Genesys Media Server 8.5

Deployment Guide

About Genesys

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

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

Genesys and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

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

Customer Care from Genesys

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

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 85gvp_dep-gms_12-2013_v85.001.00

Table of Contents

List of Procedures

Preface

Welcome to the *Genesys Media Server 8.5 Deployment Guide*. This guide introduces you to the concepts, terminology, and procedures that are relevant to Media Server 8.5.

This document is valid only for 8.5 releases of this product.

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

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 181.

About Genesys Media Server

Genesys Media Server 8.5 is a robust, carrier-grade media-processing server that is designed to handle all media interactions within the Genesys Voice Platform Solution (VPS), unifying voice and web technologies to provide a complete solution for customer self-service or assisted service.

As part of the Genesys Voice Platform (GVP) Media Control Platform, Media Server is fully integrated with the Genesys Management Framework. Media Server interacts with other Genesys components and can be deployed in conjunction with other solutions, such as Enterprise Routing Solution (ERS), Network Routing.

To install, configure, tune, activate, and manage Genesys components, including Media Server, you can use Genesys Administrator, the standard Genesys configuration and management graphical user interface (GUI).

Intended Audience

This guide is primarily intended for system administrators who will be installing and operating Media Server 8.5. This guide assumes that you have a basic understanding in the following areas:

- Computer-telephony integration concepts, processes, terminology, and applications
- Network design and operation
- Familiarity with your own network configurations

You should also be familiar with Genesys Framework architecture and functionality.

Making Comments on This Document

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

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

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

Contacting Genesys Customer Care

If you have purchased support directly from Genesys, please contact Genesys Customer Care.

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

Document Change History

This is the first release of the Genesys Voice Platform 8.5 Media Server Deployment Guide. This section list topics that are new or that have changed significantly since the last release of this document.

Release 8.5.0 List of changes since 8.1.7:

- Chapter 3, “Media Server Functions,” on page 39:
  - Added bullet points about mp3 call recording and specification to the section “Summary and Characteristics” on page 75.
  - Added the section “Call Recording Encryption” on page 77.
  - Added a note about redundancy in the Call Recording Solution to the section “Failover Handling” on page 77.
  - Added the section “Policy-based Tenant Recording Profiles” on page 77.

Chapter

1 Introduction

This chapter provides a high-level overview of Genesys Media Server 8.5, its basic architecture within the Voice Platform Solution (VPS), and its features. It includes the following sections:

Overview

Genesys Media Server 8.5 is a unified media server that handles all media interactions, such as network prompts, IVR interactions, conferencing, call-progress detection, and call recording. It generates and processes media streams in Real-time Transport Protocol (RTP) format and interacts with SIP User Agents (UA), passing the results of those interaction to SIP Server.

Media Server is a subset of the Genesys Voice Platform (GVP), containing a minimum set of core components—the Media Control Platform and Resource Manager—that provide media services within a telephony environment. These two components, along with SIP Server, are integral elements within the Genesys Voice Platform Solution (VPS). See "VPS Components" on page 36.

If you currently have GVP deployed in your environment, there is no need to deploy another Media Server `Application` to provide media services, rather you can use GVP to provide all media server functions. Alternatively, if you deploy Media Server, and later want to deploy the entire suite of GVP components, you will not have to re-deploy Media Server, even if you have multiple Media Servers in your environment.

Role of Media Server

Media Server replaces Genesys Stream Manager 7.6, and provides enhanced features and media functionality by using Media Server Markup Language (MSML). When SIP Server receives requests for media services it relies on Media Server to perform the media-processing functions. Media Server acts as a SIP UA, enabling SIP Server to act as an application server by providing media services to users who are on the network. To control the media path, SIP Server remains on the signaling path and responds to interaction results from Media Server and other external network events.

Interaction with SIP Server

Depending on the media-control interface that is used, SIP Server can send media requests either in the initiating SIP `INVITE` messages or in SIP `INFO` messages as part of a midcall request. SIP Server negotiates the Session Description Protocol (SDP) on behalf of the UA and the Real-time Transport Protocol (RTP) media is sent directly between the UA and the VP Media Control Platform. When interacting with Media Server, SIP Server acts as an MSML client, which enables it to send and receive MSML messages.

Call-Control Functions

Media Server does not perform any call-control functions. Call-control functions are performed by the VP Resource Manager, which acts as a proxy server. When you are integrating Media Server with SIP Server and you plan to deploy multiple Media Server instances, the Resource Manager is required to provide load balancing.

Features and Functionality

This section describes the media services, features, and functions that are provided by Media Server.

Media Services

Media Server provides the following services:

- Announcements—Provides a simple announcement service that plays various types of prompts, such as music, and recorded files.
- Simple Prompt and Collect—A simple service that plays prompts and collects DTMF tones input by the caller.
- Call Recording—Includes these sub features:
  - Prompts and records the audio stream from the caller.
  - Records the full call from between the caller, the agent, and (in some cases) the supervisor.
  - Performs dual channel (audio) Call Recording.

- Geo-Location—Supports SIP Server geo-location functionality for active recording, when it is used in multi-site operations. Details of this new functionality include:
  - The MCP passes the value of the header from the initial INVITE of the first call to the first CRQM recorder SIP session's INVITE message.
  - The MCP passes the value of the header from the initial INVITE of the second call to the second CRQM recorder SIP session's INVITE message.
  - The new configuration parameter `[sip] mpc.copyheaders` passes the specified SIP headers from an inbound call's INVITE message to the outbound INVITE message, to a third-party recorder. It is set to `X-genesys-geo-location` by default.
- Conferencing—Allows creation of conferences of up to 32 participants. Media server supports a few specific types of conference, such as supervisor monitoring. These conferences can be initiated in the same manner as they were in Stream Manager. Media Server also supports complex conference structures that use MSML tags.
- Secured Transports—Supports secured transports Secure Real-time Control Protocol (SRTCP), Secure Real-time Transport Protocol (SRTP) and Secure Session Internet Protocol (SIPS). In order for the entire solution to be secured, user agents, application servers (for example, SIP Server) and other third-party speech servers also must support secured transport.
- Encoding and Transcoding—Supports encoding and transcoding a wide range of audio and video codecs, which apply to various features.
- Video Support—Supports video-capable clients joining together in a conference, and allows for video prompts. Similar to audio codec support, video support applies to a subset of features for each feature group.

Selected Features

Media Server provides the following additional features:

- Audio and video transcoding support
- Audio and video codec support
- Detection and handling of DTMF digits (inband, RFC 2833, and SIP `INFO`)
- Flexible packet size and SDP `ptime` support
- Type-of-Service (ToS) tagging for Real-time Transport Protocol (RTP) packets
- Specification of maximum record size
- WAV and AVI container support
- MSML support for GVP VoiceXML applications.
- Configurable DTMF tone generation method, based on remote SDP origin

- DTMF tone forwarding in conference
- Input conference gain, based on the SDP origin
- Separate audio and video source-play combinations
- RTP dejittering

Functions

Media Server performs the following functions:

- Plays back Real-Time Streaming Protocol (RTSP) streaming-media content (such as audio and synchronized audio/video).
- The NGI supports the GVP extension property `com.genesyslab.streamingaudio`, which is the default value of the streaming attribute of <audio>. It takes a value of `true` or `false` (default).
- Pre-fetches content and caches in-memory or stores in local file.
- Performs VoiceXML Play applications.
- Detects and handles DTMF tones, SIP `INFO`, and telephone events.
- Provides audio conferencing by using coordinated video switching.
- Acts as a Recording server.

Supported Topologies

You can deploy Genesys Media Server in various supported topologies, which enables you to choose the one that best suits your environment.

Table 1 describes the deployment of components on one server and on two separate servers.

Deployment Options

Components	Description of functionality
Single-server deployment	
1. Media Server and SIP Server are co-resident	There is no SIP Server redundancy.
2. Media Server and two SIP Server instances are co-resident	To provide redundancy, the two instances of SIP Server are deployed in active/hot-standby mode. (In this case, SIP Server does not use a virtual IP.)
Dual-server deployment	
1. Media Server and SIP Server are installed on separate hosts	There is no SIP Server redundancy.

Deployment Options (Continued)

Components	Description of functionality
2. Media Server and one instance of SIP Server are co-resident on host_a. One instance of SIP Server is installed on host_b.	To provide redundancy, SIP Server is configured in active mode on host_a, and configured in hot-standby mode on host_b. (A virtual IP is used.)
3. Media Server and one SIP Server instance, and the Resource Manager are co-resident on host_a Media Server and SIP Server are co-resident on host_b.	To provide redundancy, SIP Server is configured in active mode on host_a, and configured in hot-standby mode on host_b. (A virtual IP is used.) The Resource Manager handles all SIP requests for media resources and provides load balancing for the Media Server instances. (In this case, recommended.)

You can also deploy the Resource Manager in High Availability (HA) mode. If you plan to do so, it is recommended that you deploy each of the Resource Manager instances on separate servers. For more information about deploying HA Resource Manager, see the *Genesys Voice Platform 8.5 Deployment Guide*.

Figure 1 is a simple depiction of Media Server when it is deployed with SIP Server and a Web Application Server. Media Server uses a Web Application Server to provide audio files for announcements and prompts, for Play applications, and to deliver VoiceXML applications, if support for these applications is enabled on the Media Server. (This is Media Server deployment option.)

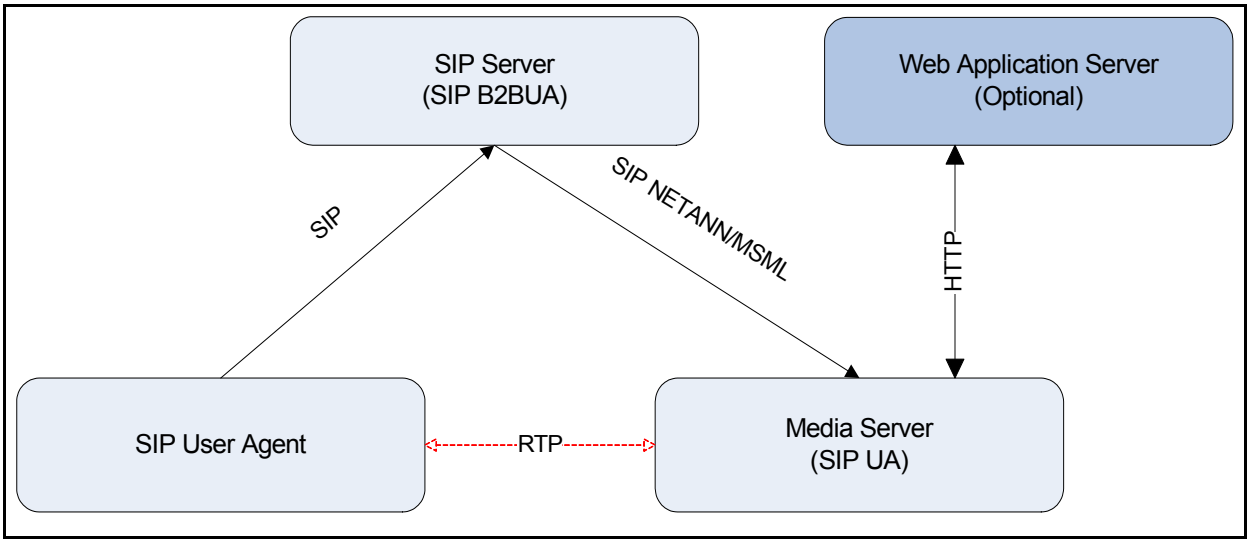

Figure 1: Simple Media Server/SIP Server Integration

Figure 2 on page 18 is an example of multiple Media Server instances when it is deployed with Resource Manager and multiple SIP Server instances.

Resource Manager is recommended in all deployments, but required when you deploy more than one Media Control Platform instance.

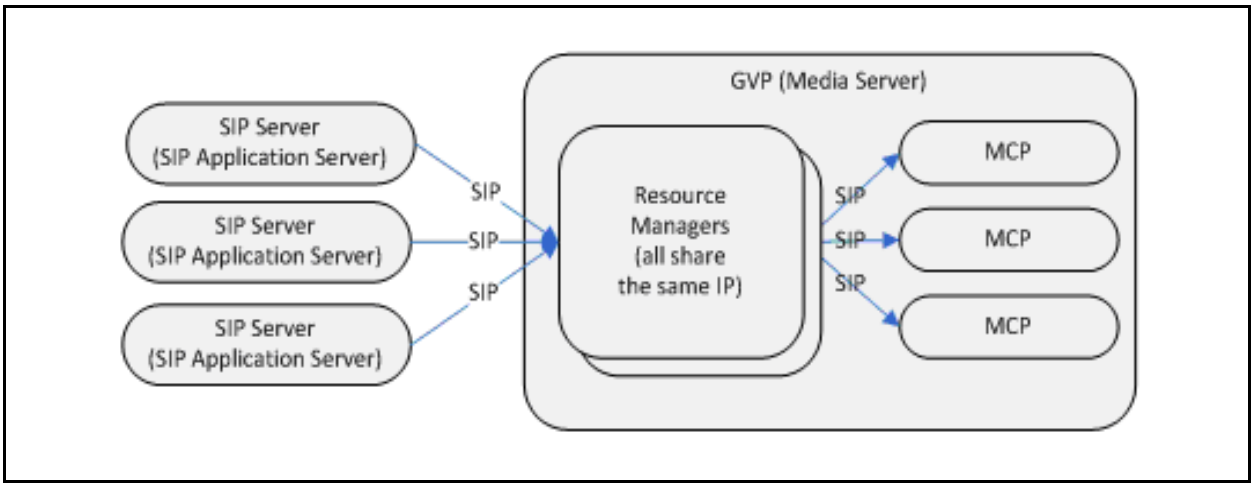

Figure 2: Multiple Media Server and SIP Server Integration with Resource Manager

Port and Virtual IP Management

When you are installing multiple components on a server, you must carefully consider port assignment and, for HA deployments, virtual IP management.

For information about port assignments and the ways in which virtual IPs can be managed, see “Port Assignments” on page 27 and “Virtual IP Management” on page 28.

Scalability

One of the benefits of deploying Resource Manager is the ability to scale the number of Media Servers linearly. Previously, when SIP Server actively managed Stream Manager, each Stream Manager instance was tightly integrated with a particular SIP Server instance which created limitations in terms of scalability. Alternatively, when the Resource Manager manages a group of Media Servers, the Media Server instances are not strictly tied to any one SIP Server instance, which results in improved scalability.

Load Balancing

The Resource Manager can provide load balancing in large-scale environments in which multiple Media Control Platforms (Media Servers) are deployed. By grouping multiple Media Control Platforms in a Logical Resource Group, you can enable the Resource Manager to allocate service requests to the physical resources within the group, while ensuring that the load is evenly distributed among the resources. Except for `conference` services, the Resource Manager selects the physical resource based on the load balancing scheme for the group.

Logical Resource Groups can be configured to use the following load balancing schemes:

- 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 that is specified by the `port-usage-type` parameter.

For more information about how usage is calculated and the other Logical Resource Group parameters, see the *Genesys Voice Platform 8.5 User's Guide*.

For more information about the Resource Manager and how it manages resources for conferences services, see the *Genesys Voice Platform 8.5 Deployment Guide*.

Supported DNs

GVP provides various media services beyond IVR, and depending on the usage, GVP can be configured as various different DN types on SIP Server. Table 1 on page 20 shows the list of DN types and whether recording is supported for each DN type.

Click to see horizontal version

Table 1: DNs Supported by GVP and Media Server

DN Type	Solution applies to:	Protocol	Usage	Can be recorded by SIP Server?	Generates TEvents?	Allows ICON reporting ?	GQM support
VoIP Service	SIP Server, GQM, HPE	SIP/MSML	Media services including music-on-hold, conferencing, call parking (treatments), call recording	No	No	No	No
Voice Treatment Port (VTP)	GVP	SIP	Legacy IVR ports for both inbound and outbound IVR calls	Yes	Yes	Yes	No – due in GQM 8.1.50
Trunk Group DN	GVP, OCS, HPE	SIP/MSML	CPD for outbound solution: ASM mode, transfer mode, and proactive notification Outbound GVP IVR calls (SSG) HPE also uses this DN type for inbound GVP IVR calls	Yes	Yes	Yes	No – due in GQM 8.1.50
Trunk DN	GVP	SIP	Inbound GVP IVR calls	No	No	No	No

Secure Communications

Media Server 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 Media Control Platform (Media Server) and the Resource Manager resources.
- Non-secure SIP—SIP over User Datagram Protocol (UDP), Transport Control Protocol (TCP), or TLS protocol, for call-control and resource-management messaging between the Media Control Platform (Media Server) and the Resource Manager resources.
- Secure Socket Layer (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 (Media Server) and the Media Gateway or endpoint.
- Secure Real-time Control Protocol (SRTCP)—provides the same security-related features as SRTP.

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

> **Note:** Although SIP Server does not support SIPS, the `transport` parameter can be specified in the URL of the SIP request to explicitly specify TLS for security. For example, `sip:dialog@sipua;transport=tls`

Key and Certificate Authentication

Media Server ships with a generic private key and SSL certificate, and default SIP transports for TLS are configured in the Media Server `Application` object. Therefore, basic security is implemented without having to configure it.

For more stringent security, Media Server 8.1.4 and above 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, Media Server supports mutual authentication for clients.

Media Server 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 more information about obtaining SSL keys and certificates, and configuring the Media Server to use SIPS and SRTP in the your deployment, see the section about enabling secure communications in the *Genesys Voice Platform 8.5 User's Guide*.

New in This Release

Genesys Media Server supports the following new features and components:

Release 8.5.0

- Additional features in support of the Genesys Interaction Recording (GIR) Solution:
  - Call Recording Encryption Support.
  - Stereo MP3 encoding for call recording.
  - Submission to S3 storage and webDAV support.
  - Interactions with the GIR.

Chapter

2 Prerequisites and Planning

This chapter provides information that will help you plan your deployment of Genesys Media Server 8.5. It includes the following sections:

Overview

This chapter is written for system administrators, contact center operations heads, and developers who are planning to deploy Genesys Media Server 8.5. The information that you gather for planning purposes will be useful when you install and configure Media Server.

In many ways, the deployment of Media Server is similar to the deployment of other Genesys Framework components, except that the voice signal is carried over the data network. This exception has serious implications for network planning and server sizing. The primary focus of this chapter is to highlight the major planning and resource concerns you face in rolling out Media Server and explain how Media Server overlaps the underlying data network. This chapter is not, however, intended to be an exhaustive guide to network planning.

If you are deploying Media Server in conjunction with a previously deployed Genesys Framework, see the *Framework 8.0 Deployment Guide* for additional information.

This document describes only those areas of deployment planning in which the Media Server differs significantly from other Framework components.

It does not discuss the following topics:

- Configuration Layer
- Management Layer
- Services Layer
- Solution availability
- Security considerations

Prerequisites

Tables 2 and 3 summarize the software requirements for deploying Genesys Media Server on Windows and Linux, respectively.

Note: Before you install any software, review "Host Setup" on page 29 and the Task Summary table at the beginning of Chapter 4 on page 83.

Table 2: Software Requirements—Windows

Category	Requirements and comments
Operating systems	
For Genesys Media Server 8.5 (Mandatory)	• Microsoft Windows Server 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	
Third-party software	• The Windows OS requires a link to OpenSSL v0.9.8o
Management and monitoring tools (Mandatory)	• Genesys Simple Network Management Protocol (SNMP) Master Agent • SNMP Network Management Software (NMS) (optional) Install the Genesys SNMP Master Agent and Media Server on the same host. 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.

Table 2: Software Requirements—Windows (Continued)

Category	Requirements and comments
Specific services and settings (Mandatory)	You must configure certain services and settings on the host before you install Media Server on Windows. For more information, see the *Genesys Voice Platform 8.5 Deployment Guide*.
Web browser (for administration) (Mandatory)	Used only from the administrator's desktop: • Microsoft Internet Explorer (IE) 6.0, SP1, or 7.0

Table 3: Software Requirements—Linux

Category	Requirements and comments
Operating system	
For Genesys Media Server 8.5 (Mandatory)	• 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
Operating system-supporting components	
Third-party software	• The Linux OS requires a link to OpenSSL v0.9.8o
Management and monitoring tools (Mandatory)	• Genesys SNMP Master Agent. • SNMP Network Management Software (optional). Installed the Genesys SNMP Master Agent and Media Server on the same host. Install the Genesys SNMP Master Agent software from the Genesys Management Framework Installation CD. See *Framework 8.0 Deployment Guide*. You can use any type of SNMP NMS—for example, HP OpenView.
Web browser (for administration) (Mandatory)	Used only from the administrator's desktop: • Microsoft Internet Explorer 6.0, SP1, or 7.0

Antivirus Software

Antivirus software can affect system performance and call response time. In an ideal deployment, antivirus software is disabled on Media Server systems. However, Genesys understands the need to have antivirus protection on servers and therefore recommends, at a minimum, that you exclude the Media Server directory from virus scanning, and that you schedule system scans to occur at times when traffic is low.

Also, be aware that antivirus software can interfere with the installation of Media Server during initial deployment. Make sure that the server is not running antivirus software, or any other third-party software, during installation.

Planning Your Deployment

This section contains information that you should be aware of when you are deploying Media Server. It contains the following sections:

Deployment Options

You can use one of two deployment options to integrate Media Server and SIP Server:

- **Single host**—All components (SIP Server, and Media Server) are installed on a single host. SIP Server communicates directly with Media Server. For small-scale enterprise environments, a single host can easily handle all requests for play treatments, conferences, and recordings; however, depending on computer capacity, there might be limitations on the number of available ports. Resource Manager is optional in this mode if only one instance of Media Server is installed.
- **Distributed**—Components are installed on separate hosts (SIP Server, Resource Manager, and multiple Media Servers). In this mode, the Resource Manager is recommended for load balancing. SIP Server communicates directly with the Resource Manager, which manages all Media Servers and is the single point of contact for all of the media resources. This means that Resource Manager can also be a single point of failure, therefore, Genesys recommends that you configure the Resource Manager for High Availability (HA).

To provide further redundancy, you can configure both SIP Server and Resource Manager in HA mode, where the primary and backup servers are installed on separate hosts and are configured in either hot-standby or warm-standby mode.

For information about the supported topologies for Media Server/SIP Server integration, see "Supported Topologies" on page 16.

For more information about how to configure the Resource Manager in HA mode, see the *Genesys Voice Platform 8.5 Deployment Guide*. For more information about how to configure the SIP Server in HA mode, see the *Voice Platform Solution 8.1 Integration Guide*.

Port Assignments

When you are installing multiple components on a server, you must carefully consider port assignments and the possibility of conflicts. Currently, the defaults port ranges for each component are as follows:

SIP Server	5060
Media Server	5070-5071
Resource Manager (not co-resident with SIP Server)	5060-5067
Resource Manager (co-resident with SIP Server)	5160-5167

Table 4 lists the configuration options that are used to assign ports for each `Application`.

Table 4: Configuration Options for Port Assignment

Application	Configuration parameters
SIP Server	`[TServer].sip-port`

Media Control Platform (Media Server)	`[sip].localport`
	`[sip].localsecureport`
	`[sip].transport.0`
	`[sip].transport.1`
	`[sip].transport.2`

Table 4: Configuration Options for Port Assignment (Continued)

Application	Configuration parameters
Resource Manager	[proxy].sip.localport [proxy].sip.localsecureport [proxy].sip.transport.0 [proxy].sip.transport.1 [proxy].sip.transport.2
	[register].sip.localport [register].sip.localsecureport [register].sip.transport.0 [register].sip.transport.1 [register].sip.transport.2
	[subscription].sip.localport [subscription].sip.localsecureport [subscription].transport.0 [subscription].transport.1 [subscription].transport.2
	[monitor].sip.localport [monitor].sip.localsecureport [monitor].transport.0 [monitor].transport.1 [monitor].transport.2

For more information about the configuration parameters for port assignment, see the *Genesys Voice Platform 8.5 User's Guide.*

Virtual IP Management

You can use one of two approaches to provide virtual IP management in your environment, as follows:

- Microsoft Network Load Balancing (NLB) with Cluster Manager—This utility can be enabled and configured on Windows platforms.
- Genesys Scripts—Use Genesys scripts to provide network load balancing to manage the failover activity. These scripts can be loaded and enabled on Windows and Linux platforms.

You can find information about how to configure the Resource Manager and SIP Server for High Availability (HA) in the *Genesys Voice Platform 8.5 Deployment Guide* and the *Framework 8.0 SIP Server Deployment Guide* respectively. In addition, various white papers on the topic of deploying SIP Server in HA mode can be found on the Genesys Customer Care website.

For more information about Microsoft NLB, see the vendor website.

Media and Signaling Channels

The SIP interface of the Media Control Platform/Media Server can be configured using the `[sip].transport.<n>` parameter, which includes an IP address that enables the Media Control Platform to determine which network interface to use for SIP traffic. For a list of parameters that are used to configure ports, see Table 4 on page 27.

Similarly, the Real-time Transport Protocol (RTP) media interface can be configured using the `[mpc].rtp.localaddr` parameter, which uses the IP address to determine which network interface to use for RTP traffic.

- If you are installing Media Server on platforms that have multiple Network Interface Cards (NIC), ensure that you configure the correct IP address for each NIC through which you want to send/receive SIP and RTP traffic.
- If you wish to send an IP address in SDP that is different from the IP address that will be used in `[mpc] rtp.localaddr`, use the option `[mpc] sdp.connection`.

  `[mpc] sdp.connection` is independent of `[mpc] rtp.localaddr`, which sets the IP address on the platform to be used; it specifies the connection value of outgoing SDP content for a call. To disable `[mpc] sdp.connection`, set it to an empty string.

Network Traffic Partitioning

Media Server supports partitioning of network traffic across various network interfaces, including SIP, HTTP, MRCP, RTSP, and RTP. The Media Control Platform's Fetching Module configuration option `[fm].interface` can be used to bind HTTP traffic to a specific IP.

For a complete description of the SIP communications, routing, and RTP configuration parameters, see the *Genesys Voice Platform 8.5 User's Guide.*

Host Setup

Media Server provides some flexibility in combining various components and applications on the same host; however, if you are installing Genesys Administrator and (a single instance of) Media Server on the same host, you must use 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 Media Server.

Caching

Caching is an important aspect of maintaining the overall efficiency of your Media Server deployment. The information in this section will help you choose the caching mechanism that best suits your deployment.

Fetching Module and Squid

The Media Control Platform uses the Fetching Module to fetch documents and perform caching. The Fetching Module, which is integrated with the Media Control Platform, maintains a high-performance in-memory cache and interfaces with the on board Squid Caching Proxy.

Note: Squid Caching Proxy is a separate IP on the GVP CD and is optional.

Unlike visual browsers, there are no end-user controls in the Media Control Platforms Next Generation Interpreter (NGI) context to enable stale content to be updated or refreshed. Instead, the recorded files themselves enforce 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.

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

- HTTP/1.1-Compliant Caching
- Caching Audio and Video Files
- Squid Configuration File on page 31

HTTP/1.1-Compliant Caching

The 8.5.x Fetching Module is HTTP/1.1-compliant and the Squid Caching Proxy is optional to provide more flexibility in the deployment. For example, when Squid is deployed with a web server, multiple Media Control Platform instances can share the same Squid proxy to optimize caching.

The caching policies of the NGI context adhere to the cache-correctness rules of HTTP/1.1. In particular, the `Expires` and `Cache-Control` headers are honored. For more information about the caching policies and caching behavior, see the *Genesys Voice Platform 8.5 Deployment Guide*.

Caching Audio and Video Files

Audio and video recordings are commonly used in Media Server deployments, and they can be very large. Because their content is also mostly static, using cached content significantly improves performance. The Media Control Platform 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 8.5.x Fetching Module itself performs in-memory caching, which is 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.

> **Note:** 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.

Squid Configuration File

The Squid configuration file (`C:\squid\etc\squid.conf` [Windows], or `<Directory>/etc/squid/squid.conf` [Linux]) controls the configuration of the caching proxy. 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 `http://squid.visolve.com/squid24s1/contents.htm`.

