL Server Management Studio development tool is installed on the SQL Server.
   - Sun Java Runtime Environment, is installed.
2. On the SQL server, select the Reporting Server database (MSSQL or Oracle) and run the appropriate script. See {{Documentation:GVP:GDG:CGRSD:DRAFT#TBL30DSF|Table: Database Script Files]] for a list of DBMSs and the corresponding name and location of the initialization script files.
3. Open the folder that matches your database type.
4. Load and execute the initialization script that corresponds to your DBMS. For information about how to upgrade the database schemas, see Migrating to GVP 8.5.

Partitioning CDR and Event Log Tables

GVP 8.1.2 and later releases support partitioned CDR and Event Log tables in the Reporting Database. The Resource Manager, Media Control Platform, Call Control Platform, VAR CDR tables, and Event Logs table tend to grow rapidly in large-scale environments. To improve the read/write performance, these database tables can be split into multiple partitions, each of which represents a specific period of time. Partitioning is supported only in the Enterprise editions of Oracle and Microsoft SQL databases.

The Reporting Database can be configured for partitioning in the following ways:

- During the installation of Reporting Server 8.1.2, if the user selects the Enterprise edition of either product, the `rs.partitioning.enabled` configuration option in the persistence section is automatically set to true.
- The `rs.partitioning.partitions-per-day` option in the persistence section is used to change the number of partitions per day. The default value is `6`, however, the value can be increased in environments that experience high call volumes.

> Tip
>
> When database partitioning is enabled, Genesys recommends that you not change the partitioning mode of operation or the number of partitions (even after the

> Reporting Server is started), because of issues that might arise if the database schema or stored data is changed.

- The Reporting Database 8.1.2 schemas are different for enterprise and standard editions and are located in different directories. To create the schema that is compatible with your database selection, see *Procedure: Setting Up a Database for the Reporting Server, on page 287*. For more a complete list of configuration options for database partitioning, see the GVP 8.5 User's Guide.

Recovery Model for Microsoft SQL Server

If you are installing your database on Microsoft SQL Server, Genesys recommends that you use the SQL Server Simple Recovery Model, which is a simple backup that can be used to replace your entire database in the event of a failure or to restore your database to another server. This mode enables you to do a complete backup (all of the data) or a differential backup (data that has changed since the last complete backup only).

This model is a basic recovery model. Every transaction is written to the transaction log, however, the space is reused by new transactions when previous transactions are complete and written to the data file. Since this space is reused, you cannot do a point-in-time recovery. The most recent restore point becomes either the complete backup or the latest differential backup. However, because the transaction log space is reused, the log does get continuously larger, as it does in the Full Recovery Model.

For more information about the types of data this recovery mode is used for and a complete list of backups you can use, see the MSSQL Tips web site.

> Tip
>
> If you choose not to use this recovery mode, Genesys recommends that you backup your transaction logs regularly.

Use one of these two methods to configure the Simple Recovery Model...

METHOD 1

Use T-SQL to enter the following command line:

```
ALTER DATABASE <dbName> SET RECOVERY recoveryOption GO
```

where `<dbName>` is the name of the database.

METHOD 2

Use SQL Server Management Studio:

1. Right-click on database name and select Properties.
2. Select the Options page.
3. Click the Recovery model: drop-down menu and select Simple.
4. Click **OK** to save the configuration.

Procedure: Specify a Non-schema-owner Oracle User on Reporting Server (Running Oracle Partition Mode)

Enable Reporting Server to run in Oracle partition mode, using an application user other than the schema owner. (Please edit and change "password", usernames and table space names if needed.)

Oracle documentation offers a reason for this procedure: System privileges can be very powerful, and should be granted only when necessary to roles and trusted users of the database. It is better to define application users and grant those users the necessary privileges on the schema owners objects.

1. Install your Oracle database.
2. Create Users, role and current schema trigger.
   Note that the schema user's system privileges must be granted directly; not through a role.
3. Run the following statements in a connection as SYSDBA:

```
-- USER SQL
CREATE USER RS_SCHEMA_OWNER IDENTIFIED BY password

DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP";
-- ROLES
-- SYSTEM PRIVILEGES
```

```
GRANT FORCE TRANSACTION TO RS_SCHEMA_OWNER ;
GRANT CREATE TRIGGER TO RS_SCHEMA_OWNER ;
GRANT CREATE SEQUENCE TO RS_SCHEMA_OWNER ;
GRANT CREATE TABLE TO RS_SCHEMA_OWNER ;
GRANT CREATE VIEW TO RS_SCHEMA_OWNER ;
GRANT CREATE SESSION TO RS_SCHEMA_OWNER ;
GRANT UNLIMITED TABLESPACE TO RS_SCHEMA_OWNER ;
GRANT CREATE PROCEDURE TO RS_SCHEMA_OWNER ;
/

-- USER SQL
CREATE USER GEN_REPORTING_APP_RS IDENTIFIED BY
password
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP";
-- ROLES
-- SYSTEM PRIVILEGES
GRANT CREATE SESSION TO GEN_REPORTING_APP_RS ;
GRANT UNLIMITED TABLESPACE TO
GEN_REPORTING_APP_RS ;
/
CREATE ROLE GEN_REPORTING_RW_ROLE
/
GRANT GEN_REPORTING_RW_ROLE TO GEN_REPORTING_APP_RS
/
CREATE OR REPLACE TRIGGER
GEN_REPORTING_APP_RS.AFTER_LOGON_TRIGGER
AFTER LOGON ON GEN_REPORTING_APP_RS.SCHEMA
BEGIN
EXECUTE IMMEDIATE 'ALTER SESSION SET
current_schema=RS_SCHEMA_OWNER';
END;
/
```

4. Run the Reporting Server script in
   `RS_INSTALL_ROOT/scripts/enterprise/`
   `oracle-schema.sql`
   using the schema owner's connection.
5. Run the following statements, using the schema owner's connection:

```
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CUSTOM_VARS_FUTURE TO GEN_REPORTING_RW_ROLE ;
```

```
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CUSTOM_VARS_STAGE_1 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CUSTOM_VARS_STAGE_2 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CUSTOM_VARS_STAGE_3 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
DATECOMPONENTS TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
EVENT_LOGS_FUTURE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
EVENT_LOGS_STAGE_1 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
EVENT_LOGS_STAGE_2 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
EVENT_LOGS_STAGE_3 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LAST_ARRIVAL_COUNTED_TABLE TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LAST_PARTITIONED TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LAST_PARTITIONED_ERROR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LAST_PROCESSED_JMS_MESSAGE TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LATENCIES TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LATENCY_CONFIGURATIONS TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LASTCUTOFFRECORD TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
LASTSUMMARIZED TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_TREAT_NV_COUNTS_FUTURE TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_TREAT_NV_COUNTS_STAGE_1 TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_TREAT_NV_COUNTS_STAGE_2 TO
```

```
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_TREAT_NV_COUNTS_STAGE_3 TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_VXML_COUNTS_FUTURE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_VXML_COUNTS_STAGE_1 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_VXML_COUNTS_STAGE_2 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_VXML_COUNTS_STAGE_3 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_COUNTED_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_IVR_ACTION_STATS_30MIN TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_IVR_ACTION_STATS_5MIN TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_IVR_ACTION_STATS_DAY TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_IVR_ACTION_STATS_HOUR TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_IVR_ACTION_STATS_MONTH TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_IVR_ACTION_STATS_WEEK TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_RESULT_STATS_30MIN TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_RESULT_STATS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_RESULT_STATS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_RESULT_STATS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
```

```
PERIOD_RESULT_STATS_MONTH TO
GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PERIOD_RESULT_STATS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
TIMECOMPONENTS TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_CDR_STAGE_1_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RM_CDR_STAGE_1_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RM_CDR_STAGE_2_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RM_CDR_STAGE_3_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RM_CDR_FUTURE_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RM_CDR_HIST_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_ARRIVALS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_ARRIVALS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_ARRIVALS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_ARRIVALS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_ARRIVALS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_ARRIVALS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_PEAKS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_PEAKS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_PEAKS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_PEAKS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_PEAKS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
PSTNC_PEAKS_WEEK TO GEN_REPORTING_RW_ROLE ;
```

```
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RMDN_ARRIVALS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RMDN_ARRIVALS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
RMDN_ARRIVALS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_STAGE_1_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_STAGE_2_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_STAGE_3_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_FUTURE_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_HIST_TABLE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_ARRIVALS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_ARRIVALS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_ARRIVALS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_ARRIVALS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_ARRIVALS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_ARRIVALS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_DURATIONS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_DURATIONS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_DURATIONS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_DURATIONS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_DURATIONS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_DURATIONS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_PEAKS_30MIN TO GEN_REPORTING_RW_ROLE ;
```

```
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_PEAKS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_PEAKS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_PEAKS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_PEAKS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
ASR_PEAKS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_ARRIVALS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_ARRIVALS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_ARRIVALS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_ARRIVALS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_ARRIVALS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_ARRIVALS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CTIC_PEAKS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CUSTOM_VARS_HIST TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
EVENT_LOGS_HIST TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_ARRIVALS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_ARRIVALS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_ARRIVALS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_ARRIVALS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_ARRIVALS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_ARRIVALS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_EXT_FUTURE TO GEN_REPORTING_RW_ROLE ;
```

```
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_EXT_HIST TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_EXT_STAGE_1 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_EXT_STAGE_2 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_CDR_EXT_STAGE_3 TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_PEAKS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_PEAKS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_PEAKS_DAY TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_PEAKS_HOUR TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_PEAKS_MONTH TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_PEAKS_WEEK TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_TREAT_NV_COUNTS_HIST TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_VXML_ARRIVALS_30MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
MCP_VXML_ARRIVALS_5MIN TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_CDR_EXT_FUTURE TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_CDR_EXT_HIST TO GEN_REPORTING_RW_ROLE ;
GRANT ALTER,SELECT,INSERT,UPDATE,DELETE ON
CCP_CDR_EXT_STAG
```

Maintaining GVP

Learn how to stop, start, and uninstall Genesys Voice Platform components.

- Starting and Stopping the Components
- Uninstalling the Components
- Managing the Cache

Starting and Stopping the Components

Use Genesys Administrator to safely and easily start, stop, and gracefully stop each of the components in a GVP Solution object. A graceful stop causes the Application or Solution object to stop accepting new requests and to wait for all media requests to complete. Media requests, such as call recording or conferences, might take a long time to complete using a graceful stop. However, using a stop abruptly terminates any call recording or conference with no chance of recovery, so using a graceful stop is recommended whenever possible.

Prioritizing the startup of the GVP components is important to ensure it is successful. After the initial installation or any time that the systems are shut down for maintenance, use the startup priority that is outlined in the section Startup Sequence for the VPS.

You can also use Genesys Administrator to configure the components to start automatically.

This section contains the following procedures:

- Starting and Stopping GVP Solution Objects
- Starting and Stopping GVP Application Objects
- Configuring Application Objects to Start Automatically

Procedure: Starting and Stopping GVP Solution Objects

Use this procedure only if you have created Solution Objects (optional).

1. Verify that:
    - The GVP components are installed. See Installing Manually on Windows or Installing Manually on Linux.
    - A Solution object is created (done automatically since version 8.1.2).
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Solutions`.
4. Select the Solution object that you want to start.

5. In the Tasks panel, click the Runtime section down arrow. The section opens to display the options **Start**, **Stop**, and **Graceful Stop**.
6. Select one of these options.

Procedure: Starting and Stopping GVP Application Objects

1. Verify that the GVP components are installed. See Manually Installing GVP on Windows or Manually Installing GVP on Linux.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Select the Application object that you want to start or stop.
5. In the Tasks panel, click the **Runtime** section down arrow.
6. Select one of the options that the section opens to display: **Start**, **Stop**, or **Graceful Stop**.

Graceful Shutdown of the Reporting Server

To minimize the risk of data loss, always shut down components with active client connections to the Reporting Server before shutting down the Reporting Server itself. Components such as MCP, RM, CCP and MRCP Proxy should be shut down gracefully first; only then can the Reporting Server be shut down safely.

If the Reporting Server goes down unexpectedly or unintentionally, restart it and give it some time to process queued-up local data before shutting down client components.

Procedure: Configuring Application Objects to Start Automatically

This procedure presents two different ways to configure the components.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select `Environment > Applications`. The Configuration tab appears.
3. Configure the Application in one of two ways:

   METHOD ONE

   a. Double-click the Application object that you want to configure to start automatically.
      In the Server Info section:
   b. Scroll down to the Auto Restart field.

c. Click the `True` check box to enable it.

METHOD TWO

On the Options tab, from the View drop-down menu:

a. Select Advanced View (Annex).
b. In the `sml` section, click **New**.
The New Option dialog box appears.
c. In the Name field, enter `autostart`.
d. In the Value field, enter `true`.

4. Save the changes.

Uninstalling the Components

Before you begin to uninstall the components, ensure that they are stopped by using the Stop applications gracefully option in Genesys Administrator. Uninstall the GVP components one at a time.

The procedures to uninstall the GVP components manually are included in this section, but uninstalling the components by using Genesys Administrator is recommended.

Procedure: Uninstalling GVP Components by Using Genesys Administrator

1. Before uninstalling a component on Linux, ensure that write permissions are configured on the LCA folder by issuing the following command as root on the server: `chmod a+w /opt/genesys/lca`
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select `Environment > Applications`.
4. Double-click the Application object that you want to uninstall.
The Configuration tab appears.
5. In the tool bar, select **Uninstall**.
A Confirm dialog box appears.
6. Click **Yes**.
A dialog box appears that indicates that the uninstallation process is complete.

Procedure: Uninstalling GVP Components Manually (Windows)

Uninstall GVP manually, one component at a time, on a Windows host.

1. Log on to the host where the component is installed, to uninstall it manually.
2. Stop the Application objects to be uninstalled by using the **Stop GVP Applications Gracefully** option in Genesys Administrator. See Procedure: Starting and Stopping GVP Application Objects.
3. From the Start menu, select `Control Panel > Add/Remove Programs`.
4. Select the appropriate GVP component from the list of currently installed programs.
5. Click **Remove**.
6. When the uninstall is complete for each of the GVP components, restart the machine.

Procedure: Uninstalling GVP Components Manually (Linux)

1. Log on to the Linux host where the component is installed.
2. Stop the components by using the **Stop GVP Applications Gracefully** option in Genesys Administrator. See Procedure: Starting and Stopping GVP Application Objects.
3. Delete the installation directory.

Managing the Cache

This section describes ways in which you can manage the Squid and Page Collector cache manually on Windows and Linux hosts, see:

Squid Cache Management

The table below summarizes the commands that you can use to force the cache to be refreshed, purged, or cleared. 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: Manual Cache Management Commands

Objective	Command
Windows OS	
Refresh an object.	`C:\squid\bin\squidclient -s -r <uri>` Where `<uri>` is the full URI of the object that you want to refresh.
Purge an object.	`C:\squid\bin\squidclient -s -m PURGE <uri>`

	Where <uri> is the full URI of the object that 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
Linux OS	
Refresh an object.	/usr/local/squid/bin/client -s -r <uri> Where <uri> is the full URI of the object that you want to refresh.
Purge an object.	/usr/local/squid/bin/client -s -m <PURGE> <uri> Where <uri> is the full URI of the object that you want to refresh.
Clear the entire cache.	/usr/local/squid/bin/squid -k shutdown echo "" > /usr/local/squid/cache/swap.state /usr/local/squid/bin/squid

For more information about how GVP handles caching, see Caching.

Rotating the Caching Logs (Windows)

Schedule a daily task in Windows Scheduler to rotate the logs for the Squid caching service. GVP does not rotate the logs automatically because Squid caching is a third-party application.

Tip

In GVP version 8.1.2 and above, the Squid Caching Proxy automatically rotates the caching logs, therefore, this procedure is only required for GVP 8.1.1 and earlier 8.x releases.

Procedure: Scheduling the Caching Logs Rotation (Windows)

Schedule a daily task to rotate the Squid caching service logs on Windows.

1. Verify that the Squid caching proxy is installed and service is running. See Manually Installing GVP on Windows.
2. From the Windows Start menu, select `All Programs > Accessories > Notepad`.
3. Enter the following script:

```
@echo
C:\squid\sbin\squid.exe -k rotate -n SquidNT
@pause
@echo
```

4. Save the file with the extension `.bat` for example, `SquidTask.bat`.
5. From the Windows Start menu, select **All Programs** > > **System Tools** > **Scheduled Tasks**.
6. Double-click **Add Scheduled Task**.
   The Scheduled Task Wizard appears.
7. Click **Next** to browse to the `.bat` file you created in an earlier step.
8. Double-click the file.
   The Scheduled Task Wizard automatically populates the Task Name field.
9. In the Perform this task: section, select **Daily**.
10. Click **Next** and enter `2:00 AM` in the Start Time field.
11. Select the **Every Day** radio button.
12. In the Start Date field, enter the date that you want the task to start for example, 5/12/2008.
13. Click **Next** to enter the username of the person who is scheduling the task
14. In the Password and Confirm Password fields, enter the password.
15. Click **Next** to finish and quit the wizard.

Tip

To uninstall the log rotation schedule, delete the scheduled task.

Scheduling the Caching Logs Rotation (Linux)

You can configure a rotation schedule for the Squid caching logs on Linux using the /etc/logrotate.d/squid file. The default configuration is to rotate the logs weekly, retain the last five files, and compress each archived file, however the file can be modified to suit your needs.

[+] This script is a typical configuration in the Squid log rotation file:

```
/var/log/squid/access.log {
weekly
rotate 5
copytruncate
compress
notifempty
missingok
}
/var/log/squid/cache.log {
weekly
rotate 5
copytruncate
compress
notifempty
missingok
}
/var/log/squid/store.log {
weekly
rotate 5
copytruncate
compress
notifempty
missingok
# This script asks squid to rotate its logs on its own.
# Restarting squid is a long process and it is not worth
# doing it just to rotate logs
postrotate
/usr/sbin/squid -k rotate
endscript
}
```

For more information about the logrotate capabilities of Linux, check the vendor documentation or visit the website.

Page Collector Cache Management

Purge the Page Collector cache manually by configuring the Media Control Platform Application object in Genesys Administrator.

Procedure: Purging the Page Collector Cache Manually

Configure the Media Control Platform to purge the Page Collector cache the next time that the server is restarted.

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select `Environment > Applications`.
3. Double-click the Media Control Platform Application object that you want to configure.
   The Configuration tab appears.
4. On the Options tab, from the View drop-down list:
   a. Select `Advanced View (Options)`.
   b. In the PageCollector section, set value of the `PurgeCache` parameter to `1`; for example, `PurgeCache = 1`.
   c. Save the changes.

Resource Manager High Availability

This topic describes how to configure the Genesys Voice Platform (GVP) Resource Manager and the Configuration Database for High Availability (HA) on Windows and Linux operating systems.

Overview

High availability for Resource Manager is achieved by combining two Resource Manager instances in an HA-pair, which enables the resource-management function to be distributed over two servers to provide redundancy. The active Resource Manager node (running on server 1) simultaneously handles the incoming IP traffic and updates the other Resource Manager node (in the HA-pair) with information about the active sessions, thereby achieving high availability and scalability.

HA prevents single point-of-failure and provides immediate failure recovery. In addition, administrators can manage the HA-pair as a single system either locally or remotely.

HA Modes

To configure the Resource Manager Application in HA mode, use the `cluster.ha-mode` parameter to set the value to active standby, active active, or None. A value of None, which is the default, means the Resource Manager is in stand-alone mode. Two HA modes exist for Resource Manager:

- **Active Standby:** Where only the active instance processes SIP requests and the backup instance processes SIP requests when the active instance fails. Two Resource Manager instances share the same virtual IP address but only one instance is actively receiving network traffic.

  To configure the Resource Manager in active standby mode, see Task Summary: Configuring the RM in HA Active Standby Mode (Windows) and Configuring the RM in High Availability Active-Standby Mode (Linux)

- **Active Active:** Where either one of the active nodes can process SIP requests. This mode is similar to the virtual IP solution; however, an external load balancer maintains the virtual IP address that forwards information to the active nodes in the cluster. The load balancer can apply proprietary load-balancing rules when it forwards the requests.

  The virtual IP address for the active active HA-pair is configured by using the cluster.virtual-ip parameter in the Resource Manager Application. The Resource Manager advertises the virtual IP address in the Record-Route and Via headers so that SIP Server or the Media Control Platform can send mid-call requests and responses to the load balancer.

  To configure the Resource Manager in active active mode, see Task Summary: Configuring the RM in HA Active-Active Mode (Windows) and Configuring the RM in High Availability Active-Active Mode (Linux)

Tip

When configured as an HA-pair, both Resource Manager instances must manage the same set of resources.

In addition, the load balancer must have the routing capability to route the same SIP transactions to the same Resource Manager instance.

(The load balancer must be SIP-aware and support the Call-ID stickiness feature to ensure this requirement is met.) The same Resource Manager node must process the SIP transaction in its entirety.

Before You Begin

Before you begin the procedures in this section, pay attention to the following information:

Ensure that the Resource Manager hosts...
...are configured and fully functional in stand-alone mode.
...reside on the same subnet.
...have at least two network interface cards (NICs) that are configured with unique IP addresses (within the same subnet).

...have one virtual IP address allocated for the cluster (to be shared by the Resource Manager hosts in the HA-pair)

Tip

If you are configuring HA on Linux by using the Simple Virtual IP failover option, only one NIC is required. However, to avoid a single point-of-failure if the NIC or network switch fails, you must use the Bonding Driver failover option with two NICs. The bonding driver is configured to share one IP address among two NICs operating one NIC as active and the other as standby. See Resource Manager HA (Linux). If you configure active active HA on either Windows or Linux, only one NIC per server is required.

- If you are deploying the Supplementary Services Gateway (SSG) on the same host as the Resource Manager, be aware of a possible port conflict. The default HTTPS Port for the Supplementary Services Gateway is 9801 (controlled by the HTTPS Port parameter in the HTTP section), which is also the default port value for the Resource Manager member.1 and member.2 cluster parameters when they are configured for HA. Use Genesys Administrator to change the values, if required.
- When you configure the Resource Manager for active standby HA, five IP addresses are required. (If you are using the Simple Virtual IP failover option, only three IP addresses are required.) The first network adapter handles the network traffic that is addressed to the server as part of the HA-pair, while the second network adapter is used for intrahost communication. Figure 32 depicts an example of how IP addresses are assigned to the servers.

- Simple Network Management Protocol (SNMP) Management Information Bases (MIB) function properly when there is only one instance of a GVP component on a host. It is not recommended that you install the SNMP MIBs on a server that is hosting more than one instance of the Resource Manager. SNMP traps are sent by the active Resource Manager only.