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 Media Server application problems. For more information about Squid log files, see the caching reference information appendix in the *Genesys Voice Platform 8.5 User's Guide*.

For information about how to install Squid, schedule log rotations, and manage the cache, see the *Genesys Voice Platform 8.5 Deployment Guide*.

Network Considerations

Media Server performance is directly linked to the performance of the underlying data network. It is essential that you perform a proper network audit to ensure that the data network is properly sized and tuned for real-time (voice) packet transport.

This section describes the factors that affect overall performance of an IP-based configuration, and provides some general rules to follow when deploying Media Server. It contains the following sections:

Voice Quality

The following factors influence voice quality:

- Network latency—Overall network delay.
- Packet loss—Voice packets that are dropped for various reasons (for example, the physical media error or time-outs due to network congestion).
- Packet jitter—Variation in voice-packet arrival times. For example, on a system in which packets are emitted at 20-millisecond (ms) intervals, some packets actually arrive at intervals ranging from 0 to 32 ms.

To minimize network latency and ensure acceptable voice quality, tune the network to prioritize real-time voice packets. There are various available schemes for prioritizing voice packets, depending on the IP router vendor.

Factors that influence voice quality include:

- Packet loss—A result of several factors, including network bandwidth.
- Packet jitter—Minimize this by using a jitter buffer at the endpoint device. Set the buffer size to the maximum anticipated deviation from the typical interpacket emission time.

- Packet misordering—Packets can arrive in the wrong order, which is similar to packet loss.
- Silence suppression—This can save bandwidth but might impact voice quality.
- Codec selection—Codecs that do not compress the audio signal produce better voice quality but use greater bandwidth.

Bandwidth Requirements

To achieve optimal performance and voice quality in your environment, it is critical that you determine the bandwidth requirements for the underlying data network. For example, the bandwidth requirements for a video connection are much higher than for voice connections. Genesys recommends that you conduct performance testing and measurements in a lab environment prior to production rollout.

The following factors affect bandwidth requirements in an IP/Ethernet network:

- Codec selection
- Protocol headers

Testing results in Genesys labs have established that a full duplex voice conversation using the G.711 codec requires approximately 64 kilobits per second (Kbps) of bandwidth. When you are estimating the bandwidth required for your network, consider factors such as network efficiency and utilization.

Remote-Agent Topology

Remote-agent capabilities range from a single remote agent, to a group of remote agents in a branch-office environment. The distributed nature of branch-office or remote-agent architectures adds to the complexity of network sizing and tuning.

Bandwidth and Network Tuning

For remote agents in a branch-office environment, allocate proper bandwidth for voice communication and tune the underlying network for real-time media, just as you would in local network deployments of a VoIP-based system. Remote agents that are using dial-up connections require greater bandwidth for the extra network overhead. The amount of bandwidth required depends on the codec that is selected; however, 56 Kbps is recommended. (Some dial-up connections might accommodate G.729.)

Choose the method of remote access wisely and avoid sending voice over an data network that is not managed, such as the public Internet where voice quality is not guaranteed. For example, a Digital Subscriber Line (DSL) connection is a better alternative than a dial-up connection.

For branch offices, network bandwidth requirements depend on the number of agents. Wide Area Network (WAN) connectivity to the corporate Local Area Network (LAN) must be tuned for real-time voice communications and end-to-end network latency should not exceed 250 ms. Ensure that the service-level agreement from your Virtual Private Network (VPN) provider includes the details about these requirements.

Network Locations

Media Server performance and the network configurations required to optimize it depends on two factors: the scenario that is implemented and the codecs that are used. This section provides some recommendations for both low- and high-processing scenarios.

Low-Processing Scenarios

Low-processing scenarios are those in which the media stream encoding and decoding is either not required or done by using a trivial codec.

Types of low-processing scenarios include:

- Announcement and IVR Service without transcoding (including music-on-hold and Recording Announcement)
- Conference Service with the G.711 codec

In these scenarios, the performance of Media Server is limited by the speed at which the RTP packets are transmitted. The speed of transmission can be faster or slower, depending on the operating system (OS) kernel, or network driver that is being used, the Media Server itself, and even overall network traffic.

Media Server performance can vary, affecting both transmission speed and network traffic, depending on the codec that is selected and the packet size it uses. Typically, Media Server can handle about 320 simultaneous media streams when a codec with a 20 ms packet size is used. Increasing the packet size by choosing another codec can improve performance, but can also have a negative affect on voice quality for endpoints that have small jitter buffers.

High-Processing Scenarios

High-processing scenarios are those in which the decoding and encoding of the media stream is done by using a nontrivial codec.

Types of high-processing scenarios include:

- All types of Conference services using codecs other than G.711 (regular conference, Silent Voice Monitoring, Whisper Coaching, and for Manual Call Recording, implicit conference).
- Announcement and IVR Service with transcoding and Call Recording in *mixed* mode.

In these scenarios, the performance of Media Server is limited by the amount of CPU power available.

A complete range of metrics that relate to the number of participants that the Media Server can handle on a typically configured computer per codec can be found in the *Genesys Hardware Sizing Guide*.

Load Control on Media Server

Media Server provides load-sharing functionality to decrease the number of rejected scenarios that are not processed. In a distributed environment where more than one Media Server exists, a dialog that is rejected solely because the primary Media Server is overloaded, is retried on another Media Server.

In addition, you can use the load balancing, which is done through the Resource Manager to enable the Media Servers in your network to reject new dialogs that exceed the configured threshold. Load balancing optimizes Media Server bandwidth usage.

Traffic Generated by Media Server

Table 5 provides basic data on link traffic between Media Server and Framework components. This data can help you to determine the optimal location for components in the network. The terms *gateway* and *desktop client application* refer to third-party SIP- or H.323-compliant components.

Table 5: Traffic Generated by Media Server

Primary data types	Average message length	Number of messages per transaction	Elements determining total message	Total traffic volume	Timeliness of message delivery
Media Server—Desktop client application, Media Server—Gateway					
Real-time media (voice packets)	24–252 bytes, depending on codec	Variable	Voice packets	Very high	Critical

Table 6 lists some examples of the estimated per-channel values for Media Server traffic that consists of 10 full-duplex conversations. The actual values will vary, depending on the configuration of Media Server.

Table 6: Media Server Traffic Volume Per Channel

Codec	Volume per channel (both directions)
G.711	20 Kbps, 100 packets/sec
G.729	2.5 Kbps, 100 packets/sec
G.729d	5.6 Kbps, 100 packets/sec
G.729e	7.0 Kbps, 100 packets/sec
G.726-16	8.0 Kbps, 100 packets/sec
G.726-24	10.0 Kbps, 100 packets/sec
G.726-32	12.0 Kbps, 100 packets/sec

For more information about Media Server performance and capacity metrics, see the *Genesys Hardware Sizing Guide*.

VPS Components

The following is a list of the mandatory and optional components, including the Media Control Platform (Media Server), that make up the Voice Platform Solution (VPS).

- A centralized instance of Genesys Management Framework that includes the following components:
  - Configuration Database
  - Log DB Server:
    Microsoft SQL Server or Oracle 10g Database Server
  - Configuration Server
  - Genesys Administrator
  - Solution Control Server
  - Solution Control Interface
  - Message Server
  - Local Control Agent—Required on all GVP 8.5 hosts
  - Optional: Genesys SNMP Master Agent on all GVP 8.5 hosts
- Session Initiation Protocol (SIP) Server 8.0.3
- IVR Server
- Stat Server
- Universal Routing Server
- T-Server (switch-specific)
- Voice Platform (VP) Resource Manager:

  - Mandatory component in VPS—One per deployment
  - Mandatory for Media Server/SIP Server integration
  - Can be deployed as an active/standby pair for high availability
  - Prerequisite: Local Control Agent
- VP Policy Server
  - Optional component—many per deployment
  - Prerequisite: Local Control Agent
  - Optional: SNMP Master Agent
- VP Media Control Platform (Media Server):
  - Mandatory component—One or more 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
- VP Reporting Server:
  - Optional component—One per deployment
  - Prerequisite: Local Control Agent
  - Prerequisite: Database Server (Microsoft SQL Server 2005, 2008 or Oracle 10g, 11 g)
  - Prerequisite: Sun JRE 6, Update 5
  - Optional: SNMP Master Agent

Option to Deploy Without VP Reporting Server

Although Genesys recommends that you deploy at least one VP Reporting Server per deployment, large-scale customers might want to use an alternate or existing third-party reporting product. VP Reporting data is available to third-party reporting products and can be viewed on the `Monitoring` tab in the Genesys Administrator interface.

Genesys Voice Platform

Genesys Voice Platform includes the following components:

- Resource Manager
- Media Control Platform
- Call Control Platform
- Reporting Server
- Squid Caching Proxy

- CTI Connector
- PSTN Connector

Media Server

At a minimum, the following components are required to deploy Media Server:

- SIP Server
- Media Control Platform (Media Server)
- Resource Manager

The Resource Manager is an mandatory component if you intend to integrate with SIP Server and/or deploy multiple instances of the Media Control Platform (Media Server).

Also, the Genesys Management Framework core components that are included in the list of VPS components are assumed to be part of your environment.

Reporting Server now ships on the Media Server CD. The installation instructions do not change, and are included in the location of the Installation Package (IP). Look for the installation instructions in the GVP Deployment Guide.

Chapter

3 Media Server Functions

This chapter describes Genesys Media Server supported functionality, and describes how recording servers and clients are managed by the Resource Manager. It includes the following sections:

Media Interfaces

This section describes the media interfaces that are supported by Genesys Media Server 8.5.

NETANN Interface

Media Server provides the following functions:

- Recording Announcement Service support (combining announcement and recording)
- Basic Conference Call support (including Active Speaker Detection, Silent Voice Monitoring, Voice Recording, and Whisper Coaching)

Recording Announcement Service

Media Server provides a nonstandard extension that combines announcement and recording services to record an incoming stream. Use the following format for the URL that is provided in the SIP `INVITE` request:

`sip:annc@MS-hostport;record=record-URL;play= prompt-URL`

In this case, the optional `play` parameter can refer to the media file with a pre-recorded *beep* sound or any other announcement. The `record-URL` parameter is used for generating the output file name, as follows:

- If the filename portion of the URL includes the correct file extension—either `.wav` (default) or `.au`, depending on the configuration of the `[netann].annc.defaultaudioext` or `[msml].defaultaudioext` option—the Media Server uses it for the recording with no modification. If a file with this name already exists, it is overwritten with the new file.
- If the filename portion of the URL does *not* include the correct file extension, the Media Server adds a unique number and file extension, for example, `<name>-<unique-num-string>.ext`.

Files are always recorded using the same codec as those used for transmission.

Optional Media Server Invocation Parameters

The following optional parameters are supported:

- `content-type`—Overrides the MIME type when the 'play' parameter specifies a file or an http URI.
- `repeat`—Specifies the number of times the prompt will be played.
- `delay`—Specifies the delay interval between announcement repetitions.
- `duration`—Specifies the maximum duration of an announcement.
- `digit-duration`—Specifies the duration of DTMF digits.
- `mode`—Set to `stream` to enable HTTP streaming for HTTP fetched media or omit to disable HTTP streaming.

Notes: When `mode` is set to `stream`:

- Availability of the prompt file is determined by checking whether all HTTP headers have been received. The remainder of the data can continue to arrive after the check.
- The fetch timeout is interpreted as the maximum time to receive all HTTP headers.
- HTTP streaming is enabled; content is played as it is being fetched.
- Only audio streaming is supported.

Additional HTTP Streaming Support

Partial fetch output is supported for HTTP media content. This means that output can begin as soon as sufficient data is available, even if the entire file has not yet been retrieved.

Sharing the HTTP stream among multiple sessions playing the same HTTP URL is supported. A shared stream means that a single HTTP connection with data being received will be reused by multiple requests to fetch the HTTP data. This is designed to be used with HTTP servers that send live streamed data via HTTP, and also provide the latest media, rather than restarting the stream when a new HTTP fetch arrives.

The HTTP stream is determined to be sharable based on whether the HTTP URL address (the <*address*> part of `http[s]://<address>[:<port>][path]`) is one of the addresses configured to be a live HTTP stream server.

- Session playing the shared HTTP stream begins, starting with the mostly recently received media.
- Shared HTTP streaming is not supported for 3GP, AVI and MP3 file contents.

VCR control is supported, but with the following limitations:

- VCR skip-back may skip back less than the requested amount, if the content to skip back has been erased from memory.
- HTTP shared streaming has some VCR skip-back limitations. The previous prompt is re-played with an initial pause equal to how long the prompt was played previously before being skipped, because the offset `0` of the previous prompt is considered to be the latest content being fetched, and starting from offset *n* seconds will require waiting for *n* seconds for the live stream.
- If VCR skip-ahead maps to the content still being fetched, play pauses until the corresponding content is fetched.
- Fetched content is never cached. Because of this, HTTP streaming will not perform as well as full HTTP fetching, and should be used with live HTTP streaming content only.
- Playing HTTP stream from Mayah Centauri Server is supported.
- Non-shared HTTP streaming is supported.
- If the HTTP stream cannot be shared, it uses non-shared HTTP streaming.
- Sessions playing the non-shared HTTP stream always start play from the beginning.

Conferencing and Call Supervision

For basic conference calls (including Silent Voice Monitoring, Voice Recording, and Whisper Coaching), Media Server uses a protocol that is compatible with current conference call standards. To establish a conference call, the `INVITE` message is sent to Media Server for each participant as a URL—for example:

`sip:conf=UniqueID` @`MS-hostport`;confrole=`conf-Role`;`URI-parameter`...

Where:

- `UniqueID`—Any string that uniquely identifies the conference call. The first `INVITE` message with the previously unknown ID creates the conference call and all subsequent `INVITE` messages with the same ID adds the participants to the conference call.
- `MS-hostport`—The Media Server location (as required by RFC 3261).
- `conf-Role`—A nonstandard extension that specifies coach or student roles for Whisper Coaching support.
- `URI-parameter`—The SIP `Request-URI` parameter, as described in RFC 3261.

Media Server allows the following additional values for the `confrole` attribute:

`confrole=regular`	Customer call leg receives audio from the mixer (or student call leg in a Whisper Coaching scenario), and video from the agent/student call leg (or from the file in a Push Video scenario).
`confrole=agent` or `confrole=student`	Agent call leg receives audio from the mixer, and video from the regular or customer call leg (or from the file in a Push Video scenario).
`confrole=coach`	Supervisor call leg in a Whisper Coaching scenario receives audio from the mixer, and the same video stream as the agent/student leg.
`confrole=monitor`	Supervisor or recording device call leg in a Silent Call Monitoring scenario receives audio from the mixer, and the same video stream as the agent/student leg.
`confrole=push`	Media playback device call leg does not receive any media; incoming audio stream is pushed to the mixer, and video stream is pushed to a regular or customer call leg.
`confrole=push-all`	Media playback device call leg provides audio to the mixer, and video stream to all call legs in the conference call.

Selection Method for Video Conferencing

MSML conferencing requests support the element selector, in which the attribute method can be set to `vas`, `fixed`, and `confrole`. If the attribute method is set to `vas`, the loudest participant in the conference is selected.

For NETANN conferencing, the [conference].`video_output_algorithm` configuration option value can be set to `confrole` (default), `fixed`, `loudest`, or `none` (disables video). If the value is set to `loudest`, the loudest participant in the conference is selected.

Active Speaker Notification

In addition, the `<asn>` element is supported as a child of `<audiomix>`. The `<asn>` element does not respect any of the attributes associated with it. Instead, it requests notification of the current loudest speaker from the conference, based on the configuration of the `[conference] active_speaker_update_time` option.

If the <asn> element is present, active speaker notification requests are sent to the conference creator at a configurable interval, by using an `msml.conf.asn` event. If the `<asn>` element is not present, the active speaker notification is not sent.

Silent Voice Monitoring

Calls that use Silent Voice Monitoring are established through Media Server as a regular conference call but with the monitoring call leg muted. The SDP for the muted (monitoring) leg must include a `=recvonly` attribute for this audio stream to indicate that this endpoint will not send Real-time Transport Protocol (RTP) packets. If it does, Media Server ignores these packets.

Barge-in functionality is supported for audio and video calls.

Whisper Coaching

Whisper Coaching functionality is controlled by a nonstandard attribute `confrole` in the URL in the conference call leg. In order to establish a Whisper Coaching session between a customer, an agent, and a supervisor, the following `INVITE` messages are sent with these URLs including the same `ConfID`:

Customer:

```
sip:conf=ConfID @MS-hostport [; URI-parameter]...
```

Agent:

```
sip:conf=ConfID @MS-hostport; confrole = student [; URI-parameter]...
```

Supervisor:

```
sip:conf=ConfID @MS-hostport; confrole = coach [; URI-parameter]...
```

Media Server mixes the voice streams so that the agent and the supervisor can hear all call parties, but the customer hears only the agent. The conference call must include only one customer and one agent. However, there are no limitations on the number of supervisor call legs.

Integration with SIP Server

SIP Server can forward NETANN requests for media services from the network to the GVP. To configure this functionality, create a Trunk DN for each type of GVP media service. For example, for NETANN announcements, configure the Trunk DN as follows:

1. Set the prefix option to `annc`. This matches the user part of the `Request-URI` in the network `INVITE`:

   ```
   INVITE sip:annc@172.24.129.75:5060;play=greetings.wav SIP/2.0
   ```

2. Set the `sip-proxy-uri-parameters` configuration option value to `true`. SIP Server matches the prefix to this Trunk, copying the URI from the network `INVITE` to the outgoing `INVITE`. It sends to this GVP Trunk DN.

MSML Interface

Media Server provides advanced control of media services according to IETF draft-saleem-msml-07.

The following topics describe how the Media Servers MSML interface enables the control of these services:

- Transport
- URS-Centric Applications on page 45
- Outbound Call Treatments on page 45
- Conferencing on page 47

Transport

Media Server transports Media Server Markup Language (MSML) content in the body of SIP `INFO` requests within the SIP dialog, which is fully supported by SIP Server and the Resource Manager.

Identifiers

There are three classes of objects defined in MSML:

- Connection—Always generated by the Media Server (see "Connection Object Identifier").
- Conference—Generated by the MSML client in the `<createconference>` tag with the `Name` attribute. If the identifier is not named in MSML, Media Server automatically assigns one and returns the identifier in the MSML response.
- Dialog—Generated by the MSML client in the `<dialogstart>` and `<dialogprepare>` tags with the `Name` attribute. If the identifier is not named in MSML, Media Server automatically assigns one and returns the identifier in the MSML response.

Connection Object Identifier

Each call leg that comes in to Media Server is considered a connection object in MSML. The Media Server automatically creates a connection object for the call leg when it accepts the call. Media Server assigns an identifier to the connection object which is the local tag in the SIP dialog. This is the tag parameter in the `To` header that is returned in the `200 OK` response to the initial SIP `INVITE` message. SIP Server uses this tag to identify the connection object in the MSML. Media Server generates the identifier, which guarantees the local uniqueness of the connection identifier.

URS-Centric Applications

This section describes the ways in which Media Server implements MSML for call treatments requests from URS-centric applications.

Announcement Treatments

Media Server 8.1 and above replaces Stream Manager 7.6 for all announcement treatment capabilities, with additional support for multiple prompts in a single SIP `INVITE` request.

Multiple Prompts

When NETANN is used to play announcements with multiple prompts, each prompt requires a separate SIP dialog to the Media Server, which means a new `INVITE request` is sent for each prompt in the treatment. This creates a barrage of `INVITE` requests to the Media Server and SIP client. When MSML is used, multiple prompts are played over a single SIP dialog and SIP Server need only send a single `INVITE` message.

Multiple Prompts and Collect Digits

Media Server relays all incoming Dual Tone Multi-Frequency (DTMF) digits that are received on the RTP stream to the SIP Server as SIP `INFO` messages. Media Server uses MSML to define a prompt list, collecting all of the DTMF digits within the treatment and delivering them in a single SIP `INFO` message.

Other Supported Treatments

SIP Server and Media Server support the following additional treatments:

- Play Application
- Music on Hold and Music on Queue
- Record User Announcement
- Busy and Fast Busy
- Silence
- Ringback

Outbound Call Treatments

This section describes how Media Server and SIP Server perform the media-related functions that are required for outbound call treatments.

ASM Mode

The Media Server and SIP Server function in the following way in an Active Switching Matrix (ASM) to process media for an outbound call:

1. SIP Server pins the agent on Media Server, and plays music to the agent. (The `RequestMakeCall` request provides an extension to allow the application to define a music-on-hold treatment to the destination.)
2. SIP Server makes and outbound call.
3. When the customer leg is connected, SIP Server orders Media Server to play a *beep* to alert the agent.

4. The media between the customer and the agent is bridged to start a conversation on one of two ways:
   - Media Server bridges the RTP streams of the customer and agent leg.
   - SIP Server redirects RTP to stream directly from the customer to agent.

CPD on the Media Gateway

- When CPD for an outbound call is performed on the media gateway, Media Server performs no operations and the SIP INVITE request contains no MSML instruction.

CPD on Media Server

- CPD for an outbound call is performed on Media Server in two phases: preconnect and postconnect. SIP Server provides a campaign identifier in the initial `INVITE` request for both call legs to the Media Server. The Resource Manager uses this identifier to ensure that the outbound call and the agent call legs that are associated with the same outbound campaign, land on the same Media Server instance.

Media Bridging

- Media Bridging occurs when the customer call leg and the agent call leg on the Media Server are joined. In larger outbound campaigns it is possible that the customer and agent that SIP Server has selected could be on different Media Server instances. In this case, it is not possible to use the join operation, therefore SIP Server uses a media transfer.

Transfer Mode

Unlike ASM mode, Transfer mode does not require the agent to be pinned directly on the Media Server. In Transfer mode a call is placed to the customer and then, it is connected to the first available agent. The `TMakePredictiveCall` event is used to perform this operation.

In this scenario, CPD might be done on the Media Server or a treatment might be required to apply to the customer leg when the call leg is connected.

If Media Server is required on an outbound call for the TMakePredictiveCall event on a route point, SIP Server must reuse the same SIP dialog with Media Server for both CPD and the apply treatment event. SIP Server drops the Media Server call leg only when the call is routed to an agent.

Proactive Notification

Proactive notification is used to send outbound calls to customers and execute IVR applications. SIP Server sends a call leg to the Media Server to execute the CPD, if it is required in the following operation sequence:

1. The GVP VoiceXML dialog is prepared before the outbound call is placed.
2. CPD is applied to the outbound call (if required).
3. The answering-machine *beep* detection is activated.
4. The GVP VoiceXML dialog is started (after the CPD).

A separate dialog is started at the same time to perform CPD, which means there are actually two dialogs running concurrently. An MSML `<dialogprepare>` tag is used to enable the Media Server to fetch and compile a

VoiceXML page from the Web server. (If GVPi is used, the Media Server makes the MSML `<dialogprepare>` tag a *no-op*, which means the VoiceXML page is fetched but not compiled.) The prepared dialog does not start until SIP Server sends an event to the dialog object to start the media.

Note: GVPi was not included in the GVP 8.1.5 installation package, but is still supported when deployed with Media Server 8.1.4.

The VP Media Control Platform (Media Server) accepts generic Universal Resource Indicator (URI) parameters from the initial SIP `INVITE` request and applies the URI parameters to any VoiceXML dialog that is started with `<dialogstart>` or `<dialogprepare>`. In this way, the SIP interface to VoiceXML service conforms to the standards that are defined in *draft-ietf-mediactrl-vxml*

When the Resource Manager receives an incoming call for an MSML service, it also accepts a URI parameter to target an IVR Profile. Resource Manager then passes the IVR Profile service parameters to the Media Control Platform (Media Server).

Conferencing

Media Server supports the creation and management of conferences through its MSML interface. MSML conferencing offers improved functionality over NETANN conferencing because it enables low level control of each conference participant, and provides the possibility of SIP Server to offer enhanced conferencing and supervision in the future.

For more information about the conference roles that are supported in MSML see Appendix B on page 159.

It is important that all media sessions land on the same instance of Media Server so that it can provide conferencing and join the media between the two connections directly. The Resource Manager enables SIP Server to define a conference identifier in the `REQUEST URI` and can guarantee additional calls with the same identifier to land on the same Media Server instance. The application server (SIP Server) is responsible for guarantying the uniqueness of the conference identifier across multiple application servers (similar to the NETANN syntax for conferences).

Note: This conference identifier is not the same identifier as the one that is defined within MSML. The MSML identifier is used by the Resource Manager for distributing incoming `INVITE` requests to the same Media Server instance.

Media File Types and Archives

Genesys Media Server can handle multiple audio and video codecs and supports `.wav` and `.au` media files for audio playback, based on the value of the `annc.defaultaudioext` option in the `netann` section or the `defaultaudioext` option in the `msml` section.

Media Server also supports binary files for video playback that use either H.263 or H.264 encoding. Binary video files can contain pictures with CIF or QCIF sizes and arbitrary frame rates (up to a maximum of 30 frames per second) when they are using H.263 encoding. The FOURCC code must be either H.264 or X.264 frames with a 4-byte start code. For the configuration options that enable you to adjust the picture size and frame rate, see the *Genesys Voice Platform 8.5 Configuration Options Reference.*

Note: The G.722, G.729, AMR, and AMR-WB codecs are disabled by default. You can enable these codecs by using the `codecs` option in the `mpc` section. If the default value for this option is retained, the less resource-intensive codecs are used first.

Supported Codecs

Table 7 shows the audio codecs that are supported by the Media Server, and the associated file names.

Table 7: Supported Audio Codecs and File Names

Audio codec name	Name in	File name
AMR	`amr/8000`	`<name>.amr`
AMR-WB	`amr-wb/16000`	`<name>.amr-wb`
G.711 Mu Law	`pcmu/8000`	`<name>_pcmu.<ext>` or `<name>_mulaw.<ext>`
G.711 A Law	`pcma/8000`	`<name>_pcma.<ext>` or `<name>_alaw.<ext>`
G.722	`g722`	`<name>_g722.<ext>`
G.726-16	`g726-16/8000`	`<name>_g726-16.<ext>`
G.726-24	`g726-24/8000`	`<name>_g726-24.<ext>`
G.726-32	`g726-32/8000`	`<name>_g726-32.<ext>`
G.729	`g729/8000`	`<name>_g729.<ext>`
G729a	`g729/8000`	`<name>_g729a.<ext>`

Table 7: Supported Audio Codecs and File Names (Continued)

Audio codec name	Name in	File name
G729d	g729d/8000	<name>_g729d.<ext>
G729e	g729e/8000	<name>_g729e.<ext>
GSM Full Rate (GSM 6.1.0)	gsm/8000	<name>_gsm.<ext> or <name>_gsmFR.<ext>
MS-GSM (Microsoft GSM)	—	<name>_msgsm.<ext> or <name>_gsmF.<ext>

The media file can be encoded by using either the G.729 or G.729a codec. The format of the encoded data is exactly the same, although the coding and decoding algorithms are different for each codec.

On the Real-time Transport Protocol (RTP) stream, Media Server treats the G.729a and G729a with Annex B codecs in the same way as Stream Manager. There is no discrepancy in the two functions. When data packets are sent, Type B packets are never sent, whether the Annex B option is negotiated or not. When data packets are received, Media Server interprets the packet and can interoperate with the codec, whether it is an Annex B packet or not. However, when is being negotiated, Media Server does not advertise that Annex B is supported.

Table 8 shows the video codecs that are supported by Media Server and their associated file-naming conventions.

Table 8: Supported Video Codecs and Video File Name Conventions

Video codec name	Name in	Stand-alone name
H.263	H263/90000	<name>_h263_XCIF=mpi, <name>.avi, or <name>.3gp
H.264	H264/90000	<name>_h264_profile-level-id=mpi, <name>.avi, or <name>.3gp

Real-Time Transcoding

Media Server can automatically convert a codec to another format, when the encoding of a file does not match that of the negotiated codec. Session Description Protocol (SDP) generation includes all configured codecs, not just those codecs for which the media file is found. However, files that have the proper media available still take precedence.