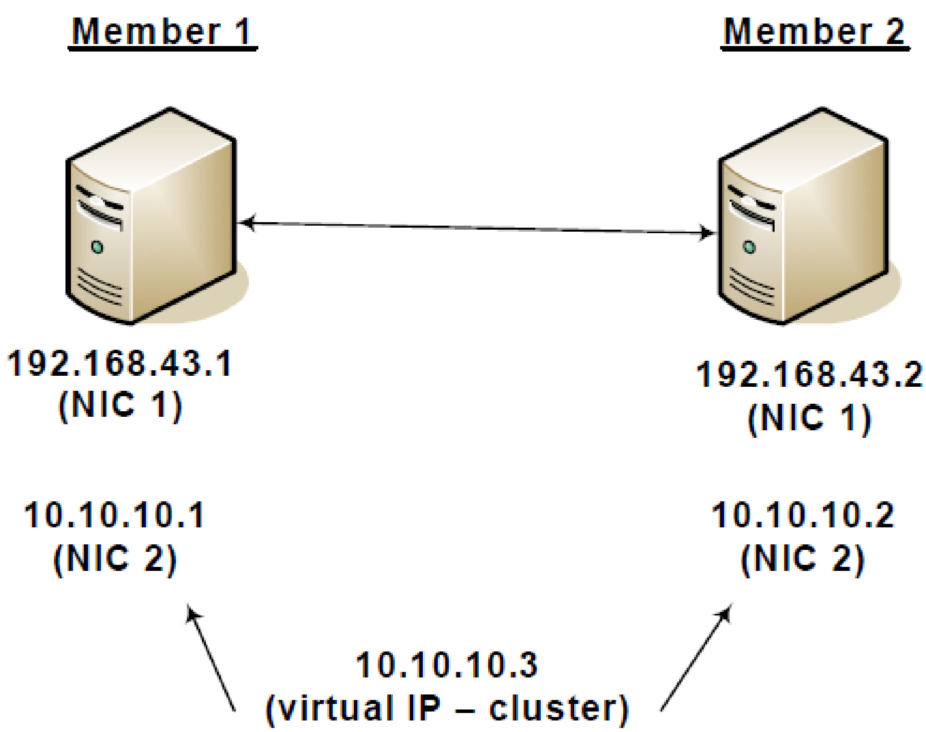


Figure: IP Address Assignation for active standby HA

Monitoring the NICs

When the Resource Manager is configured as an active standby HA-pair, it monitors the NICs to determine network error instances for example, when a cable is unplugged. If any of the NICs encounters errors, the network is considered to be down.

The Resource Manager must monitor the NICs that are used in the HA-pair to ensure that the Network Load Balancing (NLB) is working properly:

- **For Windows:** One with NLB configured, and one with the IP address of the host computer.
- **For Linux:** If only one NIC is used in the HA-pair, it must be monitored.

When there are more than two NICs present in the system and the other NICs are not part of the NLB HA-pair, it is recommended that you specify the Media Access Control (MAC) addresses of the NICs that you want to monitor.

To specify the NICs that you want to monitor, use the steps in Procedure: Specifying the NICs to Monitor (Windows), and Procedure: Specifying the NICs for Monitoring (Linux).

Tip

When Windows NLB is used, the standby Resource Manager can take up to seven (7) seconds to become active when failover occurs. However, there is no data loss, even for those calls that are in progress.

Integrating GVP with SIP Server for an Active-Active Resource Manager Configuration

For all GVP deployments with SIP Server that will support Resource Manager High Availability Active-Active in IP modes, configure a VoIP service DN.

[+] GVP as a Self-service IVR

In self-service mode, GVP works as an IVR for handling inbound calls that are forwarded by SIP Server. Calls can be self-service (where the call terminates at GVP), or assisted service (where GVP transfers the call to an agent as needed).

Configure GVP as a Trunk DN in SIP Server:

Table: Trunk DN Configuration (Inbound)

Parameter	Value
contact	`"::msml"`
prefix	Initial digits that match the DNIS number
sip-busy-type	`2`

Geo-location (optional)	`<location1, location2>`

For HPE, configure GVP as a Trunk Group DN in SIP Server:

Table: Trunk Group DN Configuration (Outbound)

Parameter	Value
contact	`"::msml"`
subscription-id	`<DN_Name>`
make-call-rfc3725-flow	`1`
refer-enabled	`false`
ring-tone-on-makecall	`false`
request-uri	`sip:msml@<RMHost>:<RMport>;gvp-tenant-id=[<Tenant-Name>]`

[+] GVP as an Inbound Media Server

In this mode, SIP Server is an application server and GVP is a media server performing requests such treatment, greeting, music, conference, and recording.

Configure GVP as a VoIP Service DN in SIP Server:

Table: VoIP Service DN Configuration Parameters

Name	Value	Type
contact-list (for IP mode)	Resource Manager's contacts	Mandatory
service-type	`msml`	Mandatory
oos-check	`<time in seconds>` (Active OOS functionality)	Mandatory
oos-force	`<time in seconds>`	Mandatory
geo-location	`<location1, location2>`	Optional

subscription-id	<tenant_name>	Optional

Parameter Notes

Contact-list

Use `contact-list` when SIP-Server Resource Manager High Availability Active-Active (SIP-S RM HA A-A) is deployed in IP mode.

`Contact-list` contains comma-separated URIs,and is used when SIP-S RM HA A-A is deployed in IP mode.

A new DN level parameter contact-list is introduced for supporting Multiple IP address configuration. This parameter will have comma-separated list of any valid SIP URI. Configure each URI using following format:
`[sip/sips:][number@]hostport[;transport=(tcp/udp/tls)]`

Where:

- *sip/sips:* is an optional prefix.
- *number* is the DN number (currently ignored.)
- *hostport* is a host:port pair, where host is a dotted IP address for the endpoint.

Tip

Media Server mode uses the same VOIP service DN that is used to configure Resource Manager contacts. This mode requires Media Server status notification functionality, so the subscription-ID parameter must be added in the VOIP service DN.

[+] GVP as an Outbound Media Server

In this mode, GVP is a media server providing outbound services such as Call Progress Analysis, music treatment, media bridging (ASM mode) and playing the VXML application.

Configure GVP as a Trunk Group with these parameters:

Parameter name	Parameter value
contact	"::msml"
subscription-id	*<DN_Name>*
make-call-rfc3725-flow	1
refer-enabled	false
ring-tone-on-makecall	false
request-uri	sip:msml@<*RMHost*>:<*RMport*>media-service=cpd;gvp-tenant-id=[<Tenan

In deployments of CTI through an IVR Server (used for playing VoiceXML applications after the outbound call is established), configure GVP as a set of Voice treatment port DNs, with this parameter:

Parameter name	Parameter value
contact	"::msml"

For Voicemail Integration, you must configure a VoIP service DN along with the msml VoIP service DN. Configure the voicemail VoIP service DN with these parameters:

Parameter	Value
contact	"::msml"
service-type	voicemail

[+] Limitations of the Active-Active RM Configuration

Integration of GVP with SIP Server for RM HA (Active-Active) is not supported with DNS SRV as RM Virtual-IP or SIP-S contact.

SIP Server supports Resource Manager configuration in these ways:

- Multiple IP addresses (IP mode) In IP mode, SIP Server does not support configuring priority and weightage for each IP address. All entries are considered of equal priority and weight. Set cluster.ha-mode to active-active.

SIP Server supports only the round-robin load-balancing method to support Resource Manager HA Active-Active.

See also in the SIP Server Deployment Guide:

- *Configuring an MSML Service* in Chapter 4: SIP Devices Support.
- *Genesys Voice Platform Integration* in Chapter 5: SIP Server Feature Support.

When SIP Server is configured for Resource Manager High Availability Active-Active, all the calls going to the RM will be in either TCP or UDP format, based on what transport is specified in the VoIP service DN.

Notes on Resource Manager Configuration for Active-Active (Load Balancing)

Set these options on each Resource Manager:

- Set `cluster.ha-mode` to `active-active`.
- Set `cluster.virtual-ip` to the RM IP address.
- Set `Cluster.member.1|2` to the corresponding RM IP:Port.
- On each MCP, set `transport.staticroutelist` to RM Node1 contact, RM Node2 contact.
- In the sip section: `routeset=<sip:ipaddress:port;lr>` must be one of the Resource Managers.

Configure half of the MCP's route-set to point to one RM, and the other half of the MCP's route-set to point to the other RM.

- In the vrmrecorder section: `routeset=<sip:ipaddress:port;lr>` must be one of the Resource Managers.

Configure half of the MCP's route-set to point to one RM, and the other half of the MCP's route-set to point to the other RM.

> Tip
>
> The configuration parameter transport.staticroutelist is also applicable for other SIP components such as CTIC and CCP.

Resource Manager HA (Windows)

Task Summaries and Procedures

Task Summary: Configuring the RM in HA Active Standby Mode (Windows)

Objective	Related Procedures and Actions
1. Configure NLB on the Resource Manager hosts.	See Procedure: Configuring Resource Manager HA (Windows 2008).
2. Configure the member IDs and NLB script path in the Resource Manager Applications.	See Procedure: Configuring the Resource Manager HA-Pair.
3. Configure the virtual IP address of the HA-pair in the INIT and NLB script files.	See Procedure: Configuring the INIT and NLB Script Files (Windows).
4. Specify the NICs that require monitoring (optional).	See Procedure: Specifying the NICs to Monitor (Windows). **Note:** In Windows environments, NICs monitoring is optional. If there are only two NICs installed on the host, omit this procedure. For more information about monitoring the NICs, see Monitoring the NICs.
5. If you are installing Resource Manager HA on Windows 2008, configure a network account with Administrator privileges.	See Procedure: Configuring the Resource Manager Service (Windows). **Note:** Windows 2008 does not support the NLB command /PASSW argument for remote procedure calls. Therefore, the Resource Manager Service

	must run as a network account that has Administrator privileges.
6. Complete finals steps before starting the Resource Manager HA-pair in NLB mode.	See Start Resource Manager HA-pair in NLB mode.

Task Summary: Configuring the RM in HA Active Active Mode (Windows)

Objective	Related Procedures and Actions
1. Configure the member IDs in the Resource Manager Applications.	See Procedure: Configuring the Resource Manager HA-Pair.
2. Configure the virtual IP in the Media Control Platform, Call Control Platform, and CTI Connector Applications.	See Procedure: Integrating Application Objects with Resource Manager and Procedure: Configuring the Call Control Platform. **Note:** When you use these procedures to configure active active HA mode, the virtual IP is used as the Resource Manager IP.
3. Configure the external load balancer.	See the vendor documentation for the type of load balancer you are using (for example, F5 or Radware).

You can configure the Resource Manager in HA active standby mode by using the Windows NLB service. Use the procedures in this section to configure the Resource Manager HA-pair on Windows 2008.

Procedure: Configuring Resource Manager HA (Windows 2008)

To set up NLB by configuring Windows 2008, perform this procedure on each of the Resource Manager hosts in the NLB cluster, specifying a unique ID for each host.

1. Verify that the Resource Manager hosts must conform to the prerequisites for Windows. See Prerequisites.
2. From the Windows Start menu, select **Administrative Tools > Server Manager**.

3. In the Feature Summary section, click **Add Features**.
   The Add Features Wizard appears.
4. Click the check box beside **Windows Network Load Balancing**.
5. Click **Install**.
   Network Load Balancing is installed on Windows.

Configuring the Cluster

6. In Administrative Tools, select **Network Load Balancing Manager**.
7. Right-click **Network Load Balancing Clusters**, and click **New Cluster**.
8. In the Host field, enter the name of the Resource Manager host that you are adding to the cluster for example, `ResMgr1`.
9. Click **Connect**.
10. Select the interface that will host the HA-pair's virtual IP address, and click **Next**. The interface selected cannot be used for the private communication between the Resource Manager nodes (for example, the IP address that is associated with this NIC cannot be used in the `[cluster] member.[n]` configuration parameter). This interface hosts the virtual IP address, which receives and load-balances the client traffic.
11. Enter the information on the Host Parameters and Cluster Parameters section, as shown in the table below:

Table: Properties of NLB Service

Section	Field	Description
Host parameters	Priority (unique host identifier)	Enter `1` for the first Resource Manager host in the cluster. Enter `2` for the second Resource Manager host in the cluster. This parameter specifies a unique ID for each host.
Cluster IP Address		Click **Add** to enter the IP address that is shared by the hosts in the HA-pair. The shared IP address for the HA-pair must be static. NLB disables DHCP on every interface that it configures, as it does not support DHCP.