Play Cache

The play cache applies to all codecs. It supports multiple tracks for the same prompt, each track corresponding to particular endpoint codec settings and media type.

Formerly, the Media Server transcoded in real time, as needed for the call, when performing playback. Now, the Media Server transcodes the first time, then caches the transcoded file, and re-uses it for future calls with the same requirement. This approach saves CPU resources and accommodates a far greater number of simultaneous calls.

Some Play Cache Characteristics

The play cache supports prompts with these characteristics:

- `http:/// URL`
- `https:// URL`
- `file:// URL`
- `rtsp:// URL`
- `qtmf:// URL`

The play cache supports the use of tracks that are generated offline, copied to the cache directory, and then read by the MS upon startup.

The play cache does not handle...

- HTTP streaming prompts
- prompts that include text overlay.
- RRU playback.

The play cache does...

- Check to see if the media content of a prompt has changed. If it did—then the play cache regenerates the tracks, with the new recording, when the prompt is next played. You can set the period for this checking with the appropriate configuration parameter:
  - The `mpc.playcache.checkversiontime` parameter applies for rtsp and qtmf prompts. The prompt source is checked for content changed upon the next play for which the time since the last check exceeds the `mpc.playcache.checkversion` configuration value.
  - The fetching module parameters for refreshing content apply for http, https, and file prompts. Each play will always use the latest source content provided by the fetching module.
- Limit the amount of disk space used to a value set by a configuration parameter. When deleting is required to remain below this limit, the cache content for the least recently played prompts will be deleted first.
- Delete the content for prompts that have not been played within a period of time the you can specify in a configuration parameter.

Media File Archives

Media Server supports `.wav` media files that are packaged in an uncompressed file archive. File archives can be specified as follows:

- `music/in_queue.wav`
- `music/on_hold.wav`
  - Media Server can also support the `music/QTMF` file structure, which describes the quad-tone multi-frequency cadences that can be used in place of other media files.

Note: Before you upgrade to Media Server, Genesys recommends that you unzip (that is, uncompress) all of your `.zip` archives. Media Server does not support `.zip` archives.

Media Server uses the following method to search for media files:

1. If the file name that is specified in the `INVITE` request or the `PLAY` request has a `.wav` or `.au` extension, that file is used for playback. File encoding does not have to match the value of the `codecs` option in the `mpc` section; transcoding is performed, if necessary.

   Note: Media Server does not support the specification of the full name for video files in the request. Because video playback is usually accompanied by an audio track, the request must use the base name (resulting in Media Server constructing different names for the video and audio components based on the codec selection).

2. If the name that is specified in the `INVITE` request or the `PLAY` request does not match an existing file, Media Server assumes that the base name is specified and adds a suffix or extension based on the following conditions:
   - If a directory that has the same base name exists in the [netann].annc.basepath option, or the [msml].play.basepath option, Media Server extracts the media file that has the matching codec to the negotiated codec in, so that transcoding does not occur. The filenames for each supported codec are listed in Table 7 on page 48 and Table 8 on page 49.
   - If the media file exists, Media Server adds the appropriate codec suffix or extension. The media file can be recorded in a different codec and used for playback, but the file suffix must match the codec that is specified in the file.
3. If a media file is still not found, and the specified file name starts with `music/`, Media Server attempts to retrieve the tone description from the `music/QTMF` file. If the tone description is found, Media Server generates the media stream as Linear PCM16 audio and then transcodes it to the correct encoding for playback.
4. If all of the previous steps fail, a `404 Not Found` response is sent.

Any folder information that is present in the archive is ignored by Media Server.

For audio data (Table 7 on page 48), the file names that are specified in the archive must be the same as the short codec name that is specified in the Session Description Protocol (SDP) and in the Media Server options, with the `.wav` extension:

For video data (Table 8 on page 49), the file names that are specified in the archive must contain the codec name and format specification:

Tone Generation

Media Server supports two types of tone generation, standard telephone tones and DTMF.

Standard Telephone Tones

Instead of keeping the prerecorded standard telephone tones, Media Server generates tones as needed from the descriptions that are stored in the `music/QTMF` file, which is stored in the standard Genesys text configuration format.

The following two sections must be included in the `music/QTMF` file:

- `file`—Describes the cadences for each allowed `music/name` parameter from the request. The element name does not include the `music/*` prefix. The value, if formed, is as follows:

  ```
  file-value ::= (cadence-description-list)

  cadence-description-list ::= cadence-description |
  cadence-description-list, cadence-description

  cadence-description ::= tone-name = duration | x repeat-counter =
  (cadence-description-list)
  ```

  Where:

  - `tone-name`—Refers to the element in the `tone` section.
  - `duration`—Any integer that specifies the duration of that tone in 10 millisecond frames.
  - `repeat-counter`—An integer that specifies how many times the sequence in the parenthesis is repeated.

- `tone`—Defines the multi-frequency tones that are used for cadence generation, as follows:

  ```
  tone-value ::= (frequencies, amp = amplitude)

  frequencies ::= frequency | frequencies, frequency

  frequency ::= f n = frequency-value-in-Hz

  n ::= 1 | 2 | 3 | 4
  ```

Where:

`amplitude`—Refers to the volume of the generated signal as a percentage of the maximum volume.

Note: The tone-definition file that is included in Media Server is for the United States and Canada. The definitions must be changed for other countries, or alternative definitions must be supplied as prerecorded files and placed into the `music` folder. However, if a media file exists, Media Server uses that file instead.

Dual Tone Multi-Frequency

Upon receiving a Dual Tone Multi-Frequency (DTMF) event in an RTP stream, Media Server sends the `INFO` message with the `application/dtmf-relay` payload that contains the played digits and the duration (in milliseconds). For example:

```
Signal= 1
Duration= 160
```

This message is always sent in the context of the existing SIP dialog.

Media Server recognizes the `TELEPHONE-EVENT` mime-type as valid for telephony events listed in RFC 2833.

Note: Genesys Media Server supports RFC 4733 for both inbound and outbound DTMF, tones and continues to support RFC 2833 for backward compatibility. For a complete list of the supported standards and specifications for GVP (including the Media Control Platform), see the *Genesys Voice Platform 8.5 Deployment Guide*.

Configuring DTMF Tone Generation

You can configure DTMF tone generation in the following ways:

- Configure Media Server to send DTMF tone information in one of three ways: digitized tones, SIP `INFO`, or telephone events, as described in RFC 2833. Selection of one or the other is made by using the `[mpc].rtp.dtmf.send` and `[mpc].rtp.dtmf.receive` configuration options, which sets the default type.
- Set the default duration of DTMF tones and events by using the `[mpc].dtmf.duration` configuration option. The default duration can be overridden, as required.
- Set the delay before starting DTMF tone generation by using the `[mpc].dtmf.gap` configuration option.

Note: Digitized tones are reliably delivered only when a low-compression codec, such as G.711, is used. Therefore, when using the tone method, Media Server gives higher priority to `pcma` and `pcmu` codecs.

DTMF Tone Generation to the Connection Endpoint

Use the following URI (which conforms to RFC 4240) in the `INVITE` message to request DTMF tone generation to the connection endpoint:

```
sip:annc@<SM_hostport>;play=dtmf:<digits>;digit-duration=t[;
<annc_parameters>]
```

Where:

- `digits`—Specifies the DTMF tone sequence that is to be sent, by using the following characters:

`0-9, *, #, A-D`	DTMF digits and special DTMF events
`p`	Pause at half the digits duration

- `digit_duration`—Sets the duration of each DTMF event, in milliseconds; if it is not specified, the configured default value (configuration option `[mpc].dtmf.duration`) is used.
- `annc_parameters`—Additional announcement parameters, such as `content-type`, can be specified. The following two announcement parameters are ignored:
  - `repeat`—The DTMF tone sequence is always generated just once.
  - `record`—DTMF tone generation cannot be combined with recording.

When Media Server receives the request, it generates the requested sequence as an audio tone (as defined in ITU-T Recommendation Q.23) and as out-of-band RTP packets (as defined by RFC 2833).

Example For example, to request Media Server to send the DTMF tone sequence `*80,6504661410`—where the comma (`,`) designates a pause, using a duration of 100 ms and the default generation method—use the following URI:

```
sip:annc@hport:6050;play=dtmf:*80p6504661640;digit-duration=100
```

Applying Audio Tones During Recording

Government regulations require some deployments to periodically generate an audio tone, to notify the participants in a call that the call is being recorded.

Media Server can generate a periodic audio tone from a URI when an active call recording is started. The following parameters can be configured in the IVR Profile for call recording:

Parameter	Description
audiosrc	The URI of the audio tone. If the URI is set to empty string, or not defined, or resolves to a bad URI, then no audio tone is applied to the call. No other notifications are generated by media server (i.e., MSML events) when no audio tone is being applied.
tonesilenceduration	Length of time between playing the audio tone in milliseconds. **Note:** If audiosrc is defined and tonesilenceduration is *not* defined, the length of silence is set to the delay time specified with conference.record_tonestartdelay — default value is 1500 ms.

These parameters can also be configured as service parameters in the IVR profile, and there they are treated as the default values for the IVR Profile. They can be overridden by AttrExtensions in RequestPrivateService on a per-call basis.

For example, the parameters would be placed into the IVR Profile section gvp.service-parameters:

```
audiosrc=http://example.com/tone.wav
tonesilenceduration=30000
```

When a participant is added to the call recording for any reason (for example, conferencing in an agent to the call or single step transfer to another agent), the new participant will hear the audio tone in the next audio tone played in the original conversation. If tonesilenceduration is set to a large value, then the new participant may not hear the audio tone for a long time until the next audio tone is played.

Video Functions

Media Server supports two types of video functions, Push Video and Video Fast Update.

Push Video

Media Server supports Push Video functionality when SIP Server is deployed. For information about how to enable and manipulate this functionality, see the *Framework 8.0 SIP Server Deployment Guide*.

Media Server can play files that contain raw video streams that are encoded with H.263 or H.264 video codecs only.

Note: Genesys does not provide a utility for converting either uncompressed video or compressed AVI files into these formats. You must use either commercial or open-source third-party converters for this purpose.

Video Fast Update

Media Server supports requests for video fast update in a video conference, as defined by the RFC 5168 standard.

When a video fast update request is received from a conference leg, the request is forwarded to the video source of the conference leg. For example, if participant A and B are in a video conference, and participant A is watching the video from participant B. The request is forwarded to B, when Media Server receives a video fast update request from A.

Media Server also supports the generation of video fast update requests in a video conference. An update request is generated to the video source in the following situations:

- When a new participant joins the conference and selects a participant as the video source, the video fast update is sent when the new participant is ready to accept video. (In other words, when the video is negotiated in the SDP in a SIP `ACK` message, and the negotiated video contains a non-zero IP and port, and the video channel is not put on hold.)
- When an existing conference participant switches to a different video source.
- When an existing conference participant's video capability is re-negotiated, and the video capability changes from inactive to active, or the remote connection changes from invalid to valid.

Video Conferencing

Media Server supports many advanced conferencing features, such as unlimited video conference participants, several types of mixing layouts, loudest active speaker, and video text overlay. All of these features are described in detail in this section.

Unlimited Video Conference Participants

Media Server supports an unlimited number of video conference participants, restricted only by the application specification and capabilities of the host machine. Video input from two or more conference participants is mixed together to make up the video output stream.

You can select the video type on a per-conference basis—either video mixing or video switching (where each participant is sent a single video input). The output type is inherited from the session that created the conference.

Supported Mixing Layouts

Media Server supports the following video-mixing layouts, as specified in the *draft-ietf-mediactrl-mixer-control-package-14*.

- Single view
- Dual view
- Dual view crop
- Dual view 2x1
- Dual view 2x1 crop
- Quad view
- Multiple 3x3
- Multiple 4x4
- Multiple 5x1

All layouts have a black background.

Media Server automatically selects the video layout that will be used, based on the number of conference participants providing the video input. The number of participants is mapped to the video layout per session and is configurable, using the `[conference] video_mixer_layouts` option.

Loudest Active Speaker

Media Server periodically determines which participant provided the loudest audio during the most recent sampling period. This participant is designated as the loudest speaker (or the active speaker). The duration of the sampling period is set by configuring the `[conference] active_speaker_update_time` option.

Media Server also keeps track of the last time each participant became the loudest speaker. It select the video inputs of the participants that were most recently determined to be the loudest speaker, and includes them in the current video layout. For example, if a layout has four positions, the participants that were the four most recent loudest speakers, are selected.

The participants are not in any particular order in the video layout. If a participant becomes the most recent loudest speaker, that participant replaces the least-recent loudest speaker in the layout.

Loudest Position in the Video Layout

If a video layout has a loudest position, Media Server displays the most recent loudest speaker in that position. The most recently loudest speaker is also displayed in one of the normal display positions in the mixed video output with a highlighted white border. Of all the currently supported layouts, only the multiple-5x1 layout has a loudest position.

Media Server produces one video mix output and sends it to all of the conference participants that accepted a video output. Unlike audio mixing,

where the participants cannot hear their own audio, each participant in the video mix sees their own video input if they are one of the selected inputs.

No Video Image Display

A *no video* image is displayed in the following scenarios:

- If a caller is connected to a video-mixed conference, but does not provide video input.
- If the SDP includes video, but the video is not received by the platform (for whatever reason).
- If the caller sends video, but there is a lapse before the caller sends video again

The *no video* image is input from a configurable `.jpeg` file at system startup. A default image file is provided.

If threaded outputs for conferences is enabled, Media Server transcodes audio output and transmits to the media layer with threads, separate from the main mixing thread.

Video Text Overlay

Text can be overlaid in video frames that are being played out, by using the YUV420 video format. Transcoding occurs to and from this format.

Multiple text overlays can be specified, together with their starting position, a font name and style, font size, font color, and background color.

Font Size

The starting position is defined by using `x + y` coordinates, where `x` is the distance from the left margin and `y` is distance from the top margin. Each of these values is specified within a scale of 0 to 1000 pixels, and is scaled to the width and height of the video frame. Both coordinate's default value is `0`.

Note: If the font width is not specified, Media Server uses the value set for the height (or size) in pixels. In other words, there is no additional scaling for width.

Font Name and Style

To specify a font, Media Server uses the standard name and style in a font file. You can specify multiple font directories, by using the `[mpc] font_paths_<platform>` configuration option, where `<platform>` represents either Windows in or Linux. Media Server reads the name and style from all font files in the configured directories at start-up, and caches this information, so that when text overlaid is used, the file name can be looked up, based on name and style.

Media Server supports the True Type (`.ttf`), and PostScript Type 1 (`.pfa` or `.pfb`) font file formats.

Font Color and Background

The font color and background color are specified, by using the HTML hexadecimal triplet RGB color code with an optional number sign (`#`) at the beginning. Media Server supports `newline` and `tab` characters in the ASCII text. The `newline` character moves the text that follows to the beginning of the

next line. The `tab` character advances the next character to the next tab stop, with a tab size and length of 4 times the font size and width.

To ensure the video-mixing and text overlay features work properly, enable the following video transcoders:

- H263—If any of the participants in the conference send or receive H263 or H263+ video.
- H264—If any of the participants in the conference send or receive H264 video.

An error notification is generated, if the required transcoders are not enabled.

H.263 and H.263+ Video Format Transcoding

Media Server provides H.263 video format transcoding with resolution down-scaling. The following rules apply:

- Resolution downscaling—If outgoing video packets exceed the maximum resolution as indicated by the remote `fmtp-line,` outgoing packets are down-scaled to the maximum resolution.
- Frame rate throttling—If the outgoing frame rate exceeds the maximum frame rate, as indicated by the remote `fmtp-line,` the outgoing frame rate is throttled to the maximum frame rate.

  Note: The frame rate is determined by the MPI that accompanies the resolution (`QCIF=1`) or by the level (`LEVEL=10`); If both are specified, level takes precedence.

- Bit rate throttling—If the outgoing bit rate exceeds the bit rate, as indicated by the remote `fmtp line,` the outgoing bit rate is throttled to the maximum bit rate.

  Note: The bit rate is determined by the level (`LEVEL=10`). If level is not specified, Media Server does not impose a maximum bit rate.

For detailed information about level and its maximum frame and bit rate, see Appendix X.4 in the *ITU-T Recommendation H.263 2005* specification.

Media Server performs transcoding from H.263 or H.263+ to H.264, but it is enabled only if the `mpc.transcoders` configuration option includes an H.263 token. If transcoding is disabled, the Media Server skips H.263 or H.263+ to H.263 or H.263+ transcoding and the video media is passed as is, with no generated error. However, H.263 or H.263+ to H.264 transcoding does generate a media error.

H.264 Video Format Transcoding

Media Server provides H.264 video format transcoding with resolution down-scaling. The following rules apply:

- Profile (`profile_idc`) transcoding—If the profile of the outgoing video packet does not match the profile, as indicated by the remote `fmtp-line`, the outgoing packets are transcoded to the desired profile.

  Note: The H.264 transcoder/decoder component does not support advanced profile high 4:4:4 (`profile_idc=0xF4`), profile extended (profile_idc=0x58), or profile CAVLC 4:4:4 (`profile_idc=0x2C`). When H.264 transcoding is enabled, SDP negotiation blocks the use of these profiles.

- Resolution down-scaling—If the outgoing video packet exceeds the maximum resolution, as indicated by the remote `fmtp-line`, it is down-scaled to the maximum resolution.
- Frame rate throttling—If the outgoing frame rate exceeds the maximum frame rate, as indicated by the remote `fmtp-line`, the outgoing frame rate is throttled to the maximum frame rate.
- Bit rate throttling—If the outgoing bit rate exceeds the bit rate, as indicated by the remote `fmtp-line`, the outgoing bit rate is throttled to the maximum bit rate.

  Note: The maximum resolution, frame rate, and bit rate is determined by the receiving side's `level_idc` parameter, which is defined by the SDP `fmtp profile-level-id` parameter. For more information about these parameters, see Table A.1 of the *ITU-T Recommendation H.264* specification.

Media Server performs transcoding from H.264 to H.263 or H.263+, but it is enabled only if the `mpc.transcoders` configuration option includes an H.264 token. If transcoding is disabled, the Media Server skips H.264 to H.264 transcoding and the video media is passed as is, with no generated error. However, H.264 to H.263 or H.263+ transcoding does generate a media error.

Maximum supported resolution for transcoding to/from is 720p.

VP8 Transcoder Support

The VP8 transcoder is not required if the file being played contains VP8 video, and/or the caller has requested VP8 video.

Text Overlay UTF-8 Support

Text overlay supports UTF-8-encoded multi-byte characters. For release 8.1.6 and above, the NGI is the only component providing characters, and NGI

provides only UTF-8 characters to the Media server, transcoding from other encodings if it can.

Characters which are not UTF-8 are treated as ASCII characters (1 byte each) and if the font file needed for displaying the UTF-8 encoded character is present in the system, then the appropriate character is displayed.

If the font file needed for displaying the UTF-8 encoded character is not present in the system then a question mark (?) or some other symbol—depending upon the available font—is displayed.

A warning is logged if the string passed to the media layer does not appear to be in UTF-8 format.

Call Recording

Media Server supports the call-recording methods and functionality that are described in the following topics:

Regular Method

Media Server performs non-emergency call recording of a two-way call by converting the call into a two-party conference. Both call participants are re-invited to the conference call by sending the following URL in the `INVITE` messages to Media Server:

```
sip:conf = conf-ID @MS-hostport; record = record-URL
```

Where the parameter specifies the file name for recording. It is processed by using the same method that the Recording Announcement Service uses. For more information about this service, see “Record User Announcement” on page 70.

Note: The parameter values must be identical for both parties.

The following scenario describes a two-step process that is used for codec negotiation:

1. SIP Server sends an `INVITE` message with the capabilities from the original conversation with the addition of the `a=inactive` attribute, in order to notify Media Server of the endpoint capabilities. Codec negotiation for each party in the conference is done independently, therefore, Media Server must know the endpoint capabilities in advance, to avoid selecting different codecs for different parties, and to minimize transcoding.
2. SIP Server sends a re-`INVITE` request, to which Media Server responds in the same way that it does for a regular conference (except that it uses only those codecs that are supported by both participants).

Media Server configuration controls which of the following files the call-recording service produces:

- An audio file with mixed streams (recommended)
- Separate pcap files with all of the packets captured for each stream

Manual Method (Emergency Recording)

SIP Server implements Call or Conference Call Recording by initiating a conference call with a recording device, such as Media Server, or with a third-party recording device. A new SIP Server call leg is created for every recording performed by Media Server.

Note: In distributed environments, Media Server can be installed on different hosts and on different operating systems, but still shares a single storage area for media files. In this scenario, accessibility for all hosts to the shared storage must be determined and configured at the Administration level.

Dual-Channel Call Recording

Media Server performs many types of recording functions, including advanced MSML server functions, such as, dual-channel call recording.

An MSML call is initiated when the user-part of the `Request-URI` from the incoming SIP `INVITE` message is in the format `msml[=<48ehavio>]`. The `<48ehavio>` value is optional and specifies a conference ID. (The conference ID parameter is used by the Resource Manager.)

When dual-channel (audio) call recording is initiated, the conference ID is used to match the two calls that are in the same session. The `record` parameter (with no value) must be included in the SIP URI, in the following format:

```
sip:msml=<unique_id>@mediaserver;record;dn=<dn>;recdest=<rec_address>[;
recmediactl=<1 | 2>;recorder_parameters...]
```

where:

- <dn>—The DID that is associated with the call leg that is sent to the recording server. It is used to identify the audio stream and is inserted by SIP Server.
- <rec_address>—The URI for the address of the recording server. It is inserted by the Resource Manager, based on the IVR Profile. It must be specified, however, it might be overridden by the dest parameter that is specified in the MSML request.

If the recmediactl parameter value is 2, separate SIP sessions are established on the recording server—one for each recording stream (default behavior). If the value is 1, a single SIP session is used for both streams (with multiple m-lines in the Session Description Protocol [SDP]). This parameter is inserted by the Resource Manager, based on the IVR profile. Its value can be overridden by the corresponding MSML parameter, although this is not typical or required.

Dual RTP Streams

Media Server can replicate the RTP streams of two inbound calls that are part of a Call Recording session (indicated by the Request-URI) to a third-party recorder by using the MSML. SIP Server initiates the request with an INFO message by using MSML.

The SDP and other connection-specific parameters are passed in a specific attribute of this message. The attribute is also used to start multiple additional recordings, pause, stop, and restart streaming.

Interoperability With Third-Party Recorders

Media Server supports dual-channel call recording for third-party recorders by extending the MSML MPCConference class. Audio only and audio-plus-video dual-call recordings are supported, by using two SIP sessions for each recording session, depending on the start parameter. However, Media Server always uses an independent RTP session to stream audio for each call in the correct order. Media Server also supports multiple recordings of the same session.

Note: Media Server 8.1.5 and later supports RTP activities on both IPv4 and IPv6 interfaces.

To establish SIP sessions with the third-party recorder, Media Server interfaces with the recording client by using VRMInterface. It generates the SIP URI in the following format:

```
sip[s]:record=<unique_id>@<recorder_address>;calluuid=<uuid>;dn=<dn>[;<other_params>]
```

where:

- <unique_id> and <recorder_address>—Obtained from the recorder's id and dest parameters in the MSML request (the latter can also be from the Request-URI of the inbound call), and is the same for both channels.
- <dn>—Obtained from the SIP URI parameter of the corresponding inbound call leg. When one SIP session is used for both streams with the recorder, the otherDN=<dn> parameter is used also.
- <uuid>—Obtained from the value of X-Genesys-CallUUID SIP header of the inbound call leg. This is the same for both participant A and B.
- <other_params>—Obtained from the unknown parameters in the MSML request.
- sip[s]—The TLS parameters are obtained from the vrmrecorder.sip.transport.[n] configuration option.

Support for SIP RE-INVITE Messages

If the inbound connection is modified, Media Server supports sending a RE-INVITE message to the recorder. It also supports pause and resume of a recording session by sending RE-INVITE messages with active and inactive SDP offers, respectively.

If Media Server receives a RE-INVITE message from the recorder, it supports changing the RTP stream destination.

Modification of SDP

Bandwidth Line in SDP

Some third-party vendors require bandwidth lines in the SDP for their devices. To inter-operate with these devices, Media Server adheres to the RFC 3264, (Section 5.8) definition for SDP bandwidth lines, and handles them in the following way:

If the SDP in the initial SIP INVITE contains the b=xxxx bandwidth line, Media Server includes this line in the SDP in its 200 OK response.

Media Server supports two options in its mpc section to configure bandwidth; sdp.audiobandwidth and sdp.videobandwith. If these configuration options are populated with non-empty strings, when Media Server receives an empty SDP in the SIP INVITE, it includes the bandwidth line in its audio or video media description, with the bandwidth line equal to the same string value as the one that is specified in the sdp.audiobandwidth or sdp.videobandwith configuration options.

For example, if sdp.audiobandwidth=TIAS:38000, the audio media description in Media Server's SDP includes b=TIAS:38000

m-Lines in SDP

Media Server supports the answerwithonecodec configuration option in the mpc section, which enables Media Server to generate an SDP response with one media codec only in each m-line, plus the telephone-event parameter in the audio m-line.

In each m-line, Media Server accepts the first supported codec in the offer and returns it in the response. In addition, if the `telephone-event` parameter is enabled, it is returned in the audio m-line.

The `answerwithonecodec` option is disabled by default.

During SDP negotiation, the Media Server takes the first `sendonly` or `sendrecv` m-line that specifies a single port for that media type. Subsequent m-lines for the same media type are answered in the following ways:

- If the m-lines are offered as `sendonly` or `inactive`, Media Server responds with `inactive`.
- If the m-lines are offered as `sendrecv`, Media Server responds with `sendonly`.

Note: In this scenario, a remote IP address, such as 0.0.0.0 or :: puts the RTP or RTCP stream on mute.

IPv6 Support in M-Lines

The Media Server accepts IPv6 or IPv4 SDP connections in an offer if it is configured to support those interfaces. Each m-line is treated independently during negotiation against an IPv4 or IPv6 connection attribute.

The Media Server uses the `[mpc] preferredipinterface` configuration parameter (IPv4 or IPv6) to determine which IP address to include in the root connection line of an SDP offer or answer, and to decide which IP interface to use when media is offered on a particular m-line. It rejects any SDP answer that attempts to change the IP interface version from the one that was last sent.

fmtp-line in SDP

Media Sever supports an `a=fmtp` line in the SDP negotiation when H.263 transcoding is enabled. Both H.263+ (H.263-1998) and H.263-2000 parameters are supported.

The `mpc.h263.fmtp` configuration option defines a list of `fmtp-lines` that are use to make an offer or as remote SDP offer acceptance criteria. This option's parameters and fmtp-lines are supported, as defined in Section 8 of RFC 4629.

Negotiation fails if any of the following parameters are present in the remote `fmtp-line`:

- `K`
- `BPP`
- `PAR`
- `HRD`
- `CPCF`
- `INTERLACE`

The following rules apply when `fmtp-lines` are used as remote SDP offer acceptance criteria:

- At least one `fmtp line` (defined by the configuration) must match for the SDP offer to be accepted.
- If the remote offer defines the `PROFILE` parameter, the `fmtp` configuration must define exactly the same `PROFILE` parameter.

- If the remote offer defines the LEVEL parameter, the fmtp configuration must define the same LEVEL parameter or a level above.
- The maximum resolution (frames per-second [fps]) required by the remote offer must be less than or equal to the maximum resolution (fps) that is defined in the `fmtp` configuration.
- If the remote offer supports the `K`, `N`, `CPCF`, `HRD`, or `INTERLACE` parameters, the offer is rejected. The `PAR` and `BPP` parameters are ignored.

Note: The only supported value for the `PROFILE` parameter is `0` (zero).

The maximum supported resolution is 4CIF, which corresponds to the maximum level of `70`.