Cluster parameters	IP Address	Enter the virtual IP address of this cluster.
	Subnet mask	Enter the subnet mask for your network. (The subnet mask is not required for IPv6 addresses.)
Cluster operation mode	Unicast	Enable this radio button. In Unicast mode, the MAC address of the cluster is assigned to the network adapter for the computer, and the built-in MAC address of the network adapter is not used.

12. In the Port Rules section, click **Edit**.
13. Configure the port rules as shown in the table below:

Table: Port Rules Configuration

Section	Field	Value and Description
Port range	From	Accept the default value, `0`.
	To	Accept the default value, `65535`.
Protocols		Select **Both (TCP & UDP)**.
Filtering Mode	Multiple hosts	Enable this radio button.
	Affinity	Set to None.

14. Click **Finish**.
To add another host to the cluster, right-click the new cluster, click **Add Host to Cluster**, and repeat Steps 9 to 15.
15. Next, configure the Resource Manager Applications for HA. See Procedure: Configuring the Resource Manager HA-Pair.

Tip

The following information applies to NLB configuration on Windows 2008:

In active standby mode, when the active Resource Manager nodes NLB-dedicated NIC cannot be reached (due to an unplugged cable, a disabled

NIC, or a shutdown host), it can take several seconds to several minutes before the traffic is re-routed to the standby Resource Manager node.

When the active Resource Manager node cannot be reached, the standby node issues the wlbs command (see Procedure: Configuring the INIT and NLB Script Files (Windows)) as part of the failover sequence. If the dedicated NLB NIC of the currently active Resource Manager node cannot be reached, then the wlbs command can hang for several seconds and cause the failover to be delayed.

In addition, if the failover occurred because the Resource Manager machine was shut down, the previously active Resource Manager might temporarily take over the traffic when the machine reboots. To resolve this issue:

1. In the Network Load Balancing Properties on both of the Resource Manager hosts in the cluster, go to the Host Parameters section.
2. In the Initial host state section, select Stopped from the Default state drop-down menu.

Procedure: Configuring the Resource Manager HA-Pair

Complete this procedure for each Resource Manager HA Application in the HA-pair. You will configure the member IDs and NLB script path in the Resource Manager HA Applications for active standby mode.

1. For active standby mode only, ensure NLB clustering is set up on each Resource Manager host in the cluster. See Procedure: Configuring Resource Manager HA (Windows 2008), or Procedure: Configuring Simple Virtual IP Failover, or Procedure: Configuring Bonding Driver Failover.
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select **Environment > Applications**.

4. Select the Resource Manager HA Application you want to configure.
   The Configuration tab appears.
5. On the Options tab, enter the mandatory information in the Cluster section as shown in the table below:

Table: Options Tab Cluster Section

Option Name	Value
members	Retain the default value of 1 2.
member.1	Enter <localhost_IP_add_1>:9801 ...where <localhost_IP_add_1> is the private IP address of the first Resource Manager host, which corresponds to the IP address of the network interface that does not have the virtual IP address assigned (not the IP address that is associated with the NLB cluster).
member.2	Enter <localhost_IP_add_2>:9801 ...where <localhost_IP_add_2> is the private IP address of the second Resource Manager host, which corresponds to the IP address of the network interface that does not have the virtual IP address assigned (not the IP address that is associated with the NLB cluster).
hotstandby	Enter true.
mymemberid	◦ For the Resource Manager HA Application that represents the first Resource Manager host in the HA-pair, enter 1. ◦ For the Resource Manager HA Application that represents the second Resource Manager host in the HA-pair, enter 2. The first and second Resource Manager hosts must correspond to the first and second Resource Manager hosts that you specified in Step 5 or the table Properties of NLB Service in Procedure: Configuring the INIT and NLB Script Files (Linux). Also, if both Resource Manager instances are running, memberid 2 will be the active one.

`virtual-ip`	Enter <virtual_IP_add>, where <virtual_IP_add> is the designated Virtual IP address that is shared by all of the Resource Manager hosts in the HA-pair.
`virtual-ip-in-via`	Retain the default value of `true`.
`electiontimer`	Retain the default value of `3000`.
`FailOverScript`	Retain the default value, <Installation Directory>\bin\NLB.bat where <Installation Directory> is the directory where the NLB.bat file is installed. Tip Configuration of this option is not required for active active HA mode.
`heartbeattimer`	Set to `2000`.
`ha-mode`	Enter active active or active standby.

Tip

Many other options can be configured for the Resource Manager HA-pair. For a complete list of the available options, and descriptions of them, see the GVP 8.5 User's Guide.

6. Click **Save**.
7. Repeat Steps 3 to 5 for each Resource Manager HA Application in the HA-pair.
8. If you have not already done so, configure a connection to the Message Server in each Resource Manager Application in the HA-pair. See Procedure: Creating a Connection to a Server.

9. Specify the NICs you want to monitor (optional). See Procedure: Specifying the NICs to Monitor (Windows).

Procedure: Specifying the NICs to Monitor (Windows)

Specify the NICs that you want the Resource Manager to monitor. If the GVP section in the Resource Manager HA Application is not configured, all of the NICs installed on the host are monitored for network errors.

1. More than two NICs are configured on the same host and are fully functional.
2. Two NICs are configured as part of an HA-pair. See Procedure: Configuring Resource Manager HA (Windows 2008) and Procedure: Configuring the Resource Manager HA-Pair.
3. Log in to Genesys Administrator.
4. On the Provisioning tab, select **Environment > Applications**.
5. Select the Resource Manager HA Application that you want to configure.
6. On the Options tab, scroll to the GVP section.
7. For the `nic.eth0` option, in the Value field, enter the MAC address of the first NIC that you want to monitor. For example: `nic.eth0 = 00-0F-1F-6D-EB-CA` (*this example assumes chosen network interfaces numbered 0 and 1*).
   This value must be provided for IP takeover, even if when a single NIC is used.
   - Repeat this step, to add `nic.eth1` and the MAC address of the second NIC that you want to monitor. For example: `nic.eth1 = 00-0F-1F-6D-EB-CA` (*this example assumes chosen network interfaces numbered 0 and 1*).
   - NICs must be configured with a value of 0 (zero) for IP takeover, even though a single NIC is used.
   - If more than two NICs exist, configure the nics option value to `0 1`.
8. Click **Save**.
9. To confirm that you have configured the NICs correctly, use the ipconfig/all command to query the MAC addresses of the NICs.
10. Configure the INIT.bat and NLB.bat script files. See Procedure: Configuring the INIT and NLB Script Files (Windows).

Procedure: Configuring the INIT and NLB Script Files (Windows)

Configure the INIT.bat and NLB.bat files on each Resource Manager host in the HA-pair.

1. Verify that NLB clustering has been set up on the hosts. See Procedure: Configuring Resource Manager HA (Windows 2008).

Configure the INIT.bat file

2. Open the INIT.bat file in a text editor.
   The INIT.bat file is located in `<Res_Mgr_Install_Dir>\bin` directory.
3. Follow the directions in the script file to change the virtual IP address and the IP address for members 1 and 2 of the HA-pair.
4. Click **File** > **Save**.
   Configure the NLB.bat file
5. Open the `NLB.bat` file in a text editor. The NLB.bat file is located in the `<RM_Install_Dir>\bin` directory.
6. Follow the directions in the script file to change the virtual IP address and the IP address for members 1 and 2 of the HA-pair. The `private_ip_member1` and `private_ip_member2` parameters represent the interfaces that are associated with the NLB interface.
7. Save the changes.
8. Execute the `INIT.bat` script on each host to disable NLB functionality on both hosts, enter `INIT.bat`.
9. Execute the NLB.bat script to re-enable NLB functionality on the host that will act as the master, enter `NLB.bat enable X`
   ...where X is the member ID of the host on which the virtual IP will accept traffic. Enter `1` or `2` for the value in the NLB.bat enable command, based on the member ID of the Resource Manager instance on that host.
   After NLB.bat script execution, the virtual IP will be active on the host that is identified as `private_ip_member1` in the NLB.bat file. Confirm this by attempting to Remote Desktop into the virtual IP address, and once logged in, check the hostname to confirm it is the correct system.
10. If you are installing the cluster on Windows 2008, configure the Resource Manager Service. See Procedure: Configuring the Resource Manager Service (Windows).
11. If you are installing the cluster on Windows 2003, execute NLB cluster mode. See Start Resource Manager HA-pair in NLB mode.

Procedure: Configuring the Resource Manager Service (Windows)

Modify the Resource Manager Service in Window 2008 to run as a network account. Complete this procedure on both servers (primary and backup) in the HA-pair.

1. At the Start menu, select **Control Panel > Administrative Tools > Services**.
2. Right-click the **Genesys VP Resource Manager Service**.
3. When the Genesys VP Resource Manager Service Properties dialog box opens, click the **Log On** tab.

4. Enable the **This account** radio button, and enter `.\Administrator`.
5. In the Password and Confirm Password fields, enter the Administrator's password.
6. Click the **Enable** button.
7. Click **OK**.
8. Start Resource Manager HA-pair in NLB mode (below).

Procedure: Starting Resource Manager HA-pair in NLB mode

1. Execute the `INIT.bat` file.
   When you execute this file, load balancing is disabled on both members in the HA-pair. Executing in NLB mode, the Resource Manager checks the status of other Resource Manager instance before it assumes Active Standby status.
2. Start both Resource Manager instances, running as a High Availability pair.

Resource Manager HA IP Address Takeover for Windows

Beginning with release 8.1.6, Resource Manager supports IP Address Takeover for Windows.

Resource Manager (RM) is used with SIP Server for Media Server applications, and with GVP for VoiceXML applications. Resource Manager provides an essential intelligence for GVP and media policy management, resource management and reporting. Resource Manager monitors the availability of media servers and directs SIP Server to connect sessions to the most suitable media server.

Resource Manager High Availability means that the RM function can be deployed as a pair of RM processes. You can configure RM pairs to send updates to each other regarding the status of requests and system states. RM pairs can be deployed in Active-Standby or Active-Active modes.

If SIP Server is sending requests to an RM Active-Active configuration, it requires that a load balancer be placed between SIP Server and the Resource Manager Pairs. F5, NLB, and Radware are examples of third party load balancers that can perform these functions. Active-Active designs require load balancing to maintain stickiness of the session between SIP Server and the RM selected.

If SIP Server is sending requests to an RM pair configured in Active-Standby, then SIP Server is directing requests only to the active RM, using a Virtual IP. The RMs have an internal selection mechanism to determine which node should be the active one. The standby RM is used to take over the role of the Primary only when the current active RM process fails. There are two ways to accomplish the takeover: you can use Windows NLB for monitoring and switching.

Or, you can use the Genesys Solution Control Server (SCS) to monitor alarms sent to it for RM; if the active RM goes down, then SCS can execute scripts that change the Virtual IP addressing between SIP Server to the formerly-standby-now-active RM. The RM also has its internal mechanism of performing failover using heartbeat monitoring between the pair. The active standby configuration does not require a load balancer, but does need an effective script solution.

Table: IP Address Takeover vs. Load Balancing A Comparison

RM Configuration	RM Scripts	RM Load Balancing	Pro	Con
Active-Standby (Load Balancing)	—	Windows NLB	Comes with the product, slightly easier config; supports Windows.	—
Active-Standby (IP Address Takeover)	IP-Takeover Patch with Scripts	—	Comes with the product, slightly easier configuration; supports Windows & Linux.	Still less reliable than NLB in this configuration for switchover timing; see the Resource Manager Release Note.
Active-Active (IP Address Takeover)	—	F5, NLB	Fast takeover.	Complex Config, 3rd Party sw, NLB is windows only.
Active-Active (Load Balancing)	—	Genesys SIP Server with internal load balancing	Comes with the product, easy configuration, baked-in function.	—

Procedure: Configure Resource Manager High Availability Using Virtual IP Address Takeover for Windows

1. New script files were added to the Resource Manager IP. Verify that the following four files are present in the installation-bin folder:

    - `INIT_IPTakeOver.bat`
    - `IPTakeOver.bat`
    - `Ping.vbs`
    - `Check_Ip.vbs`
2. Follow the instructions inside `INIT_IPTakeOver.bat` to set the parameters `VirtualIP` and `VirtualInterface`.
3. Follow the instructions inside `IPTakeOver.bat` to set the parameters `VirtualIP`, `VirtualInterface`, `GatewayIP`, `mymemberid` and `InterfaceForARPing`.