For H.264, the maximum supported resolution is 720p, which corresponds to the maximum level of `3.1`. SDP can negotiate higher levels in `fmtp`, but the resolutions and the frame rates listed for higher levels are not supported.

Crypto Line in SDP

Media Server shall generates the SDP offers to the third-party recorder, based on the negotiated SDP of the inbound calls. However, the crypto line in the SDP is generated, based on the `mpc.vrmrecorder.srtp.*` configuration options.

When separate SIP sessions are used for each RTP stream, Media Server uses a single media line per SDP, and uses two media lines in the SDP when one SIP session only is used.

AMR-NB Portion of SDP

The AMR-NB portion of an SDP offer/answer is declined by default, if the parameters in the `a=fmtp-line` turns on any of the following features (as defined in Section 8.1 of RFC 4867):

- `crc`
- `robust sorting`
- `interleaving`
- `channels (assigned to a value not equal to 1)`
- `mode-change-capability`
- `mode-change-period`
- `mode-change-neighbor`

The `max-red` parameter is ignored.

Differentiated Services Field for Outgoing RTP/RTCP

Media Server supports configuration of the Differentiated Services Field (DS Field) for outgoing RTP/RTCP packets, in accordance with the RFC 2474 standard. This includes RTP/RTCP packets sent out to the user and those that are sent out for ASR sessions

The DS Field in Audio and Video RTP packets and RTCP packets are configured rtp.tos, mpc.rtp.tos.video, and rtcp.tos options in the mpc section. If

the `default` value for these options is used, the DS Field in the outgoing RTP/RTCP packets is not enabled.

The standard values are:

- `o 0x00-DS`—Field not set
- `o 0x10-IPTOS_LOWDELAY`—Low-delay type of service
- `o 0x20-IPTOS_PREC_PRIORITY`—Priority precedence
- `o 0x40-IPTOS_PREC_CRITICAL`—Critical precedence
- `o 0xB8-DiffServ EF`—Expedited forward

MP3 Play and Record Audio Format Support

Media Server supports MP3 audio playback on two channels—mono or stereo audio channels. It can provide playback at all of the sampling rates that are supported by the mpeg 1, 2, and 2.5 audio standards—such as 8, 11.025, 12, 16, 22.05, 24, 32, 44.1 and 48 KHz, and at all of the bit-rates that are supported by the same standards. For stereo playback, the channels are combined into one.

Media Server supports RTP transport for MP3 audio format using the `X-MP3-draft-00` format when using RTSP.

Media Server records MP3 audio formats to a local file on a mono audio channel at the sampling rate that is specified by [mpc]mp3.samplingrate configuration option and at a bit-rate that is specified by the [mpc] mp3.bitrate configuration option. mpeg 1 - Layer 3 is used for sampling rates of 32 KHz or higher and mpeg 2 - Layer 3 is used for any lower sampling rates.

Audible Alert

Media Server can provide an audible alert when a call recording starts. To enable this functionality, configure the `Voice over IP Service` DN (with the `VoIP DN` service type) in Configuration Manager by adding an alert sound to the `request-uri` option. You can use the following example:

```
request-uri =
annc@MS;play=music/normal_5sec;repeat=1;record=recording/call-
```

Note: Specify the play and repeat parameters before the record parameter, because SIP Server includes the `Call UUID` attribute at the end of the `Request-URI` message.

File Creation

When it records calls, Media Server uses the following configuration options:

- `[mpc].maxrecordfilesize`—The maximum size of the audio file that is used for recording.

- [netann].record.maxrecordtime—The maximum recording time, in seconds.
- [netann].record.maxrecordsilence—The maximum allowable amount of time (in seconds) that silence can be detected during a recording. Specify additional recording parameters in the RecordUserAnnouncement treatment. For more information about how to configure this treatment, see the *Universal Routing 7.6 Reference Manual*.

When call recording is enable, based on a routing strategy, record extension is specified in the RequestRouteCall event.

File recording ceases when any of the following events occurs:

- SIP Server sends a STOP command (for example, after entering a keystroke combination that indicates that the caller has finished recording the announcement).
- The [netann].record.maxrecordtime interval has expired.
- The [netann].record.maxrecordsilence or the [mpc].maxrecordfilesize limit is reached.

When a RecordUserAnnouncement treatment is applied, SIP Server sends a STOPPED notification if the recording is interrupted and issues EventTreatmentEnd or EventTreatmentNotApplied.

When it records an incoming audio stream, Media Server combines the announcement file with the recording file. You can create the URL in the SIP Server INVITE request, as follows:

```
sip:annc@MS-hostport; record= record-URL [; play= prompt-URL]
```

The optional play parameter can refer to a media file that contains a prerecorded *beep* sound or other announcement. The record-URL parameter is used to generate the output file name, based on the following:

- If the file name that is specified in the URL includes the correct file extension, Media Server uses this name to record the file without any modifications. If the file name already exists, it is overwritten.
- If the file name is not specified, Media Server generates a unique number and the codec-specific suffix or extension.

The files are always recorded with the same codec that was used during the call.

Call Recording Filenames

For a third-party call recording, the recording is not generated by the Media Server, but the audio is streamed to the recorder. The file name is generated internally by the third-party recorder, based on the record_ID that GVP provides in the user part of the SIP Request-URI (sip:record=<record_ID>@<address>), and might not be visible to the end user.

However, you can search and find the recordings through the third-party recorder's user interface by using the record ID. The record_ID is a unique ID

and is specified in the MSML request that is sent by the SIP server to the Media Server. It is used for dual-call recording as a `<gvp:param>` with the name `id`. This ID is typically specified as an attribute in the `RequestPrivateService` T-Lib request for recording, or it can be generated by the SIP Server for full-time recording.

Recorded Files on Remote Network Paths

For both Window and Linux platforms, when you are creating recorded files that are located on remote network paths, Genesys recommends that you mount the remote paths on the local host and, when you are specifying them in the record URL, reference them as local drives and paths.

If mounting to a local drive is not an option and the network path is specified in the record URL as `file://\\remote_system\filedirectory` (`\\remote_system\directory` on windows), the Media Control Platform (Media Server) does not access or create any files on the remote path. That is because the Media Control Platform is usually running as a SYSTEM service and the remotely-shared network path cannot grant the Share permission to the Media Control Platforms SYSTEM account.

As a workaround, you can configure the Media Control Platform to run as a service and configured as a valid user, and grant the user the access permission to the remote paths shared folder. In this way, you enable the Media Control Platform to access the remote paths.

Note: Media Server can record the audio portion of a video call, but the recording leg in the conference call must contain the `confrole=monitor` attribute.

Recommended Codec

Although Media Server supports recording with any audio codec, Genesys recommends that you only use the G.711 codec when recording calls (unless there are specific reasons to use another codec). The `codecs` option in the mpc section must be set to `pcmu` (mulaw) or `pcma` (alaw). The CPU resources that are required for processing this codec are significantly fewer than for other codecs. In addition, the G.711 codec provides the best quality and is the most compatible with other software.

Media Server supports both mixed and `pcap` recording for conference recording, and can be configured by using the `[netann].conference.recordmode` parameter. The parameter default is mixed mode.

Record User Announcement

The call-recording function is supported by the `RecordUserAnnouncement` treatment. When SIP Server receives a `RecordUserAnnouncement` request, it sends the message to Media Server. When multiple Media Servers are deployed, SIP Server selects from among all of the available Media Servers by using the round-robin algorithm.

By default, the recorded user's announcement is saved into a folder named users. The file name that is specified in the `RecordUserAnnouncement` treatment can be any file name. The format of the recorded file is determined by the audio codec that is chosen during the negotiation procedure. For information about how to configure the `codecs` option in the `mpc` section, see the *Genesys Voice Platform User's Guide.*

Recording Servers and Clients

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:

- Provisioning third-party recording servers.
- Provisioning Media Server resources.
- Handling load balancing and failover of Media Servers.

Notes: Media Server initiates recording sessions to the recording server via Resource Manager, and for each session, Resource Manager selects a Recording Server resource from the provisioned `recordingserver` Logical Resource Group (LRG).

The options for provisioning the `recordingserver` LRG will vary with the deployment model of the recording vendor. Consult the load-balance-scheme parameter for your implementation.

Call Recording Definitions

In the context of call recording, the following definitions apply:

- Recording Server—The collector of the recorded media, including Media Server and all 3rd-party recording servers.
- Recording Client—The source of the recorded media; under file-based call recording that is Media Server.
- Communication Session—The call to be recorded.
- Recording Session—The SIP/RTP session that provides the recorded media.

Figure 3 on page 71 is a simple depiction of each element in a call recording session.

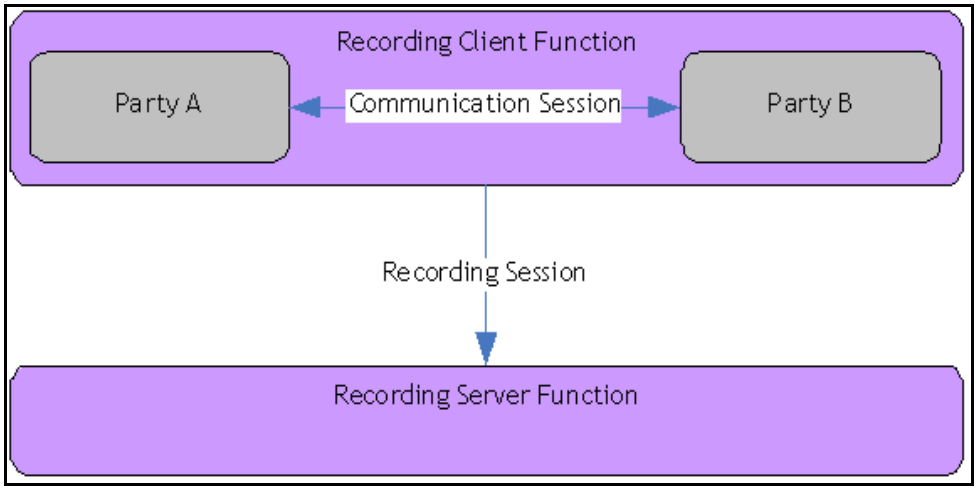

Figure 3: Elements in a Call Recording Session

Recording Clients

Media Server is the recording client—the source of recorded media.

When the Resource Manager receives a request with the `record` parameter in the `Request URI`, it identifies this as a request for the `recordingclient` service, and sends the request to the Media Control Platform. The Media Control Platform uses the recordingclient service type to provide the recordingclient service.

The Resource Manager manages the client-related resources by grouping them in the `recordingclient` resource group.

When the Media Server gets a configured list of SIP headers from inbound calls, it sets them into the outgoing SIP INVITE messages to the recorder, via the recording client.

Recording Servers

When the Resource Manager receives a request with the `record=unique_identifier` parameter in the user part of the `Request URI`, it identifies this as a request for the `recordingserver` service, and sends it to a third-party recording server. The recording server provides the service by using the `recordingserver` service type.

Unique identifiers enable the Resource Manager to distinguish between requests for different recording sessions. When the Media Server (acting as a recording client) initiates a recording request to a third-party recording server through the Resource Manager, and the `REQUEST URI` contains a unique

identifier, then the Resource Manager routes all requests that already share the same identifier to the same third-party recording server.

The Resource Manager manages third-party recording server resources by grouping them in the recordingserver resource group. The Resource Manager actively monitors the availability of recording servers and ensures that requests are forwarded to active instances only.

Load Balancing with Parallel Forking

The Resource Manager uses *parallel forking* to perform load-balancing for recording servers. This scheme enables the Resource Manager to send SIP requests in-parallel to all of the resources in a specific resource group.

Parallel forking is configured in the `load-balance-scheme` option in the `recordingserver` resource group. When parallel forking is enabled, the `port-capacity` option can also be configured to enable the Resource Manager to enforce, for the resource group, a resource usage limit that cannot be exceeded.

Monitoring Recording Servers and Clients

This section describes how the Resource Manager detects and monitors recording servers and clients in the Call Recording solution.

Monitoring Recording Clients

To monitor the status of recording clients, configure the `sip.proxy.release-recordingclient-session-on-fail` option in the `monitor` section of the Resource Manager `Application`, to determine if the resource that is used to initiate the recording session is offline (unavailable). If it is not available, then the option value dictates how new calls are routed. For example:

- If the option value is `true` (the resource is not available), then all associated `recordingclient` sessions are released and new calls are routed to the next available `recordingclient` resource.
- If the option value is `false` (the resource is available), then new calls that are joining the `recordingclient` session receive an error until the `recordingclient` session is released.

The default value for this option is `true`.

Monitoring Recording Servers

To monitor the status of recording servers, configure the `sip.proxy.release-recordingserver-session-on-fail` option in the `monitor` section of the Resource Manager `Application`, to determine if the resource,

that is to initiate the `recordingserver` session, is offline (unavailable). If it is not available, then the option value dictates how new calls are routed. For example:

- If the option value is `true` (the resource is not available), then all associated `recordingserver` sessions are released and new calls are routed to the next available `recordingserver` resource. This is the default configuration.
- If the option value is `false` (the resource is available), then new calls that are joining the `recordingserver` session receive an error until the `recordingserver` session is released.

Policy Enforcement and Resource Selection

Policy enforcement and resource selection for call recording clients and servers occurs in the same way as it does for any other GVP resource or request for service. The Resource Manager supports all of the same policy and resource configurations options. For more details, see the topics "Policy Enforcement" and "Resource Management" in Chapter 3: How GVP Works in the *Genesys Voice Platform Deployment Guide*.

Failover Mitigation

To prevent the Media Server from becoming a single point of failure for a communication session, the Resource Manager supports failover by notifying SIP Server that the Media Server is unavailable. This enables SIP Server to take alternative action to complete the call.

The Resource Manager registers with Management Framework's Solution Control Server (SCS) to monitor the status of the resources that it manages. When the SCS detects a resource startup or shut-down, it sends an asynchronous notification to the Resource Manager with the current status of the resource.

The SCS and Resource Manager use the following method to register and update resource statuses:

- If the SCS cannot determine the status of a particular resource, it sends a special `unknown` status for that resource, and the Resource Manager considers the resource to be offline (unavailable).
- If the SCS is unavailable, the Resource Manager receives a similar notification and attempts to contact the backup SCS (if one is configured) to register for resource status notifications.
- If a backup SCS is not available, the Resource Manager attempts to re-connect to the SCS periodically.
- If the Resource Manager cannot find the SCS, it does not update the status of existing resources.

To enable the Resource Manager to receive resource status notifications, ensure that both the primary and backup SCS are added as a connection in the

Resource Manager `Application`, and that the `monitor-method` configuration option value in the resource group is set to `mf`.

Routing Requests for Servers and Clients

To route requests for recording servers and clients, Resource Manager uses DNS procedures as defined in the RFC 3263 standard, which is supported by the Voice Platform's underlying SIP stack.

Exceptions to RFC 3263 Standard

This feature is enabled for clients or servers by configuring the `sip.transport.dnsharouting` option. However, the way in which the DNS procedure is used differs slightly for clients and servers, and the following exceptions exist:

- Client transactions adhere to "Client Usage," Section 4 of the standard, with the following exceptions:
  - `CANCEL` request
  - `ACK` request for non-2xx INVITE response
  - `ACK` request for `200 OK` responses when the UDP transport is used (No corresponding response exists to confirm the successful reception.)
- Server transactions adhere to "Server Usage", Section 5 of the standard with the exception of "Responses when the UDP transport is used".

Routing Functions

The Resource Manager generates an ordered target list by using the host name and port number of the server or client to which calls are routed. The `Request-URI` is used for SIP requests, unless the Route header is present, in which case the Route header takes precedence.The `Sent-By` field is used for the SIP responses.

The Resource Manager uses the following process to route client and server requests:

- If the destination data contains an IP address, the address and port number in the SIP URI specifies that target *only*. If the port information is missing, then the default port that corresponds to the transport is used.
- If the destination contains a host name (not an IP address) and a port, then an `A/AAAA` record (address record) look-up is used to generate the list of targets. The order in which the generated list is returned by the DNS query is retained.
- If the destination contains a host name (not an IP address) only, then service record (`SRV`) look-up is used to generate the list of targets. If the target contains non-IP data, the `A/AAAA` record is used to resolve the targets to IP addresses. The procedure outlined in the RFC 2782 standard is used to order the targets.

The Resource Manager tries the targets in order, iterating to the next target if one of the following occurs:

- Transport errors occur during TCP transport.
- No response is received for the SIP request within the configured timeout interval. The timeout interval is controlled by the `sip.transport.routefailovertime` configuration option.
- Timer `B` (the INVITE transaction timeout timer) or Timer `F` (the NON-INVITE transaction timeout timer) is exceeded and triggered.

The failed target is marked as unavailable and is not re-tried for the interval that is specified in the `sip.transport.routerecoverytime` configuration option. (Failed-target marking is performed for failed requests only, and not for failed responses.)

When the next target is tried for a SIP request, a new `Via` branch tag is generated, but the rest of the SIP request remains the same. Also Timer `B` (or `F`) is reset. This is effectively the same as creating a new transaction for each new target.

If the SIP response to the previously-failed target arrives later, it is ignored because the response does not match the transaction of the currently-tried target, since the `Via` branch tag is not the same.

The Resource Manager supports the configuration of the `DNS SRV` domain name in the outgoing `Record-Route`, `Via`, and `Contact` SIP header by using the `sip.transport.localaddress` configuration option.

File-based Call Recording

Summary and Characteristics

File-based recording utilizes SIP Server's ability to assign active recording to MCP. Some of the specifics are:

- Individual call segments are recorded in the G.711 [`.wav`|`.mp3`] format. These segments contain specific metadata that is written with the recorded files. For example, the metadata can contain information about the segment date and time, the parties involved, and the UUID.
- The recorded files are stored with separate audio channels for customer and agent. This aids speech analytics, because speech engines can determine the active speakers in the conversation. The recorded files do not support audio channel mixing.
- For a recording with a Pause / Resume operation, no blank occurs during the pause; just silence padding.

- When MCP writes the recording file in the .wav format, it uses this template:
  `<dest>/<recording-filename>-<uniqueid>.wav`

  Where:
  - `<dest>` is the recording destination. Configure it in the IVR Profile for the recordingclient service parameter `recdest`. See page 161 and page 163 in "Appendix B: MSML Specification" of this book.
  - `<recording-filename>` is a SIP Server configuration parameter. See the options `request-uri` and `recording-filename` in "Chapter 7: SIP Server Configuration Options" of the *SIP Server 8.1 Deployment Guide*.
  - `<uniqueid>` is a string that MCP automatically generates and appends to recoding-filename, to avoid filename collision.
- When MCP writes the recording file in the mp3 format, it uses this template:
  `<dest>/<recording-filename>-<uniqueid>.mp3`
- When you specify the destination using `file://`, the recording is placed in a storage location.
- To specify the recording format, set the `recordingclient.type` (for the storage recording) and/or `recordingclient.type2` (for analysis) in the IVR profile to:

  `fixed,audio/wav[;codec=[ulaw|alaw]]`
  or
  `fixed,audio/mp3`
- If not specified, the format will be determined by the `[mpc] default_audio_format in a .wav` container.
- To specify `.mp3` as the default format, use [conference] `callrec_default_type`. Example formats: `audio/mp3`.
- MCP can use `dest2` and `type2` (different codecs) as secondary targets for recording storage. MCP supports different file and URI formats for `dest` and `dest2`.
- SIP Server defines `recording-filename` in this format:
  `<recording-filename> = $UUID$_$AGENTDN$_$ANI$_$DNIS$_$DATE$_$TIME$`

  Where:
  - `<recording-filename>` is a SIP Server configuration parameter.
  - `$UUID$`, `$AGENTDN`, `$ANI$`, `$DNIS$`, `$DATE$`, `$TIME$`, `$AGENTID$`, and `CONNID$ are metadata elements that can be` embedded in the `recording-filename` description. The default includes only `$UUID$`.
- Call Recording (also known as Genesys Interaction Recording or GIR) recovers recording sessions following restarts.

Call Recording Encryption

- MCP can now encrypt Amazon S3, http/https, and MSML file-based recordings, using the `PKCS#7` format.
- Configuring a public key path for a tenant enables the encryption functionality.
- If the public key cannot be verified by the MCP, an alarm is raised and no recording is made.
- If no public keys are configured, then the recording is not encrypted.

See the Genesys Interaction Solution Recording Guide for details on encryption and configuring the public key, including authority and chain keys.

Failover Handling

At the call recording's inception, MCP places the recording in a persistent state, to preserve the recording during a failover scenario. The MCP generates a filename for each file-based record of the call and appends a unique identifier after each filename, to ensure uniqueness of the filename in the case of filename collision. If a failover generates multiple files, you can use the timestamp (also known as creation date/time) on the files to determine their correct order before you begin listening.

You can choose where to archive these file-based recordings. Once you have selected a directory, move the files there manually.

Policy-based Tenant Recording Profiles

The GVP Call Recording Solution generates a policy-based recording profile for each tenant. Each profile contains eleven parameters that affect call recordings such as encryption, file format, etc., and are specific to that tenant.

Each parameter can specify one or more variables that configure the profile according to the tenant's preferences.

Use the Genesys Interaction Recording Plug-in for GAX, which is documented in the Genesys Interaction Solution Recording Guide, to modify these parameters.

Table 9 on page 78 lists all gvp:params handled by MCP for all types of recording, including parameters to handle file-based and S3-based recording.

Table 9: gvp:params Handled by MCP for All Recording Types

gvp:param name	Applies to recording type	Description
id	All	Specifies the unique identifier for the recording session if defined. This also represents the filename (no file suffix) when writing to a file, http, and S3 URI. By default SIP Server always includes this parameter in MSML.
dest	All	Specifies the first destination of recorder. The following URI formats are supported • sip: sips: • file: - the directory where the file shall be stored • S3 URI – the S3 bucket where the file is stored. Format: s3:bucketname/bucketpath where: ◦ bucketname is the bucket name in S3, which effectively translates to a virtual hosted-style URI: http://bucketname.s3.amazonaws.com bucketname is also used to create the signature for authorization. ◦ bucketpath (optional) contains the file URL in this format: http://$bucketname.s3.amazonaws.com/$bucketpath/$filename **Note:** The S3 bucket must already exist (or you must create it in Amazon S3) before MCP can upload recording files to it. • http: or https – similar to S3 URI, this is an HTTP URI where MCP can send a PUT to without requiring specific AWS authorization convention.
dest2	All	Specifies the second destination of recorder if defined. The URI formats are supported as per dest.
httpauthorization	http	Value of authorization header when http/https is used for dest.

Table 9: gvp:params Handled by MCP for All Recording Types (Continued)

gvp:param name	Applies to recording type	Description
`httpauthorization2`	http	Value of authorization header when http/https is used for `dest2`. **Note:** Use the format `username:password` to specify the credentials for the parameters `httpauthorization` and `httpauthorization2`. MCP uses an HTTP Basic authentication algorithm to generate the authentication credentials for them.
`AWSAccessKeyId`	S3	Specifies the Amazon AWS Access Key Id for the purpose of generating the authorization header.
`AWSAccessKeyId2`	S3	Same as `AWSAccessKeyId`, but for building the authorization header for `dest2` if `dest2` is an S3 URI.
`AWSSecretAccessKey`	S3	Specifies the Amazon AWS Secret Access Key for the purpose of generating the authorization header.
`AWSSecretAccessKey2`	S3	Same as `AWSSecretAccessKey`, but for building the authorization header for `dest2` if `dest2` is an S3 URI.
`recordDN`	All	Specifies the recording device DN; this parameter gets passed as a request URI parameter to the recording session
`type`	File, http, S3	Specifies the file format (MIME type) to be written by MCP when the dest is not of `sip:` format. Example: audio/wav for .wav file, audio/mp3 for .mp3 file.
`type2`	File, http, S3	Specifies the file format for `dest2` when `dest2` is not of `sip:` format.
`audiosrc`	All	The URI of the audio tone. If the URI is set to empty string, or not defined, or resolves to a bad URI, then no audio tone is applied to the call. No other notifications are generated by the Media Server (ie. MSML events) when no audio tone is being applied.
`tonesilenceduration`	All	Length of time between playing the audio tone in milliseconds. Mandatory if `audiosrc` is defined, otherwise no audio tone is applied.

Table 9: gvp:params Handled by MCP for All Recording Types (Continued)

gvp:param name	Applies to recording type	Description
callrec_dest	http, S3	Defines an http/https: URI for MCP to post the metadata about the call recording after the call recording is completed. MCP uses HTTP POST according to the Recording Processor to attach the call recording URI and call metadata.
callrec_authorization	http, S3	Defines the API authorization key for accessing the Call Recording API. **Note:** MCP uses an HTTP Basic authentication algorithm to generate the authentication credentials for this parameter.

Configuration in GVP

Resource Manager

Configure each tenant on the shared GVP as a separate tenant on Resource Manager. Create a gateway resource for each tenant SIP Server, pointing to the source address of SIP Server.

IVR Profiles

Create an IVR Profile for each tenant. Under each tenant, there should be `gvp.general.default-application` that points to the name of the default IVR Profile.

Service parameters in the default IVR Profile

(under `gvp.service- parameters`)

```
recordingclient.recdest = fixed,file://<directory>
```

Since each MCP instance serves all tenants, the `recdest` parameter should point to the local directory on each MCP instance for each tenant, and this subdirectory must be accessible on each MCP instance. For example,

```
/opt/recording/tenantA (Linux) OR C:\recording\tenantA (Windows)
```

Logical Resource Group

Create a new Logical Resource Group (LRG) and a dedicated group of MCP instances for call recording only. In the folder that contains the MCP instances, the folder should have a section called `gvp.lrg` with the following parameter set:

```
service-types = recordingclient
load-balance-scheme = round-robin
```

`monitor-method = option`

Make sure the other default MCP pool has `service-types` set this way:
`service-types = voicexml;conference;announcement;cpd;media`

GVP supports configuring LRGs to provide exclusive `recordingclient` service, or to provide mixture of services along with `recordingclient`. An LRG that is configured to be dedicated exclusively, by this parameter setting:

`service-types = recordingclient`

...can also be configured to provide other `service-types`, by adding them to the parameter setting. For example:

If you want two MCPs that currently serve `voicexml` and `conference` to serve `recordingclient` also, create an LRG, add two MCP instances, and configure the parameter setting this way:

`service-types = voicexml;conference;recordingclient`

MCP application

To set the codec for recording, set the configuration parameter `mpc.default_audio_format` to `ULAW` or `ALAW`.

Configuration in SIP Server

Most call recording configuration is performed on the SIP Server application, and thus is described in the *SIP Server 8.1 Deployment Guide*.

- Chapter 7: SIP Server Configuration Options describes configuring `request-uri` and `recording-filename`.
- "Feature Configuration," in the section "Call Recording—MSML-based" of Chapter 5: SIP Server Feature Support in the same book, lists and describes all call recording configuration options and requirements.

Chapter

4 Deploying Genesys Media Server

This chapter describes how to deploy Genesys Media Server 8.5 on Windows and Linux operating systems, and provision it to integrate with SIP Server 8.0. It contains the following sections:

Task Summaries

The following Task Summary: Preparing Your Environment contains a list of tasks that are required to prepare your environment for Genesys Media Server and includes links to detailed information that is required to complete these tasks.

Task Summary: Preparing Your Environment

Objective	Related procedures and actions
Plan the deployment	For limitations and recommendations to consider, see Prerequisites and Planning on page 23.

Task Summary: Preparing Your Environment (Continued)

Objective	Related procedures and actions
Prepare your environment—Install common Genesys Framework components	1. Management Framework. Deploy Genesys Management Framework, and ensure that it is fully operational and running. See the *Framework 8.0 Deployment Guide.* Management Framework is the centralized element-management system for all Genesys software.
	2. Genesys Administrator. Install Genesys Administrator, and ensure that it is fully operational. See the *Framework 8.0 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 host. After the SNMP Master Agent has been installed on the host, assign the SNMP Master Agent to Media Server to capture alarm and trap information. See Procedure: Creating a Connection to a Server, on page 114. The Genesys Voice Platform 8.5 CD 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 select the default installation path:, `C:\Program Files\GCTI\gvp\VP MIB 8.5` **Note:** The SNMP Master Agent is required only if you are capturing alarm and trap information. For more information about the MIBs, see the *Genesys Voice Platform 8.5 SNMP and MIB Reference.*

The Task Summary: Installing Genesys Media Server and Resource Manager, on page 85 contains a list of tasks that are required to install Media Server and the Resource Manager, and includes links to detailed information that is required to complete these tasks.

Task Summary: Installing Genesys Media Server and Resource Manager

Objective	Related procedures and actions
Prepare the hosts	1. Stop any antivirus software that might be running on systems that will host Media Server or Resource Manager. Check the vendor documentation for your antivirus software configuration.
	2. Install the Local Control Agent on the Media Server and if required, Resource Manager hosts. See Procedure: Installing the Local Control Agent (Windows), on page 86 or Procedure: Installing the Local Control Agent (Linux), on page 87.
Configure the hosts	• Configure a new host in the Configuration Database for the Media Server and, if required, Resource Manager. See Procedure: Configuring a Host in Genesys Administrator, on page 89.
Install the components	1. Create the `Application` objects: a. Import the templates. See Procedure: Importing Application Object Templates Manually, on page 93. b. Create the `Application` objects. See Procedure: Creating Application Objects Manually, on page 95.
	2. Install the Media Server. See Procedure: Installing Media Server (Windows), on page 98.
	3. Install the Resource Manager. If you are planning to provide load balancing for Media Server, install the Resource Manager. See Procedure: Installing the Resource Manager (Windows), on page 102 or Procedure: Installing the Resource Manager (Linux), on page 104.
Start the components	• Start the Media Server and, if required, Resource Manager components manually (or configure the components to start automatically). See Procedure: Configuring Application Objects to Start Automatically, on page 105.

Preparing the Host

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, the Genesys Local Control Agent (LCA) must be installed on the Media Server host (and Resource Manager host, if required).

Note: The Resource Manager can be deployed as a High Availability (HA) pair. For procedures to deploy the Resource Manager in this mode, see Appendix E in the *Genesys Voice Platform 8.5 Deployment Guide.*

In addition, a new host is created for Media Server (and for the Resource Manager, if required) in the Configuration Database by using Genesys Administrator so that the Configuration Server can detect its 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:

- Installing the Local Control Agent (Windows)
- Installing the Local Control Agent (Linux) on page 87
- Configuring a Host in Genesys Administrator on page 89

Procedure: Installing the Local Control Agent (Windows)

Purpose: To install and configure the Local Control Agent on a Windows host.

Summary

Install the LCA on the Media Server host to ensure that it is controlled and monitored by the Solution Control Server. When you install the LCA, the Genesys Deployment Agent (GDA) is also installed.

Prerequisites

- The server on which you are installing Media Server meets the system requirements. For more information about these requirements, see Chapter 2 on page 23.
- The fully qualified domain names (FQDN) of Genesys servers do not contain special characters, such as the underscore (`_`) .

  Note: To ensure that Genesys software works properly, FQDNs must 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 Media Server will be installed.

- 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 the *Framework 8.0 Deployment Guide*.

Start of procedure

1. On the host, navigate to the directory that contains the installation files for the Local Control Agent and then execute the `setup.exe` file.
2. At the prompt, enter the information that identifies the host, as shown in Table 10.

Table 10: Configuration Server Properties

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`.

3. Click `Next`.
4. Restart the host computer.
5. 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.

End of procedure

Next Steps

Configure the GVP hosts in the Configuration Database. See Procedure: Configuring a Host in Genesys Administrator, on page 89.

Procedure: Installing the Local Control Agent (Linux)

Purpose: To install and configure the Local Control Agent on a Linux host.

Summary

Install the LCA on the Media Server host to ensure that it is controlled and monitored by the Solution Control Server. When you install the LCA, the Genesys Deployment Agency is also installed.

Prerequisites

- The server on which you are installing the Media Server meets the system requirements. For more information about these requirements, see Chapter 2 on page 23.
- The fully qualified domain names of Genesys servers do not contain special characters, such as the underscore (`_`).

> **Note:** To ensure that Genesys software works properly, FQDNs must 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 Media Server will be installed.
- You have obtained the Genesys Management Framework CDs, or a network path and the location the LCA software. For the directory structure of the Installation CDs, see the *Framework 8.0 Reference Guide.*

Start of procedure

1. At the Linux host, log in as root by typing `su`.
2. Log in as root and enter the path to the directory that contains the LCA installation package.
3. Run the `sh install.sh` command.

   The installation script is initiated.
4. At the prompt, enter the information that identifies the host, as shown in Table 10 on page 87.
5. At the prompt, enter the destination directory—for example: `/opt/genesys/lca`

> **Note:** Genesys recommends that you use the destination directory that is shown in the example.

6. Configure the GDA to start automatically when the server is restarted—for example,
   `/etc/rc.local /etc/rc.d/init.d/gctigda start,` and press `Enter`.

7. Configure the LCA to start automatically when the server is restarted—for example,
   `/etc/rc.local /etc/rc.d/init.d/gctilca start`, and press `Enter`.

   Alternatively, you can start the LCA and GDA manually at the Linux prompt by using the following commands:

   `/etc/init.d/gctilca start` and `/etc/init.d/gctigda start`

End of procedure

Next Steps

- Configure the GVP hosts in the Configuration Database. See Procedure: Configuring a Host in Genesys Administrator.

Procedure: Configuring a Host in Genesys Administrator

Purpose: To configure a host in Genesys Administrator to communicate with the Configuration Server.

Summary

Each new host is configured in Genesys Administrator and is controlled and monitored by the LCA.

Prerequisites

- The Genesys Administrator web application is installed on the Management Framework host.
- You have obtained the Universal Resource Locator (URL) of Genesys Administrator.

Start of procedure

1. In a web browser, type the URL to Genesys Administrator—for example:
   `http://<Genesys Administrator host>/wcm/`
2. In the `Login` dialog box, enter the information, as shown in Table 11.

Table 11: Genesys Administrator Login

Field	Description
User Name	Enter the user name, typically `default`
Password	Enter the password, typically `password`.

Table 11: Genesys Administrator Login (Continued)

Field	Description
Application	Enter the application name of the Configuration Server, typically default.
Host Name	Enter the host name of the Configuration Server—for example, CnfgS1.
Port	Enter the port number of the Configuration Server, typically 2020.

3. Click OK.

   The Genesys Administrator graphical user interface (GUI) is displayed.
4. On the Provisioning tab, click Environment > Hosts> New.
5. In the General section of the Configuration tab, enter the information that identifies the host, as shown in Table 12.

Note: 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 will fail when the hostname command is executed.

Table 12: Host Properties

Field	Description
Name:	Enter the host name of the Media Server (Media Control Platform) host—for example, MCP1
IP Address:	Enter the IP address of the 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).
State:	Enter a check mark in Enabled.

6. Save the configuration.

End of procedure

Next Steps

- Complete the preinstallation activities. See "Preinstallation Activities".

Preinstallation Activities

Before you begin the preinstallation activities, ensure that the Local Control Agent (LCA) is installed on the Media Server host and that it is configured in the Configuration Database. See "Preparing the Host" on page 85.

To install the Media Server create an `Application` object in the Configuration Database and import the Media Server object template from the installation CD or from a shared network directory. After the template is imported, you can use it to install subsequent instances of the same component. For example, if you are installing more than one Media Server instance, you can use the same template for each Media Server `Application` object.

Note: As a best practice, whenever you are using these manual procedures to install `Applications`, import all of the `Application` object templates that you require before you begin to deploy the components.

You can find the Media Server object template and metadata files in the `<Genesys Solutions Dir>\Templates\` directory on the CD. The object template and metadata file names are:

- `VP_MediaControlPlatform_81x.apd`
- `VP_MediaControlPlatform_81x.xml`

Creating Application Objects

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 New Application Wizard
- Procedure: Importing Application Object Templates Manually, on page 93
- Procedure: Creating Application Objects Manually, on page 95

Procedure:
Using the New Application Wizard

Purpose: To create Application objects in the Configuration Database for each component.

Summary

The New Application Wizard in Genesys Administrator imports the Application object templates and creates the Application objects for you.

Prerequisites

- The Installation Packages are accessible from the CD or from a shared network directory.

Start of procedure

1. Log in to Genesys Administrator.
2. On the Provisioning tab, click Environment > Applications.
3. In the Task pane, select Create Application.

   The Create New Application Wizard appears.
4. Click Browse for File to import a template.

   Notes: If the templates were previously imported, you can use an existing template by selecting Browse for Template.
5. Click Add to navigate to the directory that contains the template (.apd) files.

   Figure 4 on page 94 shows the Add dialog box.
6. Click Next to specify the metadata.
7. Click Browse > Add to import the metadata for the Application object your are creating.
8. Click Next to configure the application parameters.
9. In the Host field, click the Browse icon to select the host on which you want to install the application.

   Note: 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.
10. After the host appears on the Application Parameters page, click Create.

    The Results page appears, to confirm the Application object is created.

11. Click Finish.

End of procedure

Next Steps

- Install the Media Server. See "Installing Media Server (Windows)" on page 98 or "Installing Media Server (Linux)" on page 100.

Procedure:
Importing Application Object Templates Manually

Purpose: To import an Application object template to the Configuration Database manually before you install the Application object.

Summary

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 the Applications, you can omit this procedure, because the wizard imports the component Application object template and creates the Application object for you.

Prerequisites

- The Media Server host is prepared for deployment. See "Preparing the Host" on page 85.

Start of procedure

1. Log in to Genesys Administrator.
2. On the Provisioning tab, click Environment > Application Templates.

   A Waiting.. dialog appears.
3. In the Tasks pane, click Upload Template.
4. Click Add.

   Figure 4 on page 94 is an example of the dialog box to add the .apd template file:

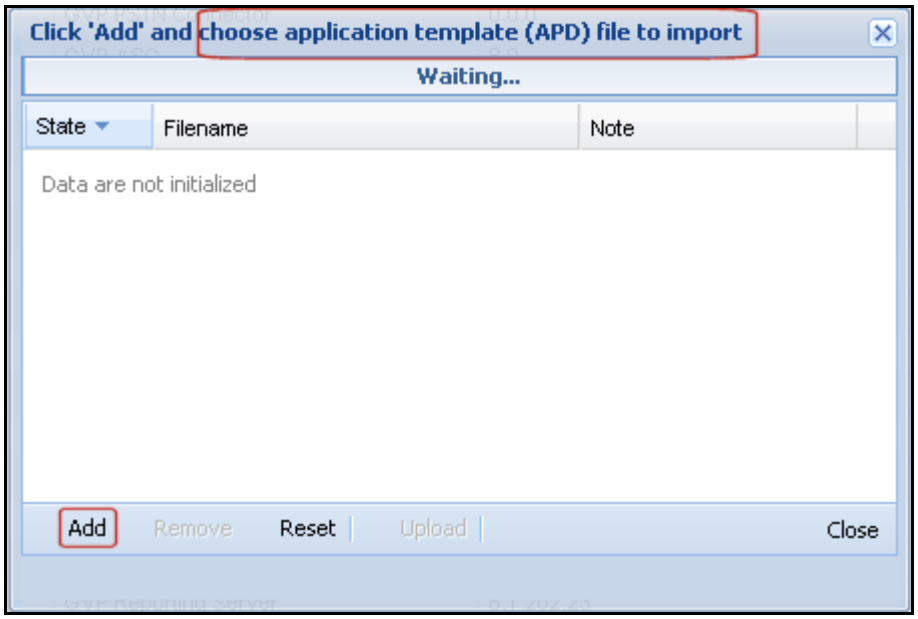

Figure 4: Importing the .apd Template

5. In the `Choose File` dialog box, navigate to the directory that contains the Media Control Platform `Application` object templates.
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.
7. Click `Import Metadata`, as shown in Figure 5:

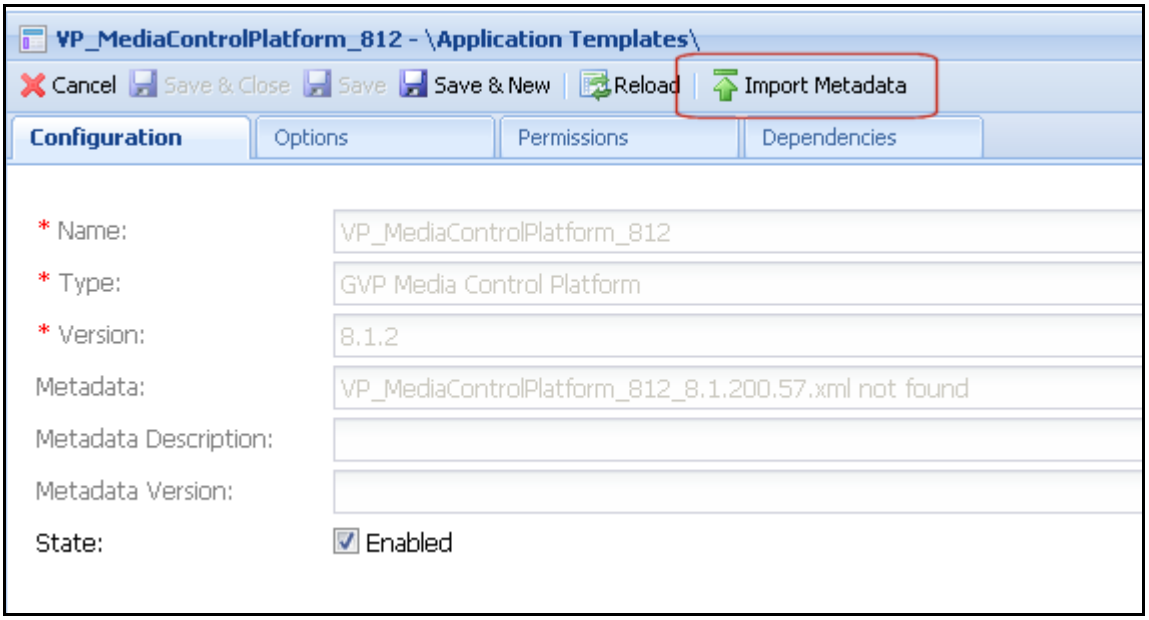

Figure 5: Import Metadata

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 Table 13.

Table 13: Application Template Properties

Field	Description
Name:	Enter a descriptive name for the template—for example, GVP_MCP_template.
Type:	From the drop-down list, select the template type: • For the Media Control Platform Application object select the template that has the same name—for example, Media Server.
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 check box to indicate Enabled.

12. Click Save.

End of procedure

Next Steps

- Create the Media Control Platform Application object in the Configuration Database. See Procedure: Creating Application Objects Manually.

Procedure: Creating Application Objects Manually

Purpose: To create an Application object manually in the Configuration Database for the application that you are installing.

Summary

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 the Media

Control Platform, you can omit this procedure, because the Wizard imports the component Application object template and creates the Application object for you.

Prerequisites

- An Application object template is imported for the type of object that you are installing. See Procedure: Importing Application Object Templates Manually, on page 93.

Start of procedure

1. Log in to Genesys Administrator.
2. On the Provisioning tab, select Environment > Applications > New.

   The Browse..\Application Templates\ dialog box appears, displaying the contents of the Application Templates directory. See Figure 6.

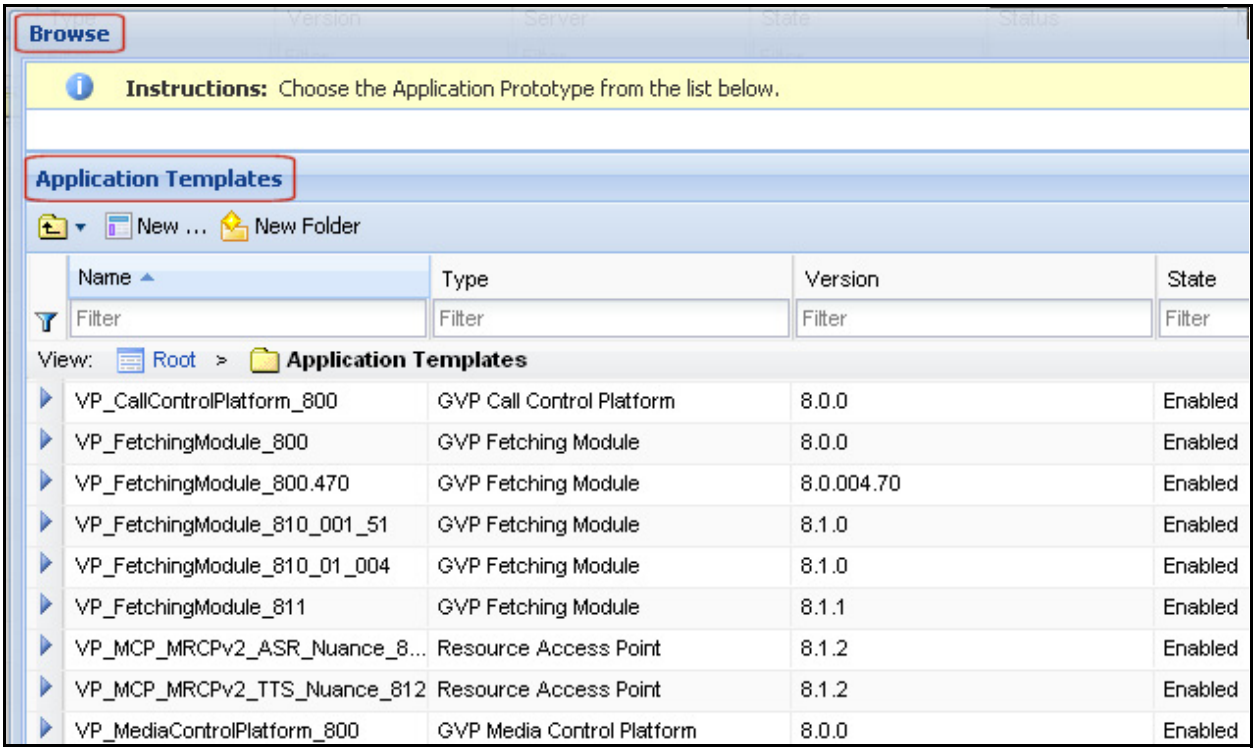

Figure 6: Browse Application Templates

3. Click the object template for the Media Control Platform Application object.

   The Configuration tab appears, with some of the fields in the General section populated and disabled.
4. In the Name field, enter the name of the application—for example, Media_Server.

5. In the `State` field, retain the default value: `Enabled`.
6. In the `Server Info` section, enter the information as shown in Table 14.

Note: Table 14 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 14: 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, \. 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`. This value must be `1` or higher and it makes sense only if you specify a backup server for this server.

Table 14: Application Object Properties (Continued)

Field	Description
Auto Restart	From the drop-down list, select `True` or `False`. The default value is `False`. Selecting `True` causes the User Interaction Layer to automatically restart the application after it fails. Selecting `False` prevents the User Interaction Layer from automatically restarting the application after it fails. **Note:** Genesys recommends that you select `True` for this parameter.

7. Click `Save`.

End of procedure

Next Steps

- Install the Media Server. See Procedure: Installing Media Server (Windows) or Procedure: Installing Media Server (Linux), on page 100.

Installing Media Server

This section describes how to install Media Server (by installing the Media Control Platform) on Windows and Linux in a new deployment, or add Media Server to an existing deployment.

Before you begin to install the component, copy the Media Control Platform installation package to a directory on the host or to a network drive from which it can be downloaded.

Note: You can install multiple instances of the Media Control Platform (Media Server) on a single host. For more information about installing multiple instances of the Media Control Platform, see the *Genesys Voice Platform 8.5 Deployment Guide*.

This section contains the following procedures:

- Procedure: Installing Media Server (Windows)
- Procedure: Installing Media Server (Linux), on page 100

Procedure: Installing Media Server (Windows)

Purpose: To install the Media Control Platform (Media Server) component, so that Session Initiation Protocol (SIP) applications can access media services.

Prerequisites

- The Media Server host is prepared for installation. See "Preparing the Host" on page 85.
- The Media Control Platform `Application` object template is imported, and an `Application` object is created. See "Preinstallation Activities" on page 91.

Start of procedure

1. Execute the `setup.exe` setup file:
   - If you are using the software CD, browse to the `<Installation_CD>\solution_specific\windows\mcp\` folder.
   - If the CD image is on a network drive, copy the `<CDImage>\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 Table 15.

   These are the connection parameters for the Configuration Server.

Table 15: 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. If you checked the first option in Step 8:
   - 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.

   Note: If you did not check the first option in Step 8, you can skip Step 9.
10. On the Ready to Install page, click Install.

End of procedure

Next Steps

- Configure the Media Control Platform Application object to start automatically. See Procedure: Configuring Application Objects to Start Automatically, on page 105.
- Create a connection to the Reporting Server (Optional). See Procedure: Creating a Connection to a Server, on page 114.

Procedure: Installing Media Server (Linux)

Purpose: To install the Media Control Platform (Media Server) component, so that Session Initiation Protocol (SIP) applications can access the Media Control Platform media services.

Prerequisites

- The Media Server host is prepared for installation. See “Preparing the Host” on page 85.
- The Media Control Platform Application object template is imported, and an Application object is created. See “Preinstallation Activities” on page 91.

Start of procedure

1. At the Linux host, log in as root, and then type su.

2. Navigate to the directory that contains the Media Control Platform installation package.
3. Type `chmod a+x install.sh`, and then press `Enter`.
4. Run the `./install.sh` command.

   The installation script is initiated.
5. At the prompt, enter the hostname of Media Control Platform—for example:

   ```
   Please enter the host name or press enter for "<local_host>"
   =><local_host>.
   ```
6. 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
   ```
7. 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
   ```
8. At the prompt, choose the application that you want to install—for example:

   ```
   1 : MCP-Host
   2 : MCP_8.5.200.09
   3 : MCP_8.5.200.19
   =>3
   ```
9. Add At the prompt, enter the host name and IP address of the proxy server host—for example:
   - `Proxy Serv HostName = <Prxy_Serv_Name>`
   - `Proxy Serv IP = <Prxy_Serv_IP>`
10. At the prompt, choose the audio format for your region—for example:
    - `Mulaw (North America)`
    - `Alaw (Europe)`
11. At the prompt, enter the path to the directory in which the application files will reside—for example:

    ```
    Press ENTER to confirm /<Install_Dir>/gvp81/MCP_8.5.200.xx as the
    destination directory or enter a new one =>
    /opt/genesys/gvp/VP_Media_Control_Platform_8.5.200.xx
    ```

    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.

End of procedure

Next Steps

- Configure the Media Control Platform `Application` object to start automatically. See Procedure: Configuring Application Objects to Start Automatically, on page 105.

Note: To start any `Application` object manually on a Linux host, type `<Install_Dir>/bin/run.sh`, and press `Enter`, where `<Install_Dir>` is the directory in which the application is installed.

Installing Resource Manager

If you have deployed multiple Media Servers in your environment, you must install the Resource Manager to provide load balancing.

Procedure: Installing the Resource Manager (Windows)

Purpose: To install the Resource Manager on the host.

Prerequisites

- The Resource Manager host is prepared for installation. See "Preparing the Host" on page 85.
- The Resource Manager `Application` object template is imported and an `Application` object is created. See "Preinstallation Activities" on page 91.

Start of procedure

1. Execute the `setup.exe` setup file:
   - If you are using the GMS software CDs, browse to the `<GMS_Installation_CD>\solution_specific\windows\rm\` folder.
   - If the CD image is on a network drive, copy the `<CDImage>\solution_specific\windows\rm\` folder to the local computer.
2. When the Genesys Deployment Wizard appears, click `Next`.

   On the `Connection Parameters` page, enter the information in the `Host` and `User` sections, as shown in Table 16 on page 103.

Table 16: 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.

These are the connection parameters for the Configuration Server.

3. On the Client Side Port Configuration page, select Use Client Side Port (if required). Enter the Port and IP Address.
4. On the Select Application page, select the Resource Manager Application object.
5. 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.
6. In the VP Reporting Server section, enter the information, as shown in Table 17.

Note: Step 6 is only required if you have deployed VP Reporting Server. For more information about deploying VP Reporting Server, see the *Genesys Voice Platform 8.5 Deployment Guide*.

Table 17: VP Reporting Server Section

Field	Description
Host	Enter the host name of the Reporting Server—for example, ReportServ1.
Port	Accept the default value, 61616, for the Reporting Server port number.

7. On the Ready to Install page, click Install.
8. When the installation is complete, click Finish.

End of procedure

Next Steps

- Configure the Resource Manager `Application` object to start automatically. See Procedure: Configuring Application Objects to Start Automatically, on page 105.

Procedure: Installing the Resource Manager (Linux)

Purpose: To install the Resource Manager component on a host.

Prerequisites

- The Resource Manager host is prepared for the installation of GVP components. See “Preparing the Host” on page 85.
- The Resource Manager `Application` object template is imported, and an `Application` object is created. See “Preinstallation Activities” on page 91.

Start of procedure

1. At the Linux host, log in as root, and then type `su`.
2. Navigate to the directory that contains the Resource Manager installation package.
3. Type `chmod a+x install.sh`, and then press `Enter`.
4. Run the `./install.sh` command.

   The installation script is initiated.
5. 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>.
   ```
6. 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
   ```
7. 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
   ```

8. At the prompt, choose the application that you want to install—for example:

```
1 : RM-Host
2 : RM_8.5.000.09
3 : RM_8.5.000.19
=>3
```

9. At the prompt, enter the path to the directory in which the application files will reside—for example:

```
Press ENTER to confirm /<Install_Dir>/gvp81/RM_8.5.000.xx as the
destination directory or enter a new one =>
/opt/genesys/gvp/VP_Resource_Manager_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 that indicates that the installation was completed successfully.

End of procedure

Next Steps

- Configure the Resource Manager `Application` object to start automatically. See Procedure: Configuring Application Objects to Start Automatically.

Procedure: Configuring Application Objects to Start Automatically

Purpose: To configure the GVP `Application` objects to start automatically after the installation.

Summary

This procedures explains how to configure the components to start automatically in two different ways.

Start of procedure

1. Log in to Genesys Administrator.
2. On the `Provisioning` tab, select `Environment > Applications`.
3. Double-click the `Application` object that you want to configure to start automatically.

   The `Configuration` tab appears.
4. Configure the `Application` in one of two ways:
   a. In the `Server Info` section:
      — Scroll down to the `Auto Restart` field.

— Click the `True` check box to enable it.

b. On the `Options` tab, from the `View` drop-down menu:

— Select `Advanced View (Annex)`.

— In the `sml` section, select `New`.

The `New Option` dialog box appears.

— In the `Name` field, enter `autostart`.

— In the `Value` field, enter `true`.

5. Save the changes.

End of procedure

Next Steps

- No further steps are required.

Installing Reporting Server

The Reporting Server Installation Package (IP) was moved to the Genesys Media Server distribution disk with GVP release 8.1.6. The installation procedures (Windows and Linux) are now in this book, but also remain at their original location in the *Genesys Voice Platform 8.5 Deployment Guide*.

Procedure: Installing the Reporting Server (Windows)

Purpose: To install and provision the Reporting Server on a Windows host.

Summary

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.

Prerequisites

- The Sun Java Runtime Environment (JRE) 6.0, Update 19 is installed. See “Prerequisites” on page 24.

  Note: JRE 7.0 or later is required if you are using IPv6 communications.

- The Reporting Server host is prepared for the installation. See "Preparing the Host" on page 85.
- The Reporting Server `Application` object template is imported, and an `Application` object is created. See "Preinstallation Activities" on page 91.

Start of procedure

1. 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.
2. When the Genesys Deployment Wizard appears, click `Next`.
3. On the `Connection Parameters` page, enter the information in the `Host` and `User` sections, as shown in Table 15 on page 99.

   These are the connection parameters for the Configuration Server.
4. On the `Select Application` page, select the Reporting Server `Application` object.
5. Select the destination folder in one of two ways:
   - Click `Next` to accept the default directory
6. 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 described in Table 18 on page 108.