   Tip

   `IPTakeOver.bat` also contains instructions that you should follow, for using the arping utility and other functions.

   Line 82 in `IPTakeOver.bat` should be commented out for Installations on Windows 2008 SP1 with the Hot fix installed. `http://support.microsoft.com/kb/2811463/en-us`: `REM @arping.exe -c 3 -i %InterfaceForArping% -S %VirtualIP% %GatewayIP% >> IPTakeOver.log`

4. In the RM's [cluster] section, set the `failoverscript` parameter to `$InstallationRoot$/bin/IPTakeOver.bat`.
5. Create alarm-based reaction scripts to execute the failover script which would disable VIP in case of RM crash or shutdown. To create these scripts, follow these steps:
    a. Create a new Third Party Server template.
    b. Create two Reaction Applications.
    c. Create and configure two Alarm Reaction scripts.
    d. Create two Alarm Conditions, to send an alarm when either instance of the RM is stopped intentionally.
    e. Create two Alarm Conditions, to send an alarm when either instance of the RM stops unexpectedly.
6. (Optional) Execute INIT_IPTakeOver.bat manually before starting RM in both HA nodes.

> **Tip**
>
> In some systems, the default heartbeat interval between the two RM nodes (2000 msec) is not suitable for the IP Takeover mechanism. To compensate, Genesys recommends setting the configuration option `cluster.heartbeattimer` to `8000`.

7. Virtual IP (VIP) Address Takeover for Windows is less reliable than a Windows NLB cluster configuration (below).

Virtual IP (VIP) Address Takeover for Windows is less reliable than a Windows NLB cluster configuration

Adding or removing an IP address using VIP Address Takeover is more complicated than enabling or disabling a port in the Windows NLB configuration. VIP Address Takeover fails when either command on the backup or the primary fails. By comparison, in the Windows NLB configuration there is no dependency on a backup command to succeed for most of the failover scenarios.

Netsh (the Microsoft utility used for IP Address Takeover) takes longer to modify the network configuration than it does to enable or disable a port in the NLB configuration. Also, the time may depend upon a particular NIC and its configuration. Normally, modification takes less than 15 seconds to execute, but in this situation it can take as long as 30-45 seconds.

In addition, the Netsh command can fail if the NIC configuration is already being accessed through the Network Properties User Interface.

Virtual IP Interface Takeover Scripts

Find these scripts on your Installation CD or in the IP package that you downloaded over the internet:

`Ping.vbs`: Ping host and return `1` if the ping is successful, `0` if not, `-1` if target not specified.

`Check_ip.vbs`: Check if the IP address (`arg0`) can be found on the host (`arg1`) and return 1 if found, 0 if not, -1 if address and host not specified.

`INIT_IPTakeOver.bat`: Optional. You can manually execute this batch script, to disable the Virtual IP interface in the RM box before starting the RM process.

`IPTakeOver.bat`: Enable or disable the Virtual IP interface in the RM box, during RM's own internal election process or when a failover event occurs.

For steps that configure the Resource Manager application, go to Procedure: Configuring the Resource Manager HA-Pair.

Resource Manager HA (Linux)

Two options exists to achieve high availability when the Resource Manager is installed on Linux operating systems: Simple Virtual IP failover and Bonding Driver failover.

When you configure either of these two options, each host in the HA-pair has a static private IP address, however, all hosts share one public virtual IP address. The public IP address is used by external Session Initiation Protocol (SIP) endpoints to interact with the Resource Manager on each host in the cluster. If an instance of the Resource Manager fails on any host, the failover is transparent to the caller.

Furthermore, when you use the Bonding Driver failover option, and two or more network cards are installed on the same server, the bonding driver provides active standby functionality for the individual network interfaces.

Important

If a VMWare installation is used for Resource Manager in active backup HA mode, you must use Procedure: Configuring Bonding Driver Failover. Typically two NICs are recommended for bonding (to achieve maximum robustness), however, bonding can also be achieved by using one NIC only.

Precaution for Failover Scenarios
In certain failover scenarios, it is expected that both Resource Manager instances will have an active (and identical) Virtual IP address. In most environments this is not an issue, because the

> network layer ARP table is updated automatically to prioritize traffic to the primary Resource Manager. However, Genesys recommends that you confirm with your Network Operations team that no actions are taken at the network layer when duplicate IP addresses (specifically the Virtual IP address) have been identified.

Use the procedures in this section to configure Resource Manager HA in active standby mode on the Linux host(s).

Limitation

A limitation exists when the Resource Manager is configured in HA mode on Linux, where the NICs, that are associated with the aliases and used for the virtual IPs, are active on both Resource Manager hosts. Depending on the network topology, a situation might arise where a SIP User Agent sends a query to find the active Resource Manager, and instead of sending it to the active instance, sends it to the backup Resource Manager.

To workaround for this issue, configure the primary and secondary Resource Manager instances as described in the section, Creating Alarm Reaction Scripts, Conditions, and Reaction Applications on page 435.

Task Summaries

Task Summary: Configuring the RM in High Availability Active-Standby Mode (Linux)

Objective	Related Procedures and Actions
1. Configure the Resource Manager hosts for HA.	Procedure: Configuring Simple Virtual IP Failover or Procedure: Configuring Bonding Driver Failover.
2. Configure the member IDs for the Resource Manager Applications.	Procedure: Configuring the INIT and NLB Script Files (Linux).

3. Specify the NICs that require monitoring.	Procedure: Specifying the NICs for Monitoring (Linux).
4. Complete finals steps before executing the Resource Manager HA-pair in NLB mode.	Start Resource Manager HA-pair in NLB mode.

Task Summary: Configuring the RM in High Availability Active-Active Mode (Linux)

Objective	Related Procedures and Actions
1. Configure the member IDs in the Resource Manager Applications.	Procedure: Configuring the Resource Manager HA-Pair.
2. Configure the virtual IP in the Media Control Platform, Call Control Platform, and CTI Connector Applications.	Procedure: Integrating Application Objects with Resource Manager and Procedure: Configuring the Call Control Platform. **Note:** When you use these procedures to configure active active HA mode, the virtual IP is used as the Resource Manager IP.
3. Configure the external load balancer.	the vendor documentation for the type of load balancer you are using (for example, F5 or Radware).

Procedure: Configuring Simple Virtual IP Failover

Configure Resource Manager HA in active standby mode by using Simple Virtual IP failover.

Use this failover method if you do not have multiple NICs in the Resource Manager host.

1. Verify that:
    - The Resource Manager hosts conform to the prerequisites for Linux. See Prerequisites.
    - Management Framework is installed, and fully functional. See the Management Framework 8.5 Deployment Guide.
2. At the Linux host, log in as root.
3. Copy the contents of the `/etc/sysconfig/network-scripts/ifcfg-eth0` file to the `ifcfg-eth0:1` file.

4. On the active host, modify the lines in the `ifcfg-eth0:1` file as follows, replacing `<virtual_IP_addr>` with the actual virtual IP address:

```
IPADDR=<virtual_IP_addr>
ONBOOT=no
ONPARENT=no
DEVICE=eth0:1
BOOTPROTO=none
```

For RHEL 5 Releases only:

5. Prepare the `ifup-eth` script:
   Copy `/etc/sysconfig/network-scripts/ifup-eth` file to `<RM_Install_Dir>/bin` directory.
   a. In the `<RM_Install_Dir>/bin/ifup-eth` file, comment out lines 266 to 269, as follows:

   ```
   # if !arping -q -c 2 -w 3 -D -I ${REALDEVICE} ${IPADDR};
   then
   # echo $"Error, some other host already uses address
   ${IPADDR}."
   # exit 1
   # fi
   ```

   b. Enable executable permission, by typing `chmod +x <RM_Install_Dir>/bin/ifup-eth`, and then press **Enter**.
6. Prepare the `ifup` script:
   a. Copy the `/etc/sysconfig/network-scripts/ifup` script to the `<RM_Install_Dir>/bin` directory.
   b. In the `<RM_Install_Dir>/bin/ifup` file, modify lines 145 to 149, as follows:

   ```
   OTHERSCRIPT="<RM_Install_Dir>/bin/ifup-eth"
   # if [!-x ${OTHERSCRIPT}]; then
   # OTHERSCRIPT="/etc/sysconfig/network-scripts/ifup-eth"
   # fi
   ```

   c. Enable executable permission by typing `chmod +x <RM_Install_Dir>/bin/ifup`, and then pressing **Enter**.
   d. Repeat these steps on the standby host.

For RHEL 4 Releases only:

7. Enable executable permission by typing `chmod +x <RM_Install_Dir>/bin/ifup`, and then pressing **Enter**.
8. Repeat Steps 1, 2, 3, and 7 on the standby host.
9. Modify the INIT and NLB script files. See Procedure: Configuring the INIT and NLB Script Files (Linux).

Procedure: Configuring Bonding Driver Failover

Configure Bonding Driver failover on the Resource Manager to achieve High Availability.

1. Verify that:
    - The Resource Manager hosts conform to the prerequisites for Linux. See Prerequisites.
    - Management Framework is installed, and fully functional. See the Framework 8.1 Deployment Guide.
2. At the Linux host, log in as root.

> Tip
>
> Genesys recommends that root user perform the steps in this procedure. However, if a non-root user must use this procedure, see Procedure: Enabling a Non-root User to Configure and Run Resource Manager to make that possible.

3. In the `/etc/modprobe.conf` file, on separate lines add:
   `alias bond0 bonding`
   and
   `options bond0 miimon=1000 mode=1`
4. In the `/etc/sysconfig/network-scripts` directory, copy the contents of the `ifcfg-eth0` file to the `ifcfg-bond0` file.
5. In the `ifcfg-bond0` file:
    a. Change `DEVICE=eth0` to `DEVICE=bond0`.
    b. Remove any line that refers to the hardware address for example, HWADDR=.
6. In the `ifcfg-eth0` file:

   a. Remove any line that refers to the hardware address for example, `HWADDR=`.
   b. Remove any line that refers to the IP address for example, `IPADDR=`.
   c. On a separate line, add `MASTER=bond0`.
   d. On a separate line, add `SLAVE=yes`.
7. Repeat the previous step for the `ifcfg-eth1 file`.
8. Restart the host computer.
9. After the host has restarted, modify the ifcfg-bond0:1 file, as follows, substituting <virtual_IP_addr> for the actual virtual IP address:

```
IPADDR=<virtual_IP_addr>
ONBOOT=no
ONPARENT=no
DEVICE=bond0:1
BOOTPROTO=none
```

For RHEL 5 Releases only:

10. Prepare the `ifup-eth` script:
    - Copy the `/etc/sysconfig/network-scripts/ifup-eth` file to `<RM_Install_Dir>/binINSERT_TEXT` directory.
    - In the `<RM_Install_Dir>/bin/ifup-eth` file, comment out lines 266 to 269, as follows:

```
# if !arping -q -c 2 -w 3 -D -I ${REALDEVICE} ${IPADDR}; then
# echo $"Error, some other host already uses address
${IPADDR}."
# exit 1
# fi
```

11. Enable executable permission, by typing `chmod +x <RM_Install_Dir>/bin/ifup-eth`, and then press **Enter**.
12. Prepare the `ifup` script:
13. Copy the `/etc/sysconfig/network-scripts/ifup` script to the `<RM_Install_Dir>/bin` directory.
14. In the `<RM_Install_Dir>/bin/ifup` file, modify lines 145 to 149, as follows:

```
OTHERSCRIPT="<RM_Install_Dir>/bin/ifup-eth"
# if [!-x ${OTHERSCRIPT}]; then
# OTHERSCRIPT="/etc/sysconfig/network-scripts/ifup-eth"
# fi
```

15. Enable executable permission by typing `chmod +x <RM_Install_Dir>/bin/ifup`, and then pressing **Enter**.
16. Repeat Steps 2 to 11 on the standby host in the HA-pair.