Notes:
- In Table 18, 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.
- If you are installing an Oracle database, enter the SID or *global database name* in the `Database Name` field.

Table 18: 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`.

End of procedure

Next Steps

- Configure the Reporting Server `Application` object to start automatically. See "Configuring Application Objects to Start Automatically" on page 105.

Procedure: Installing the Reporting Server (Linux)

Purpose: To install and provision the Reporting Server on the host.

Summary

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.

Prerequisites

- The Sun Java Runtime Environment (JRE) 6.0, Update 19 is installed. See the Procedure: Preparing Your Environment, on page 83.

  Note: JRE 7.0 or later is required if you are using IPv6 communications.

- The Reporting Server host is prepared for installation. See “Preparing the Host” on page 85.
- The `Reporting Server Application` object template is imported, and an `Application` object is created. See “Preinstallation Activities” on page 91.

Start of procedure

1. At the Linux host, log in as root, and then type `su`.
2. Navigate to the directory that contains the Reporting Server installation package.
3. Complete Step 3 to Step 6 in Procedure: Installing Media Server (Linux), on page 100, substituting information for the Reporting Server, where necessary.
4. 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
   ```

5. At the prompt, enter the number associated with the database server you want to select—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
   ```

Notes: GVP supports only Oracle 10g or 11g Database Servers on Linux.

6. 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 =>
```

7. 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 =>
```

8. 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 =>
```

Note: When you are installing an Oracle database, enter the SID or *global database name* in the `Database Name` field.

9. 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 =>
```

10. At the prompt, type `password`, and then press `Enter`—for example:

```
Please specify the Database Server user password =>password
```

11. 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 =>
```

12. 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 =>
```

13. 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
```

Note: 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.

End of procedure

Next Steps

- Configure the `Reporting Server Application` object to start automatically. See "Configuring Application Objects to Start Automatically" on page 105.

Provisioning Media Server

The following Task Summary: Provisioning the Media Server summarizes the tasks that are required to configure Media Server for the functionality that you want to use in your deployment and provides links to detailed information that is required to complete these tasks.

Task Summary: Provisioning the Media Server

Objective	Related procedures and actions
Integrate with Resource Manager.	• See Procedure: Integrating Media Server with the Resource Manager, on page 112. (If you have deployed only one instance of Media Server without the Resource Manager, this task is not required.)
Create server connections.	• Create connections to the servers with which Media Server will communicate. See Procedure: Creating a Connection to a Server, on page 114.
Create a Logical Resource Group	• Create the Resource Group. See Procedure: Creating a Resource Group, on page 116. (This task is required if you have deployed multiple Media Server instances and you are using the Resource Manager.)

Task Summary: Provisioning the Media Server (Continued)

Objective	Related procedures and actions
Create and configure a default IVR Profile	1. Use the IVR Profile Wizard to create a default profile. See Procedure: Creating a Default Profile, on page 118. 2. If you are configuring the default IVR Profile for a specific tenant: a. Go to `Provisioning > Environment > Tenants`. b. Double-click the tenant that you want to configure. c. On the `Options` tab, create a section named, `gvp.general`. d. In the `gvp.general` section, configure the following options: • `default-application=<Default_IVRProfile>` Where `Default_IVRProfile` is the name of the default profile you created in Step 1. See also, "Configuring an IVR Profile for Media Server/Cisco UCM Integration" on page 119.
Integrate Media Server with SIP Server	• See Integrating with SIP Server on page 119.

Integrating with Resource Manager

After the Media Control Platform `Application` object is created and the component installed, it is integrated with the Resource Manager, which acts as a proxy server. SIP devices can then make use of media-centric services through the proxy, without having to know the actual location of these resources.

This procedure is required if you have installed multiple instances of Media Server and want the Resource Manager to act as a proxy server and require load-balancing functionality. To integrate the Media Server `Application` object with the Resource Manager, you configure the Session Initiation Protocol (SIP) and secure SIP options.

Procedure: Integrating Media Server with the Resource Manager

Purpose: To integrate the Media Server (Media Control Platform) with the Resource Manager by configuring the `Application` parameters.

Prerequisites

- The Media Control Platform is installed. See Procedure: Installing Media Server (Windows), on page 98 or Procedure: Installing Media Server (Linux), on page 100.
- The Resource Manager is installed. See Task Summary: Preparing Your Environment, on page 83.

Start of procedure

1. Log in to Genesys Administrator.
2. On the `Provisioning` tab, select `Environment > Applications`.
3. Click the `Application` object that you want to configure—for example, the Media Control Platform `Application` object.

   The `Configuration` tab appears.
4. Click the `Options` tab, and use the `View` drop-down list to select `Show options in groups...`
5. If `Remdial` is used, select `sip` to find the `routeset` option.
6. In the `Value` field, type the following:
   - `<sip:IP_RM:SIPPort_RM;lr>`

   Where `IP_RM` is the IP address of the Resource Manager, and `SIPPort_RM` is the SIP port of the Resource Manager—typically, `5060`.

   Note: You must include the angle brackets in the `Value` field in the `sip.routeset` and `sip.secureroutese`t parameters.

7. In the `Value` field of the `secureroutese`t option, type the following:
   - `<sip:IP_RM:SIPSecurePort_RM;lr>`

   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`.
8. 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 Steps 6 and 7:
   - `sip.routeset`
   - `sip.secureroutese`t
9. Save the configuration.

End of procedure

Next Steps

- Create the connections to the Message Server. See "Connecting to a Server".

Connecting to a Server

Use the procedure in this section to create Media Server connections. Table 19 describes the connections that enable Media Server to send data to other servers for various purposes.

Table 19: Media Server Connections

Server or Component	Description
Message Server	To ensure that component log information reaches the Log database and can be viewed in the Solution Control Interface (SCI)
Reporting Server	To ensure that these components detect the Reporting Server to which they are sending reporting data. (Optional)
SNMP Master Agent	To ensure that alarm and trap information is captured.

Note: A single SNMP Master Agent can serve a single component only. Therefore, you must have an SNMP Master Agent installed and a connection configured for each Media Server instance.

Procedure: Creating a Connection to a Server

Purpose: To create a connection in an `Application` object to a server or component.

Prerequisites

- The components for which you are creating connections are installed.

Start of procedure

1. Log in to Genesys Administrator.
2. On the `Provisioning` tab, select `Environment > Applications`.

3. Click the `Application` object for which you are creating the connection—for example, the Media Control Platform `Application` object.

   The `Configuration` tab appears.

4. In the `General` section, in the `Connections` field, click `Add`.

   The `Connection Info` dialog box appears. See Figure 7.

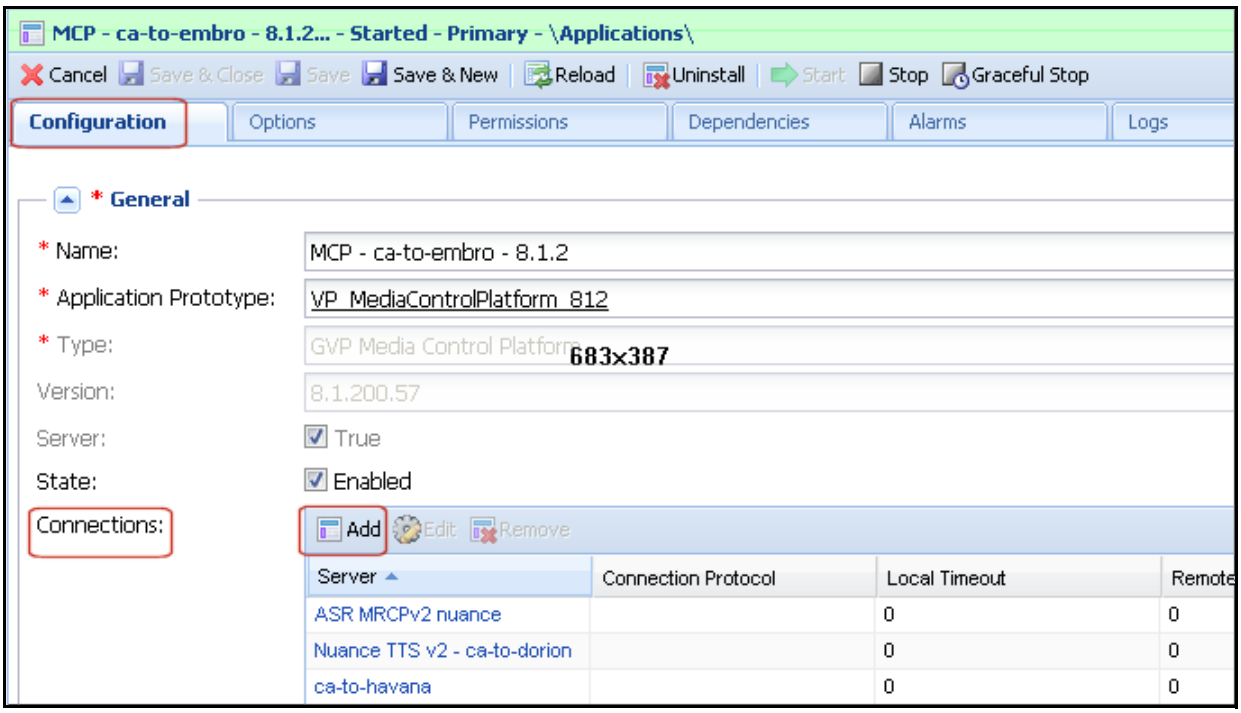

Figure 7: Application Object—Connection Info

5. In the `Server` field, click the down arrow to open the `Browse Application` dialog box.
6. Select the server or component to which you want to create a connection—for example, `Message Server` or `SNMP Master Agent`.

   The required fields in the `Connection Info` section are populated automatically.

7. Click `OK`.

   The server or component you selected in Step 6 appears under `Connections`.

8. Save the configuration.

End of procedure

Next Steps

- Create a Logical Resource Group. See Procedure: Creating a Resource Group, on page 116.

Procedure:
Creating a Resource Group

Purpose: To group resources that use common services to facilitate load balancing.

Summary

The `MCPGroup` that is created in this procedure enables the Resource Manager to easily perform load balancing for the resources within the group. This procedure is only required if you have deployed multiple Media Server instances and are using the Resource Manager.

Prerequisites

- The Resource Manager is installed. See Task Summary: Preparing Your Environment, on page 83.

Start of procedure

1. Log in to Genesys Administrator.
2. On the `Provisioning` tab, click `Voice Platform > Resource Groups`.
3. On the `Details` pane tool bar, click `New`.

   The Resource Group Wizard opens to the `Welcome` page.
4. On the `Resource Manager Selection` page, add the Resource Manager `Application` object for which you want to create the group. On the `Group Name and Type` page:
   a. Enter `MCPGroup`.
   b. Select `Media Control Platform`.

   Note: When the Media Control Platform is selected, an additional drop-down menu appears, enabling you to select or deselect the service types, such as VoiceXML, Conference Announcement. All service types are selected by default.
5. On the `Tenant Assignments` page, add the child tenant to which the Resource Group will be assigned.

   Note: This step is required only if you are creating the Resource Group in a multi-tenant environment.
6. On the `Group Properties` page, enter the information from Table 20 on page 117 for the Resource Group that you are configuring.

Note: 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 Table 20. For a complete list of resource-group options and their descriptions, see the *Genesys Voice Platform User's Guide*.

Table 20: Group Properties—Resource Group Wizard

Field name	Value
Monitoring Method	Retain the default value: `SIP OPTIONS`.
Load Balance Scheme	Select `round-robin`.
Maximum Conference Size	Enter `-1`.

7. On the `Resource Assignment` page:
   a. Select the check box 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.
   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.
8. On the `Confirmation` page, click `Finish`.

End of procedure

Next Steps

- Create a default IVR Profile. See Procedure: Creating a Default Profile, on page 118.

Procedure:
Creating a Default Profile

Purpose: To create a default IVR Profile that can be used to accept calls other than those specified in the dialing plans.

Start of procedure

Prerequisites

- There are no prerequisites for this procedure.

Start of procedure

1. Log in to Genesys Administrator.
2. On the `Provisioning` tab, select `Voice Platform > IVR Profiles`.
3. In the `Tenant:` field at the top of the GUI, browse to select the tenant to which you want this IVR Profile to belong.
4. In the `Tasks` panel, click `Define New IVR Profile`.

   The `IVR Profile Wizard` opens to the `Welcome` page.
5. On the `Service Type` page:
   a. Enter the name of the default IVR Profile, `IVR_App_Default`.
   b. Select either `Conference` or `Announcement` from the drop-down list.

   (Only one service type per IVR Profile is supported.)
6. If you selected `Conference`, on the `Service Properties` page, enter a conference ID number.
7. If you selected `Announcement`, on the `Service Properties` page, enter the URL of the announcement, for example, `http://webserver/hello.wav`.
8. Click `Finish`.

   Note: When you use the `IVR Profile Wizard` to create the default profile, the `gvp.general` and `gvp.service-prerequisites` sections are created for you and include the required parameters.
9. In the `gvp.general` section of the Tenant's Annex tab, set the `default-application` to this IVR Profile.

End of procedure

Next Steps

- Complete the procedures to integrate Media Server with SIP Server. See "Integrating with SIP Server" on page 119.

Configuring an IVR Profile for Media Server/Cisco UCM Integration

When Genesys Media Server is integrated with Cisco UCM, it can acts as a recorder device. When a SIP `INVITE` messages is sent to Media Server by the Resource Manager, the header format differs from the format at the far end of the communication path (Cisco UCM). However, you can synchronize these formats, by using the `cisco-record-file` configuration parameter in the IVR Profile Wizard, which specifies the file name pattern that will be used.

If you want to configure this option manually (without using the wizard), you can go to `Annex` of the Resource Manager `Application` and add the option to the `gvp.general` section.

Integrating with SIP Server

SIP Server integrates with the Media Server using a Voice over IP Service DN with service-type set to msml. Only one DN is required for all media services (except for recording, which requires one Voice over IP Service DN with service-type set to msml, and another with service-type set to recorder).SIP Server does not communicate directly with the Media Server (MCP), but instead sends the msml service requests to Resource Manager, which then selects and manages the MCP independently from SIP Server. This allows for efficiencies in scalability and redundancy.

Integrating with SIP Server Indirectly

The following Task Summary: Integrating Media Server with SIP Server Indirectly summarizes the tasks that are required to integrate Media Server with SIP Server through the Resource Manager and provides links to detailed information that is required to complete these tasks.

Task Summary: Integrating Media Server with SIP Server Indirectly

Objectives	Related procedures and actions
Enable MSML services on SIP Server	Configure the SIP Server `Application`: 1. In Genesys Administrator, go to `Provisioning > Environment`. 2. Select the SIP Server `Application` that you want to configure. 3. On the `Options` tab, in the `TServer` section, configure the `msml-support` option with a value of `true`.

Task Summary: Integrating Media Server with SIP Server Indirectly (Continued)

Objectives	Related procedures and actions
Configure a DN for VoIP service	Configure the DN: 1. In Genesys Administrator, go to `Provisioning > Switching > Switches` and select the SIP switch that you want to configure. 2. Create a new DN of type `VoIP Service`. 3. On the `Options` tab, in the `TServer` section, configure the following options: • `contact=sip:<RM_IP_Address>:<RM_SIP_port>` Mandatory. Points to the Resource Manager IP address and port. Specifies the contact URI that SIP Server uses for communication with the treatment server: • `service-type=msml` • `subscription-id=<String>` Configure this option value to the name of the tenant to which the SIP Server switch belongs. For single-tenant environments, set this option to `Resources`. For multi-tenant environments, set this option to `Environment` (default) or to specific tenant name. The value can be any string. • `prefix=msml=` Configure this option for Conferencing and Monitoring features only. (Optional.)
Configure a DN to support Media Server as the emergency recording server.	Enable SIP Server to play emergency recording streams. 1. Go to `Provisioning > Switching > Switches` and select the SIP switch that you want to configure. 2. Create a new DN of type Trunk and name it `gcti::record` There are no options to configure for this DN.

Task Summary: Integrating Media Server with SIP Server Indirectly (Continued)

Objectives	Related procedures and actions
Configure a DN to support Media Server as the recording server.	Configure the DN: 1. Go to `Provisioning > Switching > Switches` and select the SIP switch that you want to configure. 2. Create a new DN of type `VoIP Service`. 3. On the `Options` tab, in the `TServer` section, configure the following options: • `contact-type=sip:<RM_IP_Address>:<RM_SIP_port>` • `recovery-timeout=1` • `request-uri=sip:msml@<RM_IP_Address>:<RM_SIP_port>` Points to the Resource Manager for recording functionality. • `service-type=recorder`

Chapter

5 Preparing the Operating System for Media Server

This chapter describes how to prepare the Windows operating system for Genesys Media Server 8.5.x deployments. It contains the following section:

For information about the software prerequisites when deploying Media on the Windows platform, see “Prerequisites” on page 24.

Note: There are no requirements to prepare the Linux platform for Media Server. This section contains information about the Windows platform only.

Windows Services and Settings

Complete the tasks to configure Windows services and modify Registry settings on each host before you install the Media Server. See Task Summary: Specifying Windows Services and Settings, on page 124.

Warning! 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
Modify Windows Registry settings	**On Windows 2008 and 2012**: Accommodate environments that handle a large number of concurrent calls by changing a Windows Registry parameter on all GVP hosts before you begin the deployment. 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. If a value less than `30` is specified, the `DWORD` resets to the default of `240` seconds.

Task Summary: Specifying Windows Services and Settings (Continued)

<table>
<tr><th>Objective</th><th colspan="2">Related procedures and actions</th></tr>
<tr><td rowspan="2">Modify Windows OS settings</td><td colspan="2">3. On Windows 2008 only:
Set the <code>IP DiffServ</code> 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.</td></tr>
<tr><td colspan="2">4. 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:
<code>C:\bcdedit /set nx OptOut</code>
<code>C:\bcdedit /set pae ForceDisable</code>
Then, restart the server.</td></tr>
<tr><td rowspan="11">Enable or disable the required services, and set service start modes</td><td colspan="2">Modify the following services as indicated:</td></tr>
<tr><td>• Alerter:</td><td><code>Disabled</code></td></tr>
<tr><td>• Application Management:</td><td><code>Manual</code></td></tr>
<tr><td>• Com + Event System:</td><td><code>Manual</code></td></tr>
<tr><td>• Computer Browser:</td><td><code>Disabled</code></td></tr>
<tr><td>• Event Log:</td><td><code>Automatic</code></td></tr>
<tr><td>• Internet Information Server (IIS) Admin Service:</td><td><code>Automatic</code></td></tr>
<tr><td>• Indexing Service:</td><td><code>Disabled</code></td></tr>
<tr><td>• License Logging:</td><td><code>Disabled</code></td></tr>
<tr><td>• Messenger:</td><td><code>Disabled</code></td></tr>
<tr><td>• Net Logon:</td><td><code>Automatic</code></td></tr>
<tr><td>• NT LAN Manager (LM) Security Support Provider:</td><td><code>Manual</code></td></tr>
</table>

Task Summary: Specifying Windows Services and Settings (Continued)

Objective	Related procedures and actions	
Enable or disable the required services, and set service start modes (continued)	• 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`
Specify the recommended system performance settings.	See Procedure: Configuring Settings for System Performance.	

Procedure: Configuring Settings for System Performance

Purpose: To maximize the performance of the Media Server hosts in your deployment.

Summary

Complete this procedure for each Windows server that will host the Media Server.

Start of procedure

1. Go to `Control Panel > System > Advanced` tab.
2. In the `Performance` section, click `Settings`.
   The `Performance Options` page appears.
3. Click the `Advanced` tab.
4. In the `Processor scheduling` section, select `Background services`.
5. Set the virtual memory size:
   a. In the `Virtual memory` section, click `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. Click `Set`.
6. Click `OK` to exit all dialog boxes.
7. When prompted, restart the computer.

End of procedure

Next Steps

- No additional steps are required.

Genesys

Appendix

A Deploying the T-Server-CUCM to Media Server Connector

The Genesys T-Server-CUCM to Media Server Connector integrates with Cisco T-Servers (a Genesys component) that in turn, communicates with Cisco's Unified Communications Manager (UCM), to provide media services.

This chapter provides information about how to configure the Connector to function with Cisco T-Servers and switches. It contains the following sections:

Connector Overview

This section describes the Genesys T-Server-CUCM to Media Server Connector's role in environments where Cisco T-Servers and GVP are integrated and provides information about the Connector interfaces.

Connector Role

The Connector is a stand alone component, which acts as a gateway between Cisco T-Servers and switches, and GVP to provide media services. See Figure 8 on page 130.

Cisco T-Server (a Genesys component) is designed to communicate with Cisco switches rather than SIP switches. It does not use the SIP communication protocol to control media services. Instead, Cisco T-Server uses CP4SM (a Genesys proprietary protocol) and the Connector performs the *translation*

between CP4SM messages and SIP events. The translation is necessary, because GVP and Genesys Media Server only support SIP.

The Connector receives CP4SM message requests through its TCP connection to Cisco T-Server. It then converts these messages to SIP and MSML dialog requests and sends them to Media Server (through Resource Manager). After the Media Server has received the requests, the Connector sends Cisco T-Server an appropriate response.

Connector Interfaces

The Connector is a border element that interfaces with Cisco T-Server on one side and Resource Manager on the other. As in all GVP deployments, the Resource Manager acts as an arbitrator for the Media Control Platform (Media Server) to provide media services.

Figure 8 depicts a typical deployment architecture, in which the Connector is acting as a gateway between GVP and Cisco T-Server.

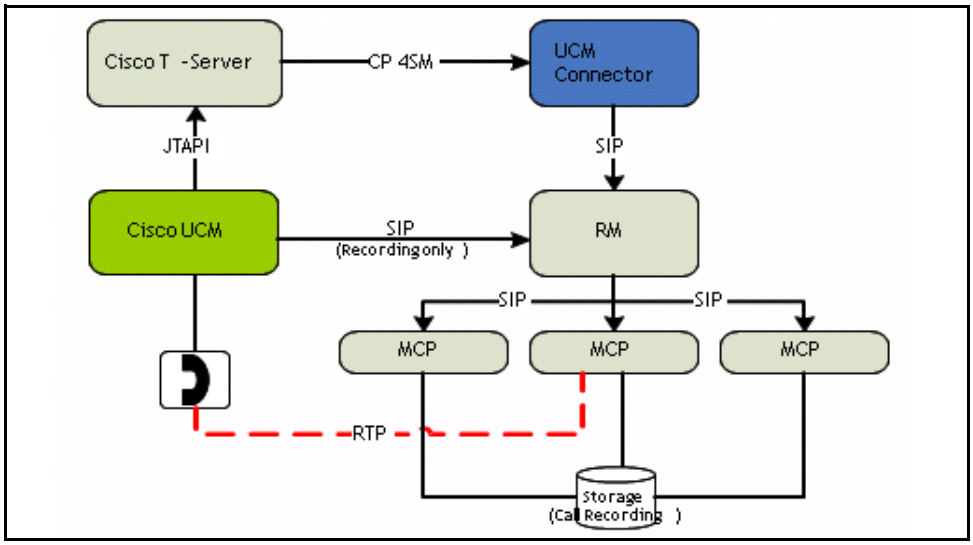

Figure 8: Connector Interfaces and Platform Architecture

How the Connector Works

This section describes how the Genesys T-Server-CUCM to Media Server Connector functions in environments where Cisco T-Servers and GVP are integrated. It contains the following topics:

- Operational Overview
- Connection and Call Setup
- Supported Media Operations

Operational Overview

The Connector uses CP4SM over TCP/IP to interact with Cisco T-Servers and switches to provide media services, such as playing media files and recording user announcement treatments.

CP4SM requests and responses have their own message structure, enabling connection establishment, configuration updates, re-connections, call setup, and tear down requests, and media operations.

The Connector supports the `PLAY FILE` and `RECORD FILE` media operations, which are described in detail in "Supported Media Operations" on page 133.

Connection and Call Setup

This section describes how the Connector establishes the initial connection to Cisco T-Server, how it reconnects if the initial attempt fails, and how it performs call setup and tear down.

Establishing the Connection with Cisco T-Severs

After the connection is configured, Cisco T-Server waits for connection requests from the Connector.

Connection Setup

The Connector establishes the connection with the Cisco T-Server by using the IP address and TCP port number that is configured in the `TServerAddress` configuration option in the `TServer` section of the Connector `Application`. It is configured in the format `<IPAddress>:<Port>`. For example `10.10.10.10:5060`.

Initial Handshake

After the connection is established, the initial handshake occurs (CP4SM `HELLO` and `HELLO ACK` messages) between the Connector and Cisco T-Server.

If the tracking timer for the `HELLO ACK` message expires before it is received and the handshake fails, the Connector disconnects and attempts to reconnect to the Cisco T-Server.

Note: If the CP4SM protocol version is not supported (for example, earlier than v1.7), an SNMP trap is generated.

Reconnect Attempts

The initial `HELLO` message that is sent to establish the connection includes an `UCMC Unique ID` parameter with a value of `0`. Each reconnect attempt uses the `UCMC Unique ID` from the previously sent `HELLO` message.Cisco T-Server also sends a `GET LEGS` message to the UCM Connector during the re-connection process. The UCM Connector responds with a `LEG INFO RECONNECTION` message, which contains the IDs of the call legs that need to be restored.

Note: The Connector cleans up the legs that are not explicitly restored before the `READY` message is received.

Multiple `LEG INFO RECONNECTION` messages can be sent in a single request, but they must all contain the same reference ID and there must be an `END LIST` message at the end of the list.

Ready Message

Cisco T-Server then sends a `READY` message (also sent after the initial handshake when connection is established and no reconnect attempts are required).

Establishing the Dialog With Media Server

The UCM Connector acts as an MSML client to provide access to the Media Server and respond to requests for media services. UCM Connector receives the service requests from Cisco T-Server in the form of CP4SM messages (`CREATE LEG`) and translates them to MSML messages that can be sent to the Media Server. The MSML content is transported in SIP `INFO` requests and responses.

After the UCM Connector receives the `CREATE LEG` request, it sends a SIP `INVITE` message to Media Server (through the Resource Manager) with the MSML user part in the SIP URI. The URI also contains the Resource Manager's IP address and port number in the format, `<RMIPAddress>:<RMPort>`. The `INVITE` message does not contain any SDP information.

SIP Dialog Establishment

After the SIP dialog is established, Cisco T-Server can send requests for media services, such as `PLAY FILE` or `RECORD FILE`. The UCM Connector initiates the MSML requests to Media Server by using SIP `INFO` messages. For each media service request from Cisco T-Server, the UCM Connector includes a `<dialogstart>` MSML element and generates a dialog identifier.

Setting Up the Call

Cisco T-Server initiates a new call setup by a `CREATE LEG` message to the UCM Connector.

Create Leg Request

When the UCM Connector receives a `CREATE LEG` message from the Cisco T-Server, it sends a SIP `INVITE` message to the Media Server (through the Resource Manager). The message contains an audio or video tag (but no SDP data) in the `Contact` header of `INVITE` message.

Establish SIP Dialog

The Media Server then sends a SIP `200 OK` response, which contains the RTP or RTCP port information. The UCM Connector extracts this port information, along with the IPv4-mapped-to-IPv6 address of corresponding Media Server, and sends it to Cisco T-Server.

Cisco T-Server sends a `LEG SETUP` message to the UCM Connector, which contains media codecs and the RTP or RTCP port information. The UCM Connector responds to Cisco T-Server with a `LEG SETUP ACK` message and then, passes the port information, along with SDP data, on to Media Server in a `SIP ACK` message.

Handling SIP Errors

If the UCM Connector receives a SIP `4xx`, `5xx`, or `6xx` error message from the Resource Manager, or the Media Control Platform (Media Server) in response to the initial SIP `INVITE` message, it passes the appropriate error code on to Cisco T-Server in the `CREATE LEG ACK` message.

Request URI Parameters

In the `Request URI` of SIP `INVITE` message, the UCM Connector also includes the `media-service` parameter with a value of `treatment` and the `tenant-dbid` parameter with the value set to the DBID of the Cisco T-Server tenant.

Note: To ensure the `tenant-dbid` parameter is included in the `Request URI` of the SIP `INVITE` message, the Cisco T-Server must be added as a connection in the `Connections` section of the UCM Connector `Application's` properties in Genesys Administrator. See the section "Creating a Connection to a Server" in chapter 7, "Post-Installation Configuration of GVP" in the *GVP Deployment Guide*.

Tearing Down the Call

The UCM Connector initiates call tear down of an existing call leg when it receives a `DESTROY LEG` message from Cisco T-Server. The leg ID is used to identify the leg that will be terminated.

The UCM Connector sends a SIP `BYE` message to Media Server and receives a SIP `200 OK` message in response. It then, responds to Cisco T-Server with a `DESTROY LEG ACK` message.

In certain cases, the UCM Connector receives a SIP `BYE` message from Media Server. When this happens, the UCM Connector does not send any further messages, but waits for Cisco T-Server to send another `DESTROY_LEG` message. The UCM Connector then identifies any stuck calls and cleans them up.

Supported Media Operations

The UCM Connector acts as an MSML client and requests media services from the Media Server for the caller. MSML is used to define and change the service to a user who is connected to Media Server. The service requests come to the UCM Connector from Cisco T-Server in the form of CP4SM protocol messages and are rendered into MSML messages and then sent to Media Server.

The UCM Connector can also send and receive SIP `INFO` requests and responses carrying MSML content. SIP `INFO` requests are used to send asynchronous mid-call messages within SIP. This occurs when the Cisco T-Server sends the UCM Connector a `CREATE LEG` message to establish a SIP dialog with Media Server (through Resource Manager).

The UCM Connector supports two types of media operations:

- `PLAY FILE`
- `RECORD FILE`

When the UCM Connector receives media service requests, such as PLAY FILE or RECORD FILE, it initiates an MSML dialog by sending a <dialogstart> element in the MSML content, with the requested file names that will be played, embedded in the <play> tag.

The PLAY FILE and RECORD FILE media operations can only be applied to call legs that are already established.

Initiating PLAY FILE

Cisco T-Server sends a PLAY FILE message to the Connector to trigger the media play functionality on the Media Server. Connector extracts the media file information, such as the number of files that will be played, and the file names. Files can be named with or without extensions or codec-specific suffixes.

Note: If the file name only is provided, Media Server checks the file extensions to find an optimal media file for the codec that is being negotiated for the call.

MSML Dialog

The Connector attempts to establish an MSML dialog with Media Server by sending SIP INFO message, which contains a MSML <dialogstart> element with a <play> element and a sequence of <audio> elements that point to prerecorded audio. The Media Server responds with a SIP 200 OK message.

The Connector sends a PLAY FILE ACK message to Cisco T-Server in response to the PLAY FILE request. The PLAY FILE ACK message contains an appropriate error code, if the Connector receives a SIP 4xx, 5xx, or 6xx error in response the SIP INFO request.

Stop Play Request

Cisco T-Server can interrupt the media file that is playing, by sending a STOP PLAY message to the UCM Connector. To abort the MSML dialog, the UCM Connector then sends a SIP INFO request to the Media Server, which contains a <dialogend> element to trigger the stop audio play and then, it sends a STOP PLAY ACK message to Cisco T-Server.

Session Info Message

If the UCM Connector receives a notification that the media has stopped playing before it receives a STOP PLAY message from Cisco T-Server, it will send a SESSION INFO message to the T-Server that contains an EndOfFile, EndOfSequence, or ErrStopped reason code.

Initiating RECORD FILE

The Cisco Unified Connection Manager (UCM) can use SIP for full-call recording by sending a request to the Media Server (through the Resource Manager) and by using a VoiceXML application. The UCM Connector is not involved in this operation. However, there might be a need to record an incoming call leg's audio stream. In this case, the Cisco T-Server can imitate the user-announcement treatment by sending a RECORD FILE message to the

UCM Connector. This recording is different from full call recording but the two recordings can be executed in parallel.

MSML Dialog The UCM Connector sends a SIP INFO message that contains a MSML <dialogstart> element to the Media Server to start the recording and then, sends a RECORD FILE ACK to Cisco T-Server, which contains the appropriate error code, if it receives SIP 4xx or 5xx errors in the SIP INFO response.

Stop Recording Request Cisco T-Server sends a STOP RECORDING message to the UCM Connector to stop the recording of the user-announcement treatment. The UCM Connector then sends a SIP INFO message that contains <dialogend> element to Media Server to stop the recording and then, sends a STOP RECORDING ACK response to Cisco T-Server.

Session Info Message If the UCM Connector receives a notification that the media has stopped recording before it receives a STOP RECORDING message from Cisco T-Server, it will send a SESSION INFO message to the T-Server that contains an Record<any>Limit reason code. Media Server can terminate the recording if it reaches the maximum recording time, size, or silence.

Session INFO and PING Requests

After receiving a SIP INFO message from the Media Server, the UCM Connector sends SESSION INFO messages to Cisco T-Server that provide information about the current play activity. The T-Server responds with a SESSION INFO ACKNOWLEDGE message.

At regular intervals, the Connector also receives CP4SM PING messages from Cisco T-Server that are sent to check the *live* status of the connection between the UCM Connector and the T-Server. The UCM Connector responds to these requests with a PING ACKNOWLEDGE message.

Deploying the Connector

This section describes how to deploy the Genesys T-Server-CUCM to Media Server Connector 8.5 on Windows and Linux operating systems, and provision it to integrate with Cisco T-Server. It contains the following sections:

- Task Summaries
- Installing the Connector
- Provisioning the Connector

Task Summaries

The Task Summary: Installing the T-Server-CUMC to Media Server Connector, on page 136 contains a list of tasks that are required to install the

Connector and includes links to detailed information that is required to complete these tasks.

Task Summary: Installing the T-Server-CUMC to Media Server Connector

Objective	Related procedures and actions
Prepare the host	1. Stop any antivirus software that might be running on systems that will host the Connector. Check the vendor documentation for your antivirus software configuration.
	2. Install the Local Control Agent on the Connector host. See Procedure: Installing the Local Control Agent (Windows), on page 86 or Procedure: Installing the Local Control Agent (Linux), on page 87.
Configure the host	• Configure a new host in the Configuration Database for the Connector. See Procedure: Configuring a Host in Genesys Administrator, on page 89.
Install the component	1. Create the `Application` object: a. Import the templates. See Procedure: Importing Application Object Templates Manually, on page 93. b. Create the `Application` objects. See Procedure: Creating Application Objects Manually, on page 95.
	2. Install the Connector. See Procedure: Installing the Connector (Windows), on page 137 or Procedure: Installing the Connector (Linux), on page 139.
Start the component	• Start the Connector component manually (or configure it to start automatically). See Procedure: Configuring Application Objects to Start Automatically, on page 105.

The Task Summary: Provisioning the T-Server-CUCM to Media Server Connector summarizes the tasks that are required to configure the Connector for the functionality that you want to use in your deployment and provides links to detailed information that is required to complete these tasks.

Task Summary: Provisioning the T-Server-CUCM to Media Server Connector

Objective	Related procedures and actions
Integrate with Resource Manager and Cisco T-Server.	• See Procedure: Integrating the Connector with Resource Manager and Cisco T-Server, on page 141.
Provision Full Call Recording.	• See "Configuring UCM Full Call Recording" on page 141.

Installing the Connector

This section describes how to install the Connector on Windows and Linux.

Before you begin to install the component, copy the Connector installation package to a directory on the host or to a network drive from which it can be downloaded.

Note: You can install multiple instances of the Connector. However, while a single Connector can interact with only one Cisco T-Server; a single Cisco T-Server can interact with multiple Connectors.

This section contains the following procedures:

- Procedure: Installing the Connector (Windows)
- Procedure: Installing the Connector (Linux)

Procedure: Installing the Connector (Windows)

Purpose: To install the Connector on the host.

Prerequisites

- The Connector host is prepared for installation. See "Preparing the Host" on page 85.
- The Connector `Application` object template is imported and an `Application` object is created. See "Preinstallation Activities" on page 91.

Start of procedure

1. Execute the `setup.exe` setup file:
   - If you are using the Genesys software DVDs for release 8.1.5 or earlier, browse to the `<GMS_Installation_DVD>\solution_specific\windows\rm\` folder.

     Note: CUMC is not present on the GVP 8.1.6 Release DVD, but you can still install it from the 8.1.5 release DVD. All future hot fixes for CUCM will be available through Software Download, as are the hot fixes for all other GVP components.

   - Locate the installation package. It may be located on a network drive, or in the FTP destination directory. Copy the `<DVDImage>\solution_specific\windows\rm\` folder to the local computer.
2. When the Genesys Deployment Wizard appears, click `Next`.

On the Connection Parameters page, enter the information in the Host and User sections, as shown in Table 21.

Table 21: 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.

These are the connection parameters for the Configuration Server.

3. On the Client Side Port Configuration page, select Use Client Side Port (if required). Enter the Port and IP Address.
4. On the Select Application page, select the UCM Connector Application object.
5. 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.
6. In the VP uCM Connector section, enter the information, as shown in Table 22.

Note: Step 6 is only required if you have deployed VP Reporting Server. For more information about deploying VP Reporting Server, see the *Genesys Voice Platform 8.5 Deployment Guide*.

Table 22: VP Reporting Server Section

Field	Description
Host	Enter the host name of the Reporting Server—for example, ReportServ1.
Port	Accept the default value, 61616, for the Reporting Server port number.

On the Ready to Install page, click Install.

7. When the installation is complete, click `Finish`.

End of procedure

Next Steps

- Configure the UCM Connector `Application` object to start automatically. See Procedure: Configuring Application Objects to Start Automatically, on page 105.

Procedure: Installing the Connector (Linux)

Purpose: To install the Connector component on a host.

Prerequisites

- The Connector host is prepared for the installation of GVP components. See “Preparing the Host” on page 85.
- The Connector `Application` object template is imported, and an `Application` object is created. See “Preinstallation Activities” on page 91.

Start of procedure

1. At the Linux host, log in as root, and then type `su`.
2. Navigate to the directory that contains the Connector installation package.
3. Type `chmod a+x install.sh`, and then press `Enter`.
4. Run the `./install.sh` command.

   The installation script is initiated.
5. At the prompt, enter the hostname of the Connector server—for example:

   ```
   Please enter the host name or press enter for "<local_host>"
   =><local_host>.
   ```
6. 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
   ```
7. 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
```

8. At the prompt, choose the application that you want to install—for example:

```
1 : UCMC-Host
2 : UCMC_8.5.000.09
3 : UCMC_8.5.000.19
=>3
```

9. At the prompt, enter the path to the directory in which the application files will reside—for example:

```
Press ENTER to confirm /<Install_Dir>/gvp81/UCMC_8.5.000.xx as the
destination directory or enter a new one =>
/opt/genesys/gvp/VP_UCMConnector_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 that indicates that the installation was completed successfully.

End of procedure

Next Steps

- Configure the Connector `Application` object to start automatically.

> **Note:** To start any `Application` object manually on a Linux host, type `<Install_Dir>/bin/run.sh`, and press `Enter`, where `<Install_Dir>` is the directory in which the application is installed.

Provisioning the Connector

Use the procedures in this section to enable the Connector to act as a translator between the Resource Manager and Cisco T-Server (a Genesys component) and to enable Cisco T-Server to access Media Server full call recording services.

Procedure: Integrating the Connector with Resource Manager and Cisco T-Server

Purpose: To configure the Connector to communicate with the Resource Manager and Cisco T-Server.

Prerequisites

- Cisco T-Server is installed and fully operational.

Start of procedure

1. Log in to Genesys Administrator.
2. On the `Provisioning` tab, select `Environment > Applications`.
3. Select the Connector `Application` object that you want to configure.

   The `Configuration` tab appears
4. Click the `Options` tab, and from the `View` drop-down list, select `Advanced View (Options)`.
5. In the `UCMC` section, in the `RMAddress` configuration option `Value` field, enter the Resource Manager's IP Address and port number in the format, `<RM_IPAddress>:RM_Port>`. For example, `10.10.10.10:5060`.
6. In the `TServer` section, in the `TServerAddress` configuration option `Value` field, enter the Cisco T-Server's IP Address and port number in the format, `<TServer_IPAddress>:TServer_Port>`. For example, `10.10.10.10:5060`.
7. In the `log` section, in the `verbose` configuration option `Value` field, enter `all` (to turn on full logs).
8. Save the configuration.

End of procedure

Next Steps

- Configure the Full Call Recording on the switch. See"Configuring UCM Full Call Recording".

Configuring UCM Full Call Recording

When UCM Full Call Recording is required, Cisco UCM initiates the recording session with Genesys Media Server directly, without the assistance of the Connector. A recording session is made up of two separate SIP dialogs, where each dialog carries the media stream for one of the two parties on the call.

To provision Cisco UCM to send full call recording requests to Media Server directly:

1. Configure a recording device on Cisco UCM. Refer to the Cisco UCM vendor documentation for the details.
2. In the Cisco UCM configuration, specify the Resource Manager's IP address and port number.

Customizing the Configuration

This section describes the key configuration options that you either must or may want to customize. You can configure these options in Genesys Administrator. Go to `Provisioning > Environment > Applications > <TS-CUCM Connector> > Options` tab.

It includes the following topics:

Important Configuration Options

The configurable Connector parameters are in the following configuration sections:

- `UCMC`—Parameters that determine the Connector's behavior.
- `TServer`—Parameters that enable the Connector's functionality.
- `ems`—Parameters determine Reporting behavior for call detail records (CDRs) and metrics.
- `log`—Parameters determine behavior for Management Framework logging.
- `sip`—Parameters required to define the SIP protocol level attributes for the SIP Stack.

Table 23 provides information about important T-Server-CUCM to Media Server Connector parameters that are not described in Chapter 3, "Configuring Common Features," in the *Genesys Voice Platform User Guide*. It provides parameter descriptions as well as the default parameter values that are preconfigured in the Connector `Application` object.

Unless indicated otherwise, all changes take effect on restart.

For information about all the available configuration options for the Media Control Platform, see the *Genesys Voice Platform 8.5 Configuration Options Reference.*

Table 23: Selected T-Server-CUCM to Media Server Connector Options

Option Name	Description	Valid Values and Syntax
`UCMC Section`		
Resource Manager IP Address and Port	Specifies the Resource Manager's IP address and port number in the format, `10.10.10.10:5080`.	string **Default value:** Empty

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
FIPS Enabled	Specifies whether FIPS mode is enabled or disabled in the Connector. When FIPS mode is enabled (set to `true`), only FIPS 140-2 approved ciphers and algorithms can be used in SSL connections.	• `True` • `False` **Default value:** `False`
Enable SIP Secure	Specifies whether or not SIP over TLS is used instead of SIP. If this parameter is set to `true`, the Resource Manager's IP address parameter name might need to be changed to point to the SIPS port of the Resource Manager.	• `True` • `False` **Default value:** `False`
`TServer` Section		
Cisco T-Server IP address and Port	Specifies Cisco T-Server's IP address and port number in the format, `10.10.10.10:5010`.	string **Default value:** Empty
Connector TCP port range for Cisco T-Server Connection	Specifies the local client-side TCP port range that will be used for the CP4SM transport protocol with Cisco T-Server. • If this configuration option is specified, the port range must be within the following range: `1030-65535`. For example, `1050-1070`. • If not specified, the Connector enables the operating system to choose the local port.	string **Default value:** Empty
T-Server Connection Retry Count	Specifies the number of times that the Connector will retry the connection to T-Server (if the connection is lost) before it goes into sleep mode for a configured amount of time.	Numeric **Default value:** `10`
`ems` Section		
Trace Flag	Specifies whether or not debug-level logging is enabled. • When enabled (set to `TRUE`), debug-level logs are processed and filtered the same way as other log levels. • When the disabled (set to `FALSE`), debug-level log messages are not processed.	• `True` • `False` **Default value:** `False`

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
MF Sink Log Filter	Specifies the behavior of the log messages that are sent to the MF sink in the format, `levels|moduleIDs|specifierIDs` (repeated if necessary). The values between the pipes can be in the following format: `m-n,o,p` -- for example, `0-4,5,6"`. The wildcard character `'*'` can also be used to indicate all valid numbers. For example: `*|*|*` indicates that all log messages must be sent to the sink. Alternatively, `0,1|0-10|*|4|*|*` indicates that `CRITICAL(0)` and `ERROR(1)` level messages with module IDs in the range of `0-10` and all `INFO(4)` level messages will be sent to the sink.	string **Default value:** `*|*|*`
SNMP Trap Sink Log Filter	Specifies which metrics will be delivered to the SNMP trap sink.	string **Default value:** `*|*|*`
`log` Section		
Verbose Level	Specifies whether or not a log output is created. If it is, the option specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are `Standard`, `Interaction`, `Trace`, and `Debug`.	Choose one of the following: `All`, `Debug`, `Trace`, `Interaction`, `Standard`, or `None` **Default value:** `Standard`
Output for level all	Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured.	string **Default value:** `../logs/UCMConnector`
Output for level debug	Specifies the outputs to which an application sends the Debug level and higher log events—that is, Standard, Interaction, Trace, and Debug level log events. The log output types must be separated by a comma when more than one output is configured.	string **Default value:** `../logs/UCMConnector`
Output for level trace	Specifies the outputs to which an application sends the Trace level and higher log events—that is, Standard, Interaction, and Trace level log events. The log output types must be separated by a comma when more than one output is configured.	string **Default value:** `../logs/UCMConnector`

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
Output for level interaction	Specifies the outputs to which an application sends the Interaction level and higher log events—that is, Standard and Interaction level log events. The log output types must be separated by a comma when more than one output is configured.	string **Default value:** `../logs/UCMConnector`
Output for level standard	Specifies the outputs to which an application sends the Standard level log events. The log output types must be separated by a comma when more than one output is configured.	string **Default value:** `../logs/UCMConnector`
Log Segmentation	Specifies the segmentation limit for a log file. If specified, this option sets the type of measurement (in kilobytes or megabytes) and the maximum size (in hours). If the current log segment exceeds the size that is set by this option, the file is closed and a new one is created.	string **Default value:** `10000`
Log Expiration	Specifies the time when the log files expire. If they do, this option sets the measurement that determines when they expire and the maximum number of files (segments) or days before the files are removed.	string **Default value:** `20`
Keep startup log file	Specifies whether a startup segment of the log that contains the initial T-Server configuration is retained. If so, this option can be set to `true` or to a specific size. • If this option value is set to `true`, the size of the initial segment is equal to the size of the regular log segment that is defined by the segment option. • If this option value is set to `false`, (that is, turned off) the size of the segment is ignored,	string **Default value:** `false`
Message file	Specifies the file name for application-specific log events. The name must be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific `*.lms` file. Otherwise, the application looks for the file in its working directory.	string **Default value:** Empty

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
Log messages format	Specifies the format of log record headers that are used by applications when logs are written to the log file. Using compressed log record headers improves application performance and reduces the log the file size. When this option value is set to `short`, the log file header or the log file segment contains information about the application, such as the application name, application type, host type, and time zone. A single log record within the file or segment omit this information. A log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg`. The message ID does not contain the prefix `GCTI` or the application type ID. The following are examples of a full format log record and a short format log record, respectivley: • `2002-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060 Application started` • `2002-05-07T18:15:33.952 Std 05060 Application started`	• `short` • `Full` **Default value:** `short`
Time generation for log messages	Specifies the method by which an application calculates the log record time when a log file is generated. The time is converted from the time-in-seconds since the Epoch (00:00:00 UTC, January 1, 1970). • Local Time (local)—The time of log record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information from the `Application's` host computer is used. • Coordinated Universal Time (UTC)—The time of log record generation is expressed as Coordinated Universal Time.	• `local` • `utc` **Default value:** `local`

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
Time format for log messages	Specifies, in a log file, how to represent the time when an application generates log records. A log record's time field can be in one of the following formats: • HH:MM:SS.sss (time)—The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. • According to the system's locale (locale)—The time string is formatted according to the system's locale. • ISO 8601 format (ISO8601)—The date in the time string is formatted according to the ISO 8601 format. For example, `2001-07-24T04:58:10.123`. Fractional seconds are given in milliseconds.	• `Time` • `locale` • `ISO8601` **Default value:** `Time`
Enable printing extended attributes	Specifies whether or not the application attaches any existing extended attributes to a log event that is sent to log output. Typically, log events of the Interaction log level and Audit-related log events contain extended attributes. • If this option value is set to `true`—Any existing extended attributes are attached to a log event that is sent to log output. • If this option value is set to `false`—Any existing extended attributes are not attached to a log event that is sent to log output. **Note:** Setting this option to `true` enables audit capabilities, but negatively affects performance. Genesys recommends enabling this option for the Solution Control Server (SCS) and Configuration Server when audit tracking is used. For other applications, refer to the *Genesys 7.5 Combined Log Events Help* to find out whether an application generates Interaction-level and Audit-related log events. If it does, enable the option only when new interaction scenarios are being tested.	• `true` • `false` **Default value:** `false`

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
Check point interval	Specifies how often (in hours) the Connector `Application` generates a check point log event to divide the log into sections of equal time. By default, the application generates this log event every hour. To prevent the generation of check-point events, set this option to `0`.	numeric **Default value:** `1`
Memory snapshot file name	Specifies the name of the file to which the Connector `Application` regularly prints a snapshot of the memory output, if it is configured to do this. The new snapshot overwrites the previously written data. If the application terminates abnormally, this file will contain the latest log messages. Memory output is not recommended for processors with a CPU frequency lower than 600 MHz.	string **Default value:** Empty
Folder for the temporary network log output files	Specifies the full path to the folder, in which an `Application` creates temporary files that are related to the network log output. If you change this option value while the `Application` is running, the change does not affect the currently open network output.	string **Default value:** Empty
Enable 6.X compatible log output priority	Specifies whether or not the Connector `Application` uses 6.x output logic. • If this option is set to `true`, the log, for the log level that is specified by the `Log Output` options, is sent to the specified output. • If this option is set to `false`, the log, for the log level that is specified by `Log Output` options and higher levels, is sent to the specified output.	boolean **Default value:** `false`
`sip` Section		
Contact Header User Name	Specifies the contact header user name that is generated by the platform.	A string of characters. **Default value:** `UCMConnector`

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
Local Transport IPv6 Address	Specifies whether or not the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message is set to this value if an IPv6 transport is used. The value must be a hostname or domain name. If this option value is left empty, the outgoing transport's actual IP and port is used for the `Via` and `Contact` headers. **Note:** If the domain name that is used in the `SRV` record query is specified, the `sip.transport.localaddress.srv` option value must be set to `true` to prevent the SIP stack from automatically generating the `port` part.	string **Default value:** Empty
Local Transport Address contains SRV domain name	Specifies whether or not the `sip.transport.localaddress option will` contain an `SRV` domain name. If this option value is set to `true`, the SIP stack does not automatically generate the `port` part. Otherwise, the outgoing transport's port number is used, together with the hostname that is specified by the `sip.transport.localaddress` option.	• `true` • `false` **Default value:** `false`
Transport Instance 0	Specifies the transport layer for the SIP stack and the network interfaces that are used to process SIP requests, in the following format: `sip.transport.x = transport_name type:ip:port [parameters]` Where: • `transport_name` is any string. • `type` is `udp`, `tcp`, or `tls`. • `ip` is the IP address of the network interface that accepts incoming SIP messages. • `port` is the port number on which the SIP stack accepts incoming SIP messages. • `[parameters]` defines any extra SIP transport parameters.	string **Default value:** Empty
Transport Instance 1	See the description for Transport Instance 0 in this table.	string **Default value:** Empty

Table 23: Selected T-Server-CUCM to Media Server Connector Options (Continued)

Option Name	Description	Valid Values and Syntax
Transport Instance 2	See the description for Transport Instance 0 in this table.	string **Default value:** Empty
Maximum Transmission Unit	Defines the maximum transmission unit (MTU) of the network interfaces. If the size of the SIP request is within 200 bytes of this value, the request is sent on a congestion-controlled transport protocol, such as TCP.	numeric **Default value:** `1500`
Local TCP Port Range	Specifies the local TCP port range that will be used for SIP transport. If this parameter is not specified, the Connector allows the operating system to choose the local port.	string **Default value:** Empty
Local TLS Port Range	Specifies the local TLS port range that will be used for SIP transport. If this parameter is not specified, the Connector allows the operating system to choose the local port.	string **Default value:** Empty
Local Transport IPv4 Address	Specifies whether or not the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message is set to this value if an IPv4 transport is used. The value must be a hostname or domain name. If this option value is left empty, the outgoing transport's actual IP and port is used for the `Via` and `Contact` headers. **Note:** If the domain name that is used in the `SRV` record query is specified, the `sip.transport.localaddress.srv` option value must be set to `true` to prevent the SIP stack from automatically generating the `port` part.	string **Default value:** Empty
SIP Static Route List	Specifies the static route groups in the form of a pipe-delimited list. Each route group contains a list or of IP addresses that are separated by commas. If a SIP request that is sent to an IP address that fails, any of the IP addresses within the route group can substitute as an alternate route destination. For example, `10.0.0.1,10.0.0.2|10.0.10.1,10.0.10.2` specifies two static route groups, with each group containing two routes that are the opposite of the other and can be used as alternate routes.	Any string of characters. **Default value:** Empty

Configuring Common Features

Task Summary: Configuring the Connector Common Features provides a number of common tasks that can be used to implement various types of functionality for the Connector.

Task Summary: Configuring the Connector Common Features

Objective	Related procedures and actions
Configure SIP Communications by configuring the SIP transports for the supported transport protocols.	Configure the `sip.transport.<x>` options: • `sip.transport.0` (Default value: `transport0 udp:any:5080`) • `sip.transport.1` (Default value: `transport0 udp:any:5080`) • `sip.transport.2` (Default value: `transport2 tls:any:5181 cert=$InstallationRoot$/config/x509_certificate.pem key=$InstallationRoot$/config/x509_private_key.pem`) **Notes:** 1. If all `sip.transport.x` values are empty, UDP, TCP, and TLS transports are all enabled, and listen from ports 5080, 5180, and 5181 respectively, on any network interface. To use this feature, the UseSecureSIP configuration option in the UCMC section of the UCMC application. 2. For setting IP `DiffServ` (ToS) field in the outgoing SIP messages, specify the ToS parameter: `sip.transport.<x>.tos`
Verify settings that determine behavior in relation to the SIP stack.	Review and, if necessary, modify the options that control such parameters as number of threads, size of the Maximum Transmission Unit (MTU) of the network interfaces, and number of connections: • For the Resource Manager, the relevant options are in the `proxy` configuration section. • For the Media Control Platform and Call Control Platform, the relevant options are in the `sip` configuration section.

Task Summary: Configuring the Connector Common Features (Continued)

Objective	Related procedures and actions
Enable IPv6 Communication.	Configure the following options: • `[sip] transport.localaddress_ipv6`—Use this option to specify that the `sent-by` field of the `Via` header and the `hostport` part of the `Contact` header in the outgoing SIP message is set to this value if an IPv6 transport is used. • `[sip] preferred_ipversion`—Use this option to specify the preferred IP version when a destination address resolves into multiple IP addresses that use different IP versions. The first IP address processed that matches the preferred IP version is used. However, if a `sip.transport` is not defined for the preferred version, a defined version that matches one of the processed IP addresses is used. Valid values are `ipv4` and `ipv6`. • `[sip] route.default.udp.ipv6`—Use these options to specify the default IPv6 route for UDP. The number denotes the transport that is defined in the `sip.transport.x` configuration option. • `[sip] route.default.tcp.ipv6`—Use these options to specify the default IPv6 route for TCP. The number denotes the transport that is defined in the `sip.transport.x` configuration option. • `[sip] route.default.tls.ipv6`—Use these options to specify the default IPv6 route for TLS. The number denotes the transport that is defined in the `sip.transport.x` configuration option. • `[sip] transprt.x`—Use this option to specify the transport layer for the SIP stack and the network interfaces that a reused to process SIP requests. **Note:** or a complete description of these UCM Connector options and their default values, see Table 23 on page 143.
Configure the Client-side connections	1. Configuration Server—TCP dynamically configured by the OS. Allows the UCM Connector to receive configuration data and updates from the Configuration Server. 2. Message Server—TCP dynamically configured by the OS. Sends logs to the Message Server if sink logging is turned on.