For RHEL 4 Releases only:

17. Enable executable permission by typing `chmod +x <RM_Install_Dir>/bin/ifup`, and then pressing **Enter**.
18. Repeat Steps 2–9 and 13 on the standby host in the HA-pair.
19. Configure the member IDs and NLB script path in the Resource Manager HA Application. See Procedure: Configuring the Resource Manager HA-Pair.
20. Modify the INIT and NLB script files. See Procedure: Configuring the INIT and NLB Script Files (Linux).

Procedure: Enabling a Non-root User to Configure and Run Resource Manager

Create a user with permission to operate Resource Manager without root access. This user will access the Resource Manager install directory, and the directories `/bin/ifup` and `/sbin/ifdown`.

Run the following commands through the root account to give sudo permission for selected commands (`<RM Install directory>/bin/ifup`, `/sbin/ifdown`) to this user.

1. Run the following commands in the console of the RHEL machine:
   a. `visudo`
      This command opens the file.
   b. Beneath the line root ALL= (ALL) ALL, add this line:
      `gvp-user ALL= NOPASSWD: <RM Install directory>/bin/ifup, /sbin/ifdown`
   c. c. Comment the new line Defaults requiretty.
2. Execute the following commands:
   ...to grant permission to gvp-user to edit/use the necessary files and folders.
   - `chown gvp-user /etc/modprobe.conf` ...for RHEL 5
   - `chown -R gvp-user /etc/modprobe.d` ...for RHEL 6
   - `chown -R gvp-user /etc/sysconfig/network-scripts`
   - `chown -R gvp-user /opt/genesys` ...Creates the directory /opt/genesys and grants permission to gvp-user for installing the GVP software.
3. Log on (using ssh) as `gvp-user`.

4. Configure Resource Manager for High Availability with the Procedure: Configuring the Resource Manager HA-Pair. Once you have completed the Resource Manager High Availability configuration...
5. Modify the following files:
   a. **NLB.bat**: add sudo to the beginning of lines 10 and 13...
      - line number 10 should read:
        `sudo ./ifup $activeIntf`
      - line number 13 should read:
        `sudo /sbin/ifdown $activeIntf`

   b. **INIT.sh**: add sudo at the beginning of line 6...
      - line number 6 should read:
        `sudo /sbin/ifdown $activeIntf`
6. The non-root user `gvp-user` should begin the Procedure: Configuring Bonding Driver Failover at step 2.

Procedure: Configuring the INIT and NLB Script Files (Linux)

Ensure that the `mymemberid` parameter that is configured in the cluster section of the Resource Manager Application for `mymemberid=2` is the same as the configuration in the NLB.bat file for `mymemberid=2`.

Configure the INIT and NLB script files so that each Resource Manager host is assigned a unique member ID in the HA-pair.

1. Verify that HA is set up on the Resource Manager hosts. See Procedure: Configuring Simple Virtual IP Failover, or Procedure: Configuring Bonding Driver Failover.
2. At the Linux host, log in as root.

NLB.bat File

3. Follow the instructions in the **NLB.bat** file to update the `mymemberids`:
   - On the Resource Manager host that is assigned `memberID=1` in the <RM_Install_Dir>/bin directory, update the `mymemberid=1`.
   - On the Resource Manager host that is assigned `memberID=2` in the <RM_Install_Dir>/bin directory, update `mymemberid=2`.
4. Save the changes.

INIT.sh File

5. Follow the direction in the `INIT.sh` file to update the following lines:
    - If you are using Simple Virtual IP Failover activeIntf="eth0:1".
    - If you are using Bonding Driver Failover activeIntf="bond0:1".
6. Save the changes.

    Note: In the `NLB.bat` and `INIT.sh` files, the `activeIntf=` parameter must match the `bonding-driver` configuration; for example, use `activeIntf="bond0:1` if you are configuring Bonding Driver failover. Use `activeIntf="eth0:1` if the bonding driver is not configured.

7. Repeat this procedure on the Resource Manager host that is assigned `memberID=2`.
8. Specify the NICs that you want to monitor. See Procedure: Specifying the NICs for Monitoring (Linux).

Procedure: Specifying the NICs for Monitoring (Linux)

Specify the NICs that are to be monitored by the Resource Manager.

1. Verify that:
    - More than two NICs are configured on the same host, and they are fully functional.
    - Two NICs are configured as part of an HA-pair. See Procedure: Configuring Simple Virtual IP Failover or Procedure: Configuring Bonding Driver Failover.
    - a unique member ID to each Resource Manager host in the HA-pair. See Procedure: Configuring the INIT and NLB Script Files (Linux).
2. Log in to Genesys Administrator.
3. On the Provisioning tab, select **Environment > Applications**.
4. Select the Resource Manager Application you want to configure.
5. On the Options tab, scroll to the GVP section. (Create the gvp section if it does not exist.)
6. Add the nic parameter that corresponds `nic.ethX` parameter.
7. If not configured by default, edit the `nic.upvalue` parameter as follows:
    `nic.upvalue = up`
8. If not configured by default, edit the nic.linkattribute parameter as follows:
    `nic.linkattribute = MII Status:`
9. Edit the `nic.eth0` parameter as follows:
    `nic.eth0 = "/proc/net/bonding/bond0"` Configure the `nic.eth0` parameter value as shown above, even when Simple Virtual IP failover is used. If the file cannot be read, the NIC status is queried directly by default during NIC detection.
10. Add the `nic.eth1` parameter as follows:
    `nic.eth1 = "/proc/net/bonding/bond0"` If Simple Virtual IP failover is used,

configure the `nic.eth1` parameter as follows (where represents an empty string):
`nic.eth1 =`

11. If more than two NICs exist, configure the nics option value to `0 1`.

> Tip
>
> The instructions in Step 8, 9, and 10 are based on the assumption that the chosen network interfaces are numbered 0 and 1. If this configuration does not match the actual interface numbers in your system, change the values accordingly.

12. Save the changes.
13. Execute the INIT file on each Resource Manager host. See Start Resource Manager HA-pair in NLB mode.

Creating Alarm Reaction Scripts, Conditions, and Reaction Applications

When an active Resource Manager goes down, but does not stop it's virtual IP, and then the backup Resource Manager becomes active and starts it's virtual IP, the two systems will claim the virtual IP. Therefore, when a system sends an ARP query to determine where the virtual IP can be reached, it might obtain or use the response from the system with the inactive Resource Manager.

To ensure this does not occur, complete each task in the Task Summary: Configuring the Resource Manager to Use Alarm Scripts on both the primary and backup Resource Manager instances.

> Tip
>
> Alternatively, you can use a wizard in Genesys Administrator to complete the first two tasks in the Task Summary table. See Procedure: Using the Create New Application Wizard. As you proceed through the wizard, enter the information in the required fields as outlined in the procedures in this section.

Task Summary: Configuring the Resource Manager to Use Alarm Scripts

Objective	Related Procedures and Actions
1. Create a new Third Party Server template.	Procedure: Creating the Third Party Server Template.
2. Create two Reaction Applications.	Procedure: Creating the Reaction Applications.
3. Create and configure two Alarm Reaction scripts.	Procedure: Creating and Configuring the Alarm Reaction Scripts.
4. Create two Alarm Conditions to send an alarm when either instance of the Resource Manager is stopped.	Procedure: Creating an Alarm Condition for an RM Stopped Intentionally.
5. Create two Alarm Conditions to send an alarm when either instance of the Resource Manager stops unexpectedly.	Procedure: Creating an Alarm Condition for an RM Stopped Unexpectedly.

Procedure: Creating the Third Party Server Template

Create a Third Party Server template to use for the Reaction Applications (which the next procedure will create).

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select **Provisioning > Environment > Application Templates**.
3. Click **New**.
4. On the Configuration tab, populate the following fields:
    - **Name: Third Party Server**
    - **Type: Third Party Server**
    - *Version: 1.0'*
5. Click **Save** & **Close**.
6. Create the Reaction Applications. See Procedure: Creating the Reaction Applications (immediately below).

Procedure: Creating the Reaction Applications

Create the Reaction Applications that stop the NIC (the virtual IP interface) on the Resource Manager that is down (intentionally or unintentionally).

1. In Genesys Administrator, go to `Provisioning > Environment > Applications`.
2. Click **New**.
3. On the Configuration tab, in the General section, populate the following fields:
    - **Name:** `stop_pri_VIP` or `stop_bac_VIP`
    - **Application template:** Third Party Server
4. In the Server Info section, populate the following fields:
    - **Host:** `<Primary RM host object>` or `<Backup RM host object>`
      (Add the name of the primary or backup Resource Manager host object.)
    - **Listening Ports:** `<Port Number>`
      (Add a default unused port, typically in the 70xx range.)
    - **Working Directory:** `<RM bin directory>`
      (Add the actual Resource Manager bin directory.)
    - **Command Line:** `./NLB.bat`
      For IPTakeover, you must use `IPTakeOver.bat`
      `Command Line=IPTakeOver.bat`}}
    - **Command Line Arguments:** disable 2
      (for stop_pri_VIP) or disable 1 (for stop_bac_VIP)
5. Click **Save** & **Close**.
6. Create the Alarm Scripts. See Procedure: Creating and Configuring the Alarm Reaction Scripts.

Procedure: Creating and Configuring the Alarm Reaction Scripts

Configure the Alarm Reaction scripts that cause the Reaction Applications to be run when the Alarm Reaction script is called.

1. In Genesys Administrator, go to **Provisioning > Environment > Scripts**.
2. Click **New**.
3. On the Configuration tab, in the General section, populate the following fields:
    - Name: **`pri_rm_not_running`** or `bac_rm_not_running`
    - Script Type: **`Alarm Reaction`**
4. Click **Save** & **Close**.

Configuring Alarm Scripts

5. In the list of Scripts, click (to highlight) the Alarm script you created in Procedure: Creating and Configuring the Alarm Reaction Scripts.
6. In the Tasks pane, click Script Wizard and enter the following in each step of the wizard:
    - **Tenant and Name:** Select the applicable tenant.
    - **Alarm Reaction Type:** Start a specified application.
    - **Alarm Reaction Details:** stop_pri_VIP or stop_bac_VIP (See Procedure: Creating the Reaction Applications.)
7. When the wizard is complete, Click **Finish**.
8. Create the Alarm Conditions. See Procedure: Creating an Alarm Condition for an RM Stopped Intentionally.

Procedure: Creating an Alarm Condition for an RM Stopped Intentionally

Create an Alarm condition under which an Alarm script is activated when the Resource Manager is stopped intentionally.

1. In Genesys Administrator, go to `Provisioning > Environment > Alarm Conditions`.
2. Click **New**.
3. On the Configuration tab, in the General section, populate the following fields as shown here:
    - **Name:** `rm_pri_stopped` or `rm_bac_stoppedINSERT_TEXT`
    - **Description:** primary RM was manually stopped or backup RM was manually stopped
    - **Detect Clearance Timeout:** `1` (change the default value from 86400)
    - **Detect Log Event ID:** `5091`
    - **Detect Selection Mode:** Select by Application (from drop-down menu)
    - **Detect Application:** Enter primary or backup Resource Manager Application object (the actual Resource Managers, not the reaction objects).
4. In the Scripts section, populate the following fields with equivalent data: colspan= "2"|Scripts

Field	Sample Data
General	
Name	rm_pri_stopped
Description	Primary RM was manually stopped
Category	Major (selected from a drop-down list)
Detect Script	(select from a search field)

Clearance Timeout	1
Detect Log Event ID	5091
Detect Selection mode	Select by Application (from a drop-down list of applications)
Detect Application Type	GVP Resource Manager (selected from a drop-down list)
Detect Application	HA_RM_2 (selected from a search field)
Cancel Log Event ID: 0	
State: Enabled	
Name	pri_rm_not_running
Script Type	Alarm Reaction
State	Enabled

5. Click Add in the Reaction Scripts: field to add `pri_rm_not_running` or `bac_rm_not_running`. (See Procedure: Creating and Configuring the Alarm Reaction Scripts.)
6. Click **Save** & **Close**.
7. Provision the Alarm Conditions. See Procedure: Creating an Alarm Condition for an RM Stopped Intentionally

Procedure: Creating an Alarm Condition for an RM Stopped Unexpectedly

Create an Alarm condition that activates an Alarm script when the Resource Manager is stopped unexpectedly.

1. In Genesys Administrator, go to `Provisioning > Environment > Alarm Conditions`.
2. Click **New**.
3. On the Configuration tab, in the General section, populate the following fields:
    - **Name:** `rm_pri_down` or `rm_bac_down`
    - **Description: primary RM stopped unexpectedly** or "backup RM stopped unexpectedly"
    - **Detect Clearance Timeout:** `1` (change the default value from 86400)
    - **Detect Log Event ID:** `5064`
    - **Detect Selection Mode:** Select by Application (from drop-down menu)
    - **Detect Application:** Enter primary or backup Resource Manage Application object (the actual Resource Manager instances, not the reaction objects).

4. In the Scripts section, populate the following fields:
5. Click Add in the Reaction Scripts: field to add `pri_rm_not_running` or `bac_rm_not_running`. (See Procedure: Creating and Configuring the Alarm Reaction Scripts.)
6. Click **Save** and **Close**.

Reporting Server High Availability

This appendix describes how to configure both the Genesys Voice Platform (GVP) Reporting Server hosts and the Configuration Database to achieve High Availability (HA) on both Windows and Linux operating systems. It contains the following sections:

- Overview
- Reporting Server High Availability Segregated Solution
- Reporting Server High Availability Shared Solution

Overview

Configure the Reporting Server for high availability by using one of two models the Shared Storage Solution or the Segregated Solution:

- **Shared Storage Solution** Both instances of the Reporting Server are connected to a shared storage solution, but only one instance has exclusive access to the ActiveMQ data message store (receive, queue, and, dequeue data from Reporting Clients).
- **Segregated Solution** Each instance of the Reporting Server uses an independent ActiveMQ data message store. However, only the server that is designated as primary activates its message store.

For more information about how the Reporting Server works when it is configured for HA, see High Availability and Scalability.

Additional Considerations

When the Reporting Server is configured to provide High Availability, the following should be considered:

For a Segregated Solution:

- When a Reporting Server fails, the data remaining in its queue is not made available for reporting purposes until the next time that the Reporting Server gains Primary status.

For a Shared Storage Solution:

- When the ActiveMQ message store is clustered, the ActiveMQ broker does not start if another Reporting Server in the cluster is in primary mode (holding an exclusive lock on the message store). The inactive Reporting Server in the cluster is considered to be in backup mode.
- The two-node HA Shared Solution uses Windows Clustering for Windows Server 2003. Before you begin setting up the this HA solution, Genesys recommends that administrators read the Microsoft document Guide to Creating and Configuring a Server Cluster Under Windows Server 2003 specifically, the section that describes how to set up the two-node cluster.
- The tasks that are described in the HA Shared Solution Task Summary are based on the assumption that a Direct-Attached Storage (DAS) disk array is the device that is being used for shared storage; however, a Storage Area Network (SAN) or other similar shared file system can be used.
- Ensure that the two HA servers have identical hardware and that each server has two network interface cards (NIC).
- The Reporting Servers that are used in the cluster must be dedicated servers, and they must not be installed with any of the other GVP components. In addition, the DBMS host that is used by the Reporting Server must be installed on a separate host.

For a Segregated or Shared Solution:

- If the primary server fails and switchover occurs, up to five (5) minutes of VAR summary data might be lost because the VAR data is held in memory and the JVM heap is not currently clustered or replicated. (This does not occur when the switchover is done manually.)
- When you are creating the Reporting Server Applications, use the same template (version) for both objects.
- In the Server Info section of both Reporting Server Applications:
    - Specify the host.
    - In the Listening Port field, enter the default port number (61616).

> **Tip**
>
> The tasks that are described in the HA Segregated Solution and HA Shared Solution Task Summaries begin with the assumption that the Reporting Server database and the prerequisites for both of the

Reporting Servers in the solution are installed. For more information about the Reporting Server prerequisites, see Prerequisites.

Reporting Server High Availability Segregated Solution

Complete the tasks that are required to setup and configure a two-node HA Segregated Solution for the Reporting Server.

Task Summary: Configuring a Reporting Server High Availability Segregated Solution

Objectives	Related Procedures and Actions
Install the first Reporting Server	This server is known as `RS_Server1`. 1. For the VP Reporting Server Parameters, enter the information for the Reporting Server database: ◦ DB Server Host ◦ DB Server Port typically 1433 ◦ Database Name ◦ User Name ◦ Password
Install the second Reporting Server	This server is known as `RS_Server2`. 2. For the VP Reporting Server Parameters, enter the information for the Reporting Server database (use the same database as for `RS_Server1`): ◦ DB Server Host ◦ DB Server Port typically 1433 ◦ Database Name ◦ User Name ◦ Password
Create the connections	3. Connect the Resource Manager, Media Control Platform, and Call Control Platform to the Reporting Server `RS_Server1`. See Procedure: Creating a Connection to a Server.
Configure the backup server	4. In the Server Info section of the `RS_Server1` Application, enter `RS_Server2` in the Backup Server field.

Reporting Server High Availability Shared Solution

Complete the tasks below to set up and configure a two-node HA Shared Solution for the Reporting Server.

Task Summary: Configuring a Reporting Server High Availability Shared Solution

Objectives	Related Procedures and Actions
Complete the preliminary setup	1. Install the DBMS server either Microsoft SQL or Oracle 11g on a host that is separate from the Reporting Server host. See Configuring the Reporting Server Database. 2. Install the Reporting Server on two hosts with Windows Server 2003 Enterprise Edition, Service Pack 2 (or a similar OS with cluster support). See Installing GVP with the Deployment Wizard. Ensure that the drive letter and the path to the installation directory are the same on both servers. 3. On each of the two Reporting Server hosts, designate a shared drive for the Quorum disk and a separate shared drive for the JMS data directory. The Quorum disk is described in the the Microsoft document Guide to Creating and Configuring a Server Cluster Under Windows Server 2003.
Create and install a cluster	4. Use the two Reporting Server hosts to install a two-node Windows Cluster as described in the the Microsoft document Guide to Creating and Configuring a Server Cluster Under Windows Server 2003. 0. Configure the cluster by first creating the shared disk resources (one for the Quorum disk and one for the JMS data directory). a. Assign the shared disk resources to the same Cluster Resource group. b. Take note of the cluster host name. It will be entered as the Host for the first Reporting Server Application for the cluster. See Steps 1 and 2 in the **Configure the hosts in Genesys Administrator** section of this table. c. When you are setting up the cluster, if you receive a message that the Quorum disk cannot be located, refer to Article IDs 888025 and 331801 on the Microsoft support website for more information.

Create a generic cluster application	5. Using the New Resource Wizard in Windows Cluster Administrator, create a generic application for the Reporting Server. 0. Obtain the command-line arguments and the path to the Reporting Server installation directory from the properties of the Reporting Server Application in Genesys Administrator: 0. On the Provisioning tab, select **Environment > Applications**. i. Double-click the Reporting Server Application you want to view. ii. On the Configuration tab, find the command-line arguments and the path to the installation directory in the Server Info section. a. In Cluster Administrator console tree, open the Groups folder. b. In the details pane, click the group that owns the shared disk to be used for the JMS data. c. From the File menu, select New > Resource. d. In the New Resource Wizard, enter the following information: ▪ **Name** Enter `Reporting Server`. ▪ **Description** Enter a description that identifies the Reporting Server. ▪ **Resource type** Select `generic application resource type`. ▪ **Group** Select the group with the shared disks. e. In the next pane of the wizard, add the two Reporting Server cluster nodes as possible owners of the resource. f. In the Available resources list, add the shared disk used as a dependency for the JMS data. g. In the Generic Application Parameters pane, enter the command line including arguments. For example: `C:\<java_bin_path>\java -Xmx512m -jar ems-rs.jar -app "<RS_app_name>" -host <MF_host_name> -port <MF_port>` Issue the command `java -Xmx512m -jar ems-rs.jar` directly, instead of through the

	`rs_startup.bat` script that is used by Genesys Administrator. h. For the Current directory, enter the full path to the Reporting Server installation directory. For example: `C:\Program Files\GCTI\<rs_dir>\`
Configure the hosts in Genesys Administrator	6. Use the cluster host name to create a host in Genesys Administrator. See Procedure: Configuring a Host in Genesys Administrator. 7. Edit the properties of the first Reporting Server Application. 0. In the Host field of the Server Info section, enter the cluster host name. a. Configure the JMS Data directory. On the Options tab, for the `activemq.dataDirectory` option (messaging section), enter the path to the activeMQ data directory on the shared JMS drive. For example: `<JMS_shared_disk_drive>/data/activemq` The second Reporting Server host in the cluster does not require any additional configuration. 8. Use Windows Cluster Administrator to start and stop the Reporting Server hosts in the cluster.

Tip

Genesys recommends that you use the Windows Cluster Administrator (not Genesys Administrator) to start and stop the Applications when the Reporting Server is set up in a cluster.

Optimizing GVP Performance Through HTTP Caching

This topic describes how HTTP caching works in GVP and how to configure caching to improve GVP performance.

HTTP Caching in GVP

HTTP caching is an important aspect of GVP deployment configuration, because it has significant impact on the performance, scalability, and robustness of the deployment. Properly designed caching rules govern properties such as, freshness and cacheability.

These rules enable the majority of HTTP requests to be fulfilled by the cache, while limiting the number of HTTP GET requests to those that are sent when the resource cannot be cached, cannot be found in the cache, or the expired cache entry requires revalidation.

As a result, caching lessens the load on the HTTP server, reduces network traffic, improves fetch response time, and increases fault-tolerance if the HTTP server becomes unavailable.

GVP can perform the caching function itself by using the Fetching Module's in-memory caching or the Squid Caching proxy (See GVP Caching), or it can use an external server a caching appliance, or a web proxy server.

Caching Within the Media and Call Control Platforms

The implementation of HTTP caching on the Media Control Platform and Call Control Platform complies fully with the HTTP 1.1 specification. When the Media Control Platform or Call Control Platform make an HTTP request, the platform handles the request in one of the following ways:

- The requested resource is not found in the cache. The Media Control Platform or Call Control Platform sends an HTTP GET request, and the HTTP server returns an HTTP response. If the response is cacheable, it is added to the cache.
- The requested resource is found in the cache, and the cache meets all the cache freshness requirements. The cache is used to satisfy the request, and an HTTP GET request is not sent.

- The requested resource is found in the cache, but it does not satisfy the freshness requirements. The Media Control Platform or Call Control Platform sends a conditional HTTP GET request, which contains an If-Modified-Since header, to validate the cache. The platform also sends the E-Tag header if the E-Tag header is provided in the cached HTTP response.
- If the resource has not changed, the HTTP server returns an HTTP 304 response code. The HTTP response does not contain a message body, and the Media Control Platform or Call Control Platform uses the cache to satisfy the request.
- If the resource has changed, the HTTP server returns a new HTTP response. The Media Control Platform or Call Control Platform removes the old cache entry and, if the new response is cacheable, adds it to the cache.

Controlling Cache Size

The in-memory cache has a limited size. Therefore, Genesys recommends that you use the following Media Control Platform and Call Control Platform configuration options to control the size of the cache, the size of each cache entry, and the number of cache entries:

- `[fm]cachemaxsize`
- `[fm]cachemaxentrysize`
- `[fm]cachemaxentrycount`

Internally, the platform manages cache entries in an ordered list. A cache entry is moved to the front of the list when it is being accessed, and the cache entry that was used least recently is removed when the cache runs out of space.

Real-Time Cache Clearing

You can clear the internal in-memory cache without restarting. Framework can send a custom message to the MCP or CCP, and the receiving program then clears the cache. Following is the configuration procedure, for the Windows and Linux operating systems.

Procedure: Clearing the Realtime Cache (Windows)

Enable clearing the internal in-memory cache without restarting, using MCP or CCP.

1. Verify that Genesys Management Framework 8.1 or above is installed.
2. Copy the file `mlcmd.exe` into the same directory that holds the `clearfmcache.bat` and `clearfmcache.txt` files.
3. Edit the `clearfmcache.bat` shell scripts in the MCP `/bin` directory (or the CCP `/bin` directory) as follows:

   a. Populate the `CSUSER`, `CSPASSWORD`, and `CSAPPNAME` variables with the same Management Framework credentials that you use to log in to the Genesys Administrator.
   b. Populate the `SCSHOST` variable with the address of name of the machine hosting the Solution Control Server (SCS).
   c. Modify the `SCSPORT` variable, if it is deployed on a non-default port. The configuration is complete.
4. To clear the cache, run the script clearfmcache.bat at the command prompt.

Procedure: Clearing the Realtime Cache (Linux)

Enable clearing the internal in-memory cache without restarting, using MCP or CCP.

1. Verify that Genesys Management Framework 8.1 or above is installed.
2. Go to the Genesys Management Framework - Solution Control Server `/bin` directory.
3. Copy the *mlcmd* command into your to your MCP `/bin` directory (or the CCP `/bin` directory).
   - For 64-bit Linux, that file is named `mlcmd_64`.
   - For 32-bit Linux, that file is named `mlcmd_32`.
4. Edit the `clearfmcache.sh` shell scripts in the MCP `/bin` directory (or the CCP `/bin` directory) as follows:
   a. Populate the `CSUSER`, `CSPASSWORD`, and `CSAPPNAME` variables with the same Management Framework credentials that you use to log in to the Genesys Administrator.
   b. Populate the `SCSHOST` variable with the address of name of the machine hosting the Solution Control Server (SCS).
   c. Modify the `SCSPORT` variable to point to your SCS, if it is deployed on a non-default port.

   The configuration is complete.

5. To clear the cache, run the script `./clearfmcache.sh` at the command prompt.

Cache Control

You can apply cache control settings on either the server side or the client side. In general, it is not necessary to apply the settings on both sides. The decision is usually based on system access policies and maintenance considerations. For example, an application developer who does not have control of the web server settings must apply the settings on the client side.

Server-Side Cache Control

HTTP servers, such as Microsoft Internet Information Service (IIS) and Apache, enable you to apply the following cache control settings to the HTTP resources:

- Expires immediately.
- Expires after a specific amount of time for example, after 5 minutes.
- Expires at a specific date and time. For example, `May 2, 2011 at 9:00 AM`.
- Expires a specific amount of time after the content was last modified. (Not currently supported by IIS.)

Based on these settings, the HTTP Server will add either an Expires header, or a Cache-Control: max-age header to the HTTP response. The Expires header indicates that the response will expire at the specified time, and the Cache-Control: max-age header indicates that the response will expire after the max-age amount of time, which is expressed in seconds.

Tip

If the Media Control Platform or Call Control Platform receives an HTTP response with both the Expires and Cache-Control: `max-age` headers, the platform ignores the Expires header, in accordance with the HTTP 1.1 specification.

Client-Side Cache Control

GVP uses properties or attributes that specify maximum age and staleness (Maxage and Maxstale Attributes) to implement cache control by VoiceXML and CCXML applications.

VoiceXML and CCXML applications support the following cache control properties or attributes:

VoiceXML

- The `audiomaxage/audiomaxstale` properties or the `maxage/maxstale` attributes of `<audio>`.
- The `datamaxage/datamaxstale` properties or the `maxage/maxstale` attributes of `<data>`.

- The `documentmaxage/documentmaxstale` properties or the `maxage/maxstale` attributes of `<choice>`, `<goto>`, `<link>`, `<subdialog>;`, or `<submit>`.
- The `grammarmaxage/grammarmaxstale` properties or the `maxage/maxstale` attributes of `<grammar>`.
- The `scriptmaxage/scriptmaxstale` properties or the `maxage/maxstale` attributes of `<script>`.
- The `maxage/maxstale` SIP REQUEST-URI parameters for the initial VoiceXML page.

CCXML

- The `maxage/maxstale` attributes of `<createccxml>`, `<fetch>`, `<dialogprepare>`, `<dialogstart>`, or `<script>`.
- The `maxage/maxstale` parameters in the HTTP POST request to the `createsession` processor.
- The `maxage/maxstale` SIP REQUEST-URI parameters for the initial CCXML page.

Maxage and Maxstale Attributes

Maxage

- Maxage Indicates that the client does not use cached content that is older than the specified time (in seconds). Setting the maxage attribute to a non-zero value enables you to force the platform to get a fresh copy of a resource before the cached copy expires. A fresh copy can be unconditionally requested by setting maxage value to 0.

Tip

If the client specifies the maxage attribute, and the cache already contains an expiration time that is calculated based on the Expires or Cache-Control: max-age header from the HTTP response, the more restrictive rule (in other words, the rule that results in an earlier expiration time) takes effect.

Maxstale

- `Maxstale` Indicates that the document does not use cached content that exceeds its expiration time by the specified amount of time (in seconds). When the `maxstale` attribute is used, an expired cache that is not too stale (according to the rules of HTTP 1.1) can be used. For example, an author who does not have direct server-side control of the expiration dates of large static files can avoid unnecessary fetching by allowing cached copies to be used after expiration.

The `maxage` or `maxstale` attribute value is first used to calculate the freshness of a cache entry. If the cache entry is not fresh enough, the `maxage` or `maxstale` attribute value is also sent in the Cache-Control header of the HTTP request, so that the HTTP proxy and server can generate the response, based on these settings.

Non-Cacheable Substrings

The Media Control Platform and Call Control Platform support the `[fm]no_cache_url_substring` configuration option, which defines a comma-delimited list of substrings. If an HTTP REQUEST-URI parameter contains any of these substrings, its response is not cached. The default list of non-cacheable substrings is: `cgi-bin, jsp, asp, ?`

Configuration Recommendations

This section provides recommendations for configuring cache control for dynamic and static resources.

Identifying Dynamic and Static HTTP Resources

HTTP resources are categorized as:

- Dynamic Refers to those resources that generate responses, based on the information that is provided in the HTTP requests. Java Server Page (JSP) and Common Gateway Interface (CGI) pages are typical examples of dynamic resources.
- Static Refers to those resources that are predefined. The same content is returned, regardless of the HTTP requests.

Static resources do get updated, but at different frequencies. For example, each one of the following static resources requires a different update frequency:

- An image file that contains a company logo might never change.

- A video file that contains the latest movie trailers might be updated anytime during normal business hours.
- A VoiceXML page that contains a dynamic TTS prompt for example, to play the customer s account balance, might be updated occasionally at any time.

Registration for ECC Variables Static and Dynamic

The CTI Connector (ICM) supports static and dynamic registration of ECC variables.

Static Registration of ECC Variables

When the CTI Configuration parameter [ICMC]eccvariablelist is populated with the desired ECC variables, during start up the CTI Connector sends a REGISTER_VARIABLES message to ICM for registration. The CTI Connector support configuration of ECC variable names along with their tag values (optional) as a comma-separated string.

Dynamic Registration of ECC Variables

If the new ECC variables are received (i.e., apart from those configured ECC variables through [ICMC]eccvariablelist) during the call setup message from the Media Gateway (incoming call), or during the call setup/transfer/end message from the VXML application in MCP, then CTIC will register these ECC variables on the fly, by a sending REGISTER_VARIABLES message.

The ECC variables format is mandatory: `ICMC_ECC_user<variable name>`
...where CTIC submits to ICM only the `user<variable name>` portion as the ECC variable name.

Squid HTTP Proxy

In GVP 8.1.1 and earlier releases, the Squid HTTP Proxy is a mandatory component, because the Fetching Module relies on Squid to support HTTP 1.0-compliant caching.

In GVP 8.1.2 and later releases, Squid is an optional component, because the Media Control Platform and Call Control Platforms are integrated with an in-process Fetching Module library that supports HTTP 1.1-compliant caching.

However, you might still want to take advantage of the features that are provided by Squid, for example:

- The overall size of cacheable resources is too large to be stored entirely in the in-memory cache of the GVP processes. Squid implements disk-based caching, which is more suitable for caching large amounts of data.
- To make use of the sophisticated access control options that Squid provides.

To use Squid or any other HTTP Proxy, use the [fm]http_proxy configuration option to specify the IP address and port number of the proxy.

Cache Control Settings

As described in Cache Control, cache control settings can be applied either on the server side or the client side.

Dynamic Resources

Never cache dynamic resources. Apply cache control settings for dynamic resources in one of the following ways:

- Server side Configure the resource to expire immediately.
- Client side Ensure that the `[fm]no_cache_ url_substring` configuration option covers the URLs that are used by the dynamic resources.

Static Resources That Never Change

Use caching as much as possible to serve static content that never changes, and avoid unnecessary conditional GET requests. Apply cache control settings for static resources that never change in one of the following ways:

- Server side Configure the cache to expire 30 days after access.
- Client side Configure the `maxage` attribute to a large number, such as `2592000` (30 days, expressed in seconds).

Static Resources That Might Change (Visible Immediately)

If the resource update must be immediately visible to the client, enable the response to be stored in the cache, and configure the `maxage` attribute with a value of `0` to ensure that a conditional GET request is sent for validation. In most cases, an HTTP 304 response is returned, which is still more efficient that receiving a `200 O`K response with the full HTTP message-body.

Apply cache control settings for static resources that might change and that must be visible immediately in one of the following ways:

- Server side Configure the cache to expire immediately after access.
- Client side Configure the `maxage` attribute to a value of `0` (zero).

Static Resource That Might Change (Not Visible Immediately)

Determine the maximum acceptable delay before a resource update is detected, to ensure that a conditional GET request is sent if the cache is older than the amount of time that is configured in the MAX_DELAY attribute.

Apply cache control settings for static resources that might change and that are not visible immediately in one of the following ways:

- Server side Configure the cache to expire MAX_DELAY after access.
- Client side Configure the `maxage` attribute to the value of the MAX_DELAY attribute. Alternatively, if the HTTP response already specifies an Expires or `Cache-Control: max-age` header value that cannot be changed on the server, configure the `maxage` attribute with a value of `0` and the `maxstale` attribute with the same value as the MAX_DELAY attribute.

Considerations and Usage Notes

Consider also, the following usage notes:

Synchronizing the Clocks

- To function properly, the HTTP 1.1 caching algorithm requires that the system clocks on the server, proxy, and client be synchronized. If the clocks are not synchronized, the cache entry might expire sooner than expected.

Gathering Statistics

- After the cache control settings are implemented, there are several ways to gather statistics for tuning and monitoring:
    - The Fetch dashboard in Genesys Administrator displays near real-time Media Control Platform and Call Control Platform fetching statistics. For more information about this dashboard, see the GVP 8.5 User's Guide.
    - The fetch_end metrics log from the Media Control Platform and the fetch_resp metrics log from the Call Control Platform contain information

about cache hits and misses, as well as other data. For information about the metrics logs, see the Genesys Voice Platform 8.5 Metrics Reference and the Genesys Voice Platform 8.5 CCXML Reference.