Task Summary: Configuring the Connector Common Features (Continued)

Objective	Related procedures and actions
Configure the Client-side connections (continued)	3. Local Control Agent (LCA)—TCP dynamically configured by the OS. Allows the UCM Connector to send status information to the Solution Control Server (SCS). 4. SIP—UDP SIP—TCP dynamically configured by the OS Option: `sip.transport.x` **Default value:** `5180` Allows the UCM Connector to provide SIP Service. 5. Cisco T-Server—TCP Option: `UCMCClientPortRange` **Default value:** Empty TCP dynamically configured by the OS with ports ranging from 1030-65535. Allows the UCM Connector to receive messages from Cisco T-Server.
Customize the SIP responses and alarms.	This option is used to signal certain events and conditions for the UCM Connector: • `sip.copyunknownheaders`
Configure the options that determine the session timers that the Resource Manager uses to manage sessions	In the `sip` section of the UCM Connector `Application`, configure the following options: • `sessionexpires`—The default session expiry value, in seconds. The SIP session expires if no `Re-INVITE` messages are sent or received within this period. Numeric **Default value:** `1800` • `min_se`—The minimum session expiry value, in seconds. The minimum duration of expiry that a SIP stack will accept from a user agent client. Numeric **Default value:** `90`
Configure Logging parameters	The following parameters (with default values) for the various log levels can be customized in the `log` configuration section of the UCM Connector `Application`: • all—`../logs/UCMConnector` • debug—`../logs/UCMConnector`

Task Summary: Configuring the Connector Common Features (Continued)

Objective	Related procedures and actions
Configure Logging parameters (continued)	• expire—`20 (files)` • interaction—`../logs/UCMConnector` • message_format—`short` • segment—`10000 KB` • standard—`../logs/UCMConnecto` • time-format—`time` • trace—`../logs/UCMConnecto` • verbose—`standard`
Configure Reporting.	In the `ems` section of the UCM Connector `Application`, configure the `logconfig.MRSINK` option with a value of `*|*|*`

Proprietary Error Codes

This section contains information about the Genesy proprietary CP4SM protocol error codes that are generated by the Connector and provides a mapping of these error codes to the SIP MSML error codes.

Table 24 lists the errors that are generated by the Connector.

Table 24: Error Codes Generated by the Connector

Name	Value	Description
OK	0	There are no errors.
Internal error	1	Any kind of internal error (interpreted as unknown error).
Network error	2	Any kind of network error.
Connection closed	3	The TCP/IP socket was closed by either client or server.
Unknown command	4	The command code is unknown.
Leg not found	5	The call leg was not found.
Unknown response	6	The response ID is unknown (no request was issued previously with this ID).
RTP socket not found	7	The RTP socket was not found in the list.
RTCP socket not found	8	The RTCP socket was not found in the list.

Table 24: Error Codes Generated by the Connector (Continued)

Name	Value	Description
No more resources	9	There are no more sockets available to be opened (either TCP or UDP).
On call error	10	If a `Break` request is not received prior to a `Join` request
Not joined error	11	If a Break request is received when the call leg is not yet in Joined state.
Unsupported codec	12	The requested codec is unsupported.
Stream Manager not found	13	The specified Stream Manager instance is not found.
Wrong event	14	The event is unknown or out of sequence.
Wrong ID	15	The leg or Stream Manager ID is incorrect.
Dummy Stream Manager	16	No real Stream Manager instance is presented and the dummy instance cannot process this command.
Wrong protocol	17	The protocol version is incorrect.
Leg already exists	18	If the leg already exists when a `create message` request is received.
Operation postponed	19	N/A to the Connector.
T.120 server error	20	Stream manager cannot create the T.120 server instance.
Bad parameter	21	A bad parameter is found in a message.
Invalid state	22	The leg state contradicts the received command.
Unknown destination	23	The network destination is unknown.
ASN compiler error	24	An ASN compiler error has occurred.
Unauthorized connection	25	An unauthorized client attempted to connect.

Table 25 provides a mapping of the SIP MSML error codes to the proprietary-generated CP4SM protocol error codes.

Table 25: SIP MSML to CP4SM Protocol Error Codes Mapping

MSML response Code	Description	CP4SM error code	Description
400	Bad request	21	`CP4SM_ERR_BAD_PARAM`

Table 25: SIP MSML to CP4SM Protocol Error Codes Mapping (Continued)

MSML response Code	Description	CP4SM error code	Description
401	Unknown element	21	CP4SM_ERR_BAD_PARAM
402	Unsupported element	21	CP4SM_ERR_BAD_PARAM
403	Missing mandatory element content	21	CP4SM_ERR_BAD_PARAM
404	Forbidden element content	21	CP4SM_ERR_BAD_PARAM
405	Invalid element content	21	CP4SM_ERR_BAD_PARAM
406	Unknown attribute	21	CP4SM_ERR_BAD_PARAM
407	Attribute not supported	21	CP4SM_ERR_BAD_PARAM
408	Missing mandatory attribute	21	CP4SM_ERR_BAD_PARAM
409	Forbidden attribute is present	21	CP4SM_ERR_BAD_PARAM
410	Invalid attribute value	21	CP4SM_ERR_BAD_PARAM
420	Unsupported media description language	21	CP4SM_ERR_BAD_PARAM
421	Unknown media description language	21	CP4SM_ERR_BAD_PARAM
422	Ambiguous request (both URI and inline description)	21	CP4SM_ERR_BAD_PARAM
423	External document fetch error	21	CP4SM_ERR_BAD_PARAM
424	Syntax error in foreign language	21	CP4SM_ERR_BAD_PARAM
425	Semantic error in foreign language	21	CP4SM_ERR_BAD_PARAM
426	Unknown error executing in foreign language	21	CP4SM_ERR_BAD_PARAM
430	Object does not exist	01	CP4SM_ERR_INTERNAL_ERR
500	Internal media server error	01	CP4SM_ERR_INTERNAL_ERR
503	Service unavailable	01	CP4SM_ERR_INTERNAL_ERR
510	Not in service	01	CP4SM_ERR_INTERNAL_ERR
511	Service unavailable	01	CP4SM_ERR_INTERNAL_ERR
520	No resource to fulfill request	01	CP4SM_ERR_INTERNAL_ERR
521	Internal limit exceeded	01	CP4SM_ERR_INTERNAL_ERR

Specifiers for EMS Logging and Reporting

Internal GVP identifiers are required for advanced configuration of EMS Logging and Reporting. The Module ID for the UCM Connector `Application` is 244. Table 26 describes the specifiers for the UCM Connector module.

Table 26: UCM Connector Specifiers

Specifier ID	Specifier Name
20101	UCMC_INITIALIZATION_ERROR
20102	UCMC_INVALID_PTR_ERROR
20103	UCMC_INTERNAL_ERROR
20104	UCMC_CCILIB_ERROR
20105	UCMC_SNMPLIB_ERROR
20106	UCMC_CONFIG_OBJECT
20111	UCMC_TSVR_CONNECTION_DOWN
20112	UCMC_TSVR_CONNECTION_UP
20113	UCMC_TSVR_CONNECT_ERROR
20114	UCMC_TSVR_HANDSHAKE_FAILURE
20115	UCMC_TSVR_CP4SM_VER_MISMATCH
20116	UCMC_TSVR_UNSUPPORTED_REQUEST %
20117	UCMC_TSVR_UNSUPPORTED_CODEC
20118	UCMC_TSVR_CP4SM_ERROR_RESP
20119	UCMC_TSVR_LEG_DOESNT_EXIST
20120	UCMC_TSVR_SESSION_INFO_NOTIFY
20131	UCMC_STARTED
20132	UCMC_STOPPING
20133	UCMC_SHUTDOWN
20141	UCMC_TRACE_CALL
20142	UCMC_TRACE_MEDIA_OPERATION
20143	UCMC_TRACE_PING

Genesys

Appendix

B MSML Specification

This appendix describes GVP support for MSML. The elements and attributes defined by MSML are listed in the appendix, along with notes on behavior important to MSML developers, but full descriptions are not given. The MSML specification, found at `http://http://tools.ietf.org/rfc/rfc5707.txt`, provides a full description of all MSML elements and attributes.

Note: CCXML supports only a subset of the MSML specification in `<dialogstart/dialogprepare>`—the Dialog Core, Dialog Base, and Dialog CPA packages.

This appendix contains the following sections:

MSML Core Package

<msml> element

Attribute

`version`

<send> element

Attributes

`event`

`target`

The target must be part of the MSML session associated with the request, following the syntax:

```
conn:connID | dialog:dialogID[/primitive[.primitiveID]]valuelist
```

`mark`

<result> element

Attributes

`response`

`mark`

<event> element

Attributes

`name`

`id`

Child Elements

`<name>`

No attributes

`<value>`

No attributes

MSML Conference Core Package

<createconference> element

Attributes:

`name`

`deletewhen`

Values supported: `nomedia` and `never`.

`mark`

Child Elements

<audiomix>

Child Element:

<asn>

<gvp:recorder>

Attribute:

`state`

Supported string values: `pause` and `start`.

Child Element:

<gvp:param>

Child Element:

<gvp:param>

Attribute

name

Valid values are as follows:

- `dest` (default)—Specifies the recording destination. A value that starts with `sip:` or `sips:` indicates that the audio streams must be sent to a third-party recorder. A value that starts with `file:` indicates a local file recording that is using a single file with dual channels, similar to NETANN multi-channel recording. This parameter value overrides the `recdest` parameter that is specified in the SIP Request URI.
- `dest2` (optional)—Specifies the recording destination for a duplicate recording. The format of this value is the same as the one for the `dest` parameter value.
- `id`—Identifies or names the recordings. If a third-party recorder is used, this ID is set in the user-part of the Request URI (for example, `record=<ID>`) to identify the recordings (in conjunction with the `dn=<dn>` Request URI parameter). For a local file recording, the ID is used to generate the file name.
- `recmediactl` (optional)—Specifies the type of media control to use for a third-party recorder. If the value is `1`, only one SIP session (with two m-lines in the SDP) is used to communicate with the recording manager. If the value is `2`, an individual SIP session is established for each stream. This value overrides the same parameter that is specified in the SIP Request URI.

`<videolayout>`

Child Element:

`<selector>`

Attribute:

`state`

Supported values:

`vas`, `fixed`, `confrole`.

<destroyconference>

Attributes:

`id`

`mark`

<modifyconference>

Attributes:

`id`

`mark`

Child Elements:

`<audiomix>`

Child Element:

`<asn>`

`<gvp:recorder>`

Attribute:

`state`

Supported string values are: `pause`, and `start`.

Child Element:

<gvp:param>

Child Element:

<gvp:param>

Attribute:

name

Valid values are as follows:

- `dest` (default)—Specifies the recording destination. A value that starts with `sip:` or `sips:` indicates that the audio streams must be sent to a third-party recorder. A value that starts with `file:` indicates a local file recording that is using a single file with dual channels, similar to NETANN multi-channel recording. This parameter value overrides the `recdest` parameter that is specified in the SIP Request URI.
- `dest2` (optional)—Specifies the recording destination for a duplicate recording. The format of this value is the same as the one for the `dest` parameter value.
- `id`—Identifies or names the recordings. If a third-party recorder is used, this ID is set in the user-part of the Request URI (for example, `record=<ID>`) to identify the recordings (in conjunction with the `dn=<dn>` Request URI parameter). For a local file recording, the ID is used to generate the file name.
- `recmediactl` (optional)—Specifies the type of media control to use for a third-party recorder. If the value is `1`, only one SIP session (with two m-lines in the SDP) is used to communicate with the recording manager. If the value is `2`, an individual SIP session is established for each stream. This value overrides the same parameter that is specified in the SIP Request URI.

`<videolayout>`

Child Element:

`<selector>`

Attribute:

`method`

Supported values are `vas`, `fixed`, and `confrole`.

<Join>

Attributes:

`id1`

`id2`

`mark`

`playbeep:` Valid values are `to-id1` or `to-id2`. If the attribute is not set, no beep is played.

`gvp:confrole:` Valid values:

- `Regular` (default)—The customer call leg receives audio from the mixer, and video from the agent/student call leg (or from a file in a Push Video scenario)

- `Agent/Student`—The agent or student call leg receives audio from the mixer, and video from the regular call leg (or from a file in a Push Video scenario)
- `Coach`—The supervisor call leg in a Whisper Coaching scenario receives audio from the mixer, and the same video stream as the agent/student leg.
- `Monitor`—The supervisor or recording device call leg in a Silent Call Monitoring scenario receives audio from the mixer, and the same video stream as the agent/student leg.
- `Push`—The media playback device call leg does not receive any media; incoming audio stream is pushed to the mixer, and the video stream is pushed to a regular or customer call leg.
- `Push-All`—The media playback device call leg provides audio to the mixer, and the video stream to all call legs in the conference call.

<modifystream>

Attributes:

`id1`

`id2`

`mark`

`gvp:confrole`: See `confrole` definitions in `<join>` element on page 163.

Child Elements:

`<stream>`

Attributes:

`media`

`dir`

<unjoin>

Attributes:

`id1`

`id2`

`mark`

Child Element:

`<stream>`

Attributes:

`media`

dir

MSML Dialog Core Package

<dialogstart>

Attributes:

target

src

type

name

mark

gvp:confrole: See confrole definitions in <join> element on page 163.

Child Elements:

<play>

See “MSML Dialog Base Package” on page 166

<dtmfgen>

See “MSML Dialog Base Package” on page 166

<record>

See “MSML Dialog Base Package” on page 166

<collect>

See “MSML Dialog Base Package” on page 166

<cpd>

See “MSML Dialog Call Progress Analysis Package” on page 171.

<dialogend>

Attributes:

id

mark

<send>

Attributes:

`event`

`target`

<exit>

Attribute:

`namelist`

<disconnect>

Attribute:

`namelist`

<dialogprepare>

The `<dialogprepare>` element is supported as an extension of the MSML dialog core package. It is equivalent to the `<dialogstart>` element, except that the dialog does not start until the *start* event is received. When a *start* event is sent to the preparing dialog, the dialog is joined to its target and starts execution. The `<dialogprepare>` element is supported for VoiceXML dialogs. only.

MSML Dialog Base Package

<play>

Attributes:

`id`

`iterate`

`maxtime`: Supported in single prompt or multiple prompts in a queue of prompts when iterate is equal to 1.

`barge` (optional)—Defaults to `false`.

`cleardb` (optional)—Defaults to `false`.

`offset`: Supported in a single prompt in a queue of prompts, with iterate equal to `1`.

`gvp:precheck` (optional): Optional. Valid values are `true` or `false` (default).

If set to `true`, GVP pre-checks file availability and responds with `file not found` when the audio or video prompt file is not found.

If set to `false`, GVP does not pre-check file availability. If the file cannot be found at prompt play time, the `<play>` element ends and the `play.end` shadow variable is set to `error`.

Event:

`terminate`

Shadow Variables:

`play.amt`

`play.end`: Possible values:

- `play.complete`
- `play.complete.barge`
- `play.terminated`
- `play.timelimit`
- `play.error`
- `play.killsession`
- `play.unknown`

Child Elements:

`<audio>`

Attributes:

`uri`

`format`

`iterate`

gvp:streaming attribute

`gvp:streaming` (optional): Valid values are `true` and `false` (default, disabled). When set to true:

- HTTP streaming (where the content is played as it is being fetched) is enabled.
- Availability of the prompt file is determined by checking whether all HTTP headers have been received (the rest of the data can continue to arrive after the check).

- Fetch timeout is interpreted as the maximum time to receive all HTTP headers.

`<video>`

Attributes:

`uri`

`format`

`iterate`

`<playexit>`

No attributes

<dtmfgen>

Attributes:

`id`
`digits`
`dur`
`interval`

Event:

`terminate`

Shadow Variables:

`dtmfgen[.id-if-specified].end`: Possible values:

`dtmfgen.complete`
`dtmfgen.terminated`
`dtmfgen.error`
`dtmfgen.killsession`
`dtmfgen.unknown`

Child Element:

`<dtmfgenexit>`

No attributes

Child Element:

`<send>`

<record>

Attributes:

id

dest

format

profile

level

maxtime

prespeech

postspeech

termkey

gvp:record_mode Possible values:

- parallel—Recording starts at the same time as the prompt starts.
- sequential (default)—Recording starts after the prompt ends.
- cpd—Recording is for Call Progress Detection. There must be no attribute in the <record> element (mandatory). The recording is done in the same way as if the recording was initiated from <cpd> element.

Event:

terminate

Shadow Variables:

record.len

record.end: Possible values are as follows:

- record.failed.prespeech
- record.complete.maxlength
- record.complete.postspeech
- record.complete.termkey
- record.complete.sizelimit
- record.error
- record.terminated
- record.killsession
- record.complete.unknown

record.recordid

record.size

Child Elements:

<play>

<recordexit>

No attributes

<collect>

Attributes:

id
cleardb
iterate

Event:

terminate

Shadow Variables:

dtmf.digits
dtmf.len
dtmf.last
dtmf.end

Child Elements:

<play>
<pattern>

Attributes:

digits: Required format: max=n

...where n specifies the number of digits to collect.

format: Valid value: moml+digits

iterate

Child Element:

<send>

<dtmfexit>

No attributes

Child Element:

<send>

MSML Dialog Call Progress Analysis Package

<cpd>

<cpd> is a GVP-specific entity.

The CPD primitive supports three states of detection and one state of nondetection.

- preconnect—Detects preconnect events.
- postconnect—Detects postconnect events.
- beepdetect—Detects answering-machine beep.
- buffer—Does not detect events, but buffers a configurable amount of audio that is used after transitioning to a postconnect state.

Events can be sent to the CPD primitive to change the detection state. The CPD primitive automatically changes from the postconnect state to the beepdetect state if it detects an answering machine while it is in the postconnect state.

The CPD primitive is complete when it detects one of the following terminating results:

- cpd.sit.nocircuit
- cpd.sit.reorder
- cpd.sit.operationintercept
- cpd.sit.vacantcircuit
- cpd.sit.custom1 (2,3,4)
- cpd.busy
- cpd.connect
- cpd.human
- cpd.fax
- cpd.machine
- cpd.beep
- cpd.preconnect_timeout
- cpd.silence
- cpd.beeptimeout
- cpd.unknown

The primitive then executes <cpdtimeout>, <cpdsilence>, <beeptimeout>, or <cpddetect> (if present) depending on the result, and then execute <cpdexit>.

Attributes:

- `beeptimeout` (optional)—It defines the amount of time in seconds for CPD to timeout in `beepdetect` state. When the timeout elapses, the child element `<beeptimeout>` is executed. This timeout applies only when the primitive is in the `beepdetect` state. This timeout is implicitly canceled when the state changes to another state. When this attribute is not set, the `beeptimeout` state defaults to the value of the `[msml].cpd.beeptimeout` configuration parameter.
- `connectnosignal` (optional)—Valid values are `true` or `false` (defaults to `false`). When the value is set to `true`, and in the preconnect state, the CPD element automatically transitions to the `postconnect` state when a call is determined to be connected. Otherwise, the CPD element remains in the `preconnect` state until it is notified via an MSML event to transition to a new state or until the `preconnecttimeout` event occurs.
- `id`—Identifies a `<cpd>` element when events are sent to it. The sender must refer to the `<cpd>` element using an address in the form `cpd[.id=if-specified]`.
- `initial` (optional)—Defines the initial detection state. Valid values are as follows:
  - `preconnect` (default)
  - `postconnect`
  - `beepdetect`

`postconnectpref` (optional)—Optional, defines how postconnect CPD detection prioritizes which results are detected. You can configure the valid values in the MCP configuration. Default uses the configured defaults for length detection.

Valid values:

- `default` (default)—Answering Machine (AM) detection is performed as configured by default. A longer audio response indicates that a machine answered.
- `machine`—Decrease the maximum audio length setting that indicates connection to an answering machine, i.e., assume that a machine response is likely.
- `no_machine`—When SIT tones and FAX have not been detected, the connected call is considered to have been answered by a live voice.
- `voice`—Increase the maximum audio length setting that indicates connection to an answering machine, i.e., assume that a live voice response is likely.

- `postconnecttimeout` (optional)—Defines the amount of time in seconds for the CPD to timeout in the `postconnect` state. When the timeout elapses, the child element `<cpdsilence>` is executed. This timeout applies only when the primitive is in the `postconnect` state. The timeout is implicitly canceled when the state changes to another state. If this attribute is not set, it defaults to the `[msml].cpd.postconnecttimeout` configuration parameter.

- `record` (optional)—Defines whether or not the media that is received during CPA is recorded. Valid values are:
  - `true`—Recording occurs.
  - `false` (default)—Recording does not occur.

Recordings are saved to the directory provided by the value for the `[msml].cpd.record.basepath` parameter. The format type and file extension are determined by the value of the `[msml].cpd.record.fileext` parameter. The name of the recording is randomly generated.

Events:

Note: While the CPD element is executing, it can receive the following events, which produce changes in state.`beepdetection`—Sets the CPD primitive states to beep detection.

- `postconnect`—Sets the CPD primitive state to postconnect.
- `terminate`—Terminates the `<cpd>` element.

Shadow Variables:

- `cpd[.id-if-specified].result`—The string value that specifies the result of CPD. Valid values:
  - `cpd.sit.nocircuit`
  - `cpd.sit.reoder`
  - `cpd.sit.operatorintercept`
  - `cpd.sit.vacantcircuit`
  - `cpd.sit.custom1`
  - `cpd.sit.custom2`
  - `cpd.sit.custom3`
  - `cpd.sit.custom4`
  - `cpd.busy`
  - `cpd.connect`
  - `cpd.human`
  - `cpd.fax`
  - `cpd.machine`
  - `cpd.beep`
  - `cpd.preconnect_timeout`
  - `cpd.silence`
  - `cpd.beeptimeout`
  - `cpd.unknown`
- `cpd[.id-if-specified].end`—The string value that specifies the reason for terminating the `<cpd>` element. The possible values are as follows:
  - `cpd.terminated`
  - `cpd.completed`
  - `cpd.failed`

- cpd.recordfailed
- cpd.unknown

- cpd[.id-if-specified].recfile—Contains the path to the CPA recording. If there is no recording, the value is undefined.

Child Elements:

<beeptimeout>

Executed when the beeptimeout is elapsed, meaning that the CPD did not detect an answering-machine beep within the timeout period. The primitive is completed after this element and executes the <cpdexit> element.

No attributes

<cpddetect>

Executed when a call-progress event is detected.

No attributes

<cpdexit>

Invoked when the CPD is completed or is terminated as a result of receiving the terminate event.

No attributes

<cpdsilence>

Executed when the postconnect timeout is elapsed, meaning that there is silence on the media stream. The primitive is completed after this element and executes the <cpdexit> element.

No attributes

<cpdtimeout>

Executed when the preconnect timeout is elapsed. The primitive is completed after this element and executes the <cpdexit> element.

No attributes

MSML Usage Example

The following code sample initiates the CPD detection at the preconnect state, with five-second timeout period:

```
<?xml version="1.0" encoding="UTF-8"?>
<msml version="1.1">
   <dialogstart target="conn:xxxx" name="cpd"
   type="application/moml+xml">
      <cpd initial="preconnect" preconnecttimeout="5s">
         <cpdtimeout>
```

```
            <send target="source" event="done" namelist="cpd.recfile
cpd.end cpd.result"/>
         </cpdtimeout>
   </cpd>
   </dialogstart>
</msml>
```

If no media activity is detected during the `preconnect` state, after five seconds, the CPD completes and sends the event with the following shadow variables:

- `cpd.recfile = undefined`
- `cpd.end = cpd.completed`
- `cpd.result = cpd.preconnect_timeout`

Appendix

C Call Flows

This appendix describes Cisco T-Server (a Genesys component) and T-Server-CUCM to Media Server Connector 8.5 call flows when requesting media services. It contains the following section:

PLAYFILE and RECORDFILE

This section contains call flows diagrams that illustrate how Cisco T-Server and the T-Server-CUCM to Media Server Connector establish and terminate the call leg, and how it handles `PLAYFILE` and `RECORDFILE` requests.

Call Leg Establishment

Figure 9 on page 178 illustrates how the call leg is established.

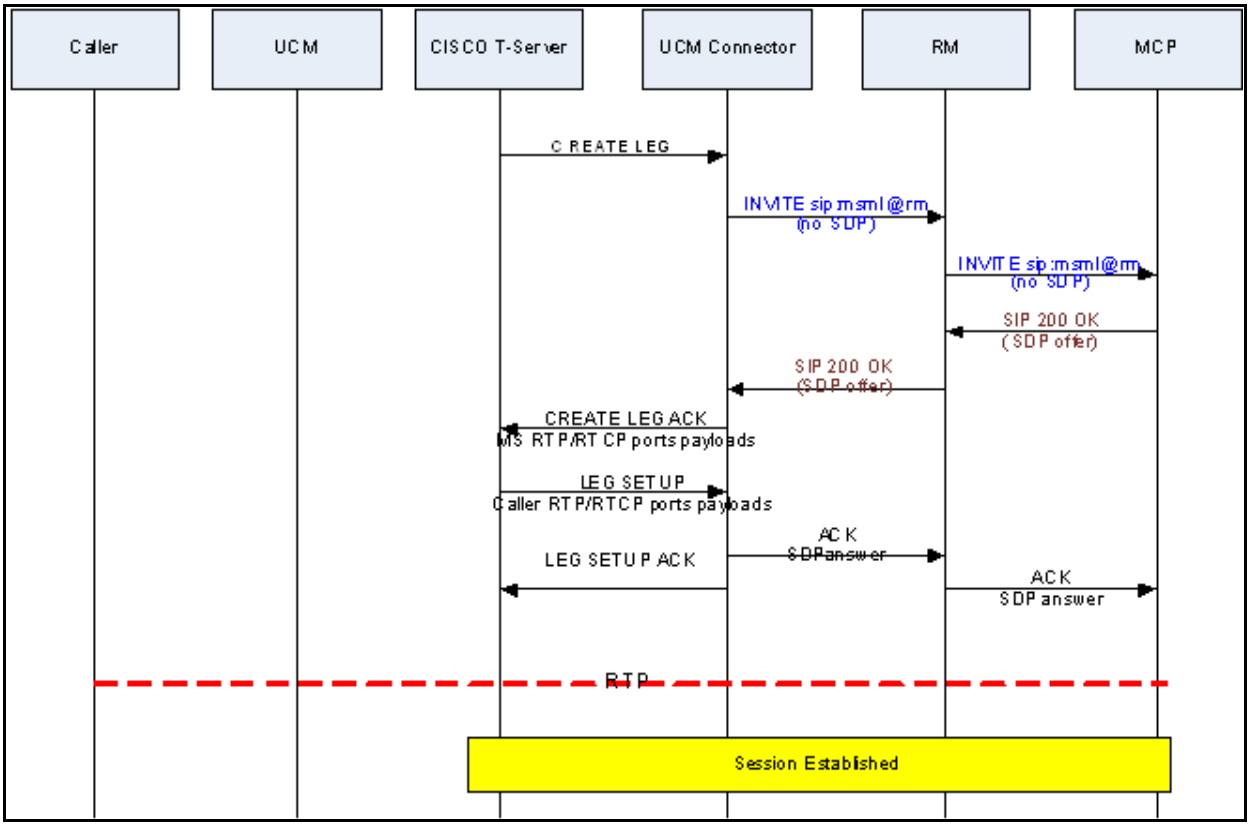

Figure 9: Call Leg Establishment—UCM Connector

PLAYFILE Request

Figure 10 on page 179 illustrates how the a `PLAYFILE` request is handled.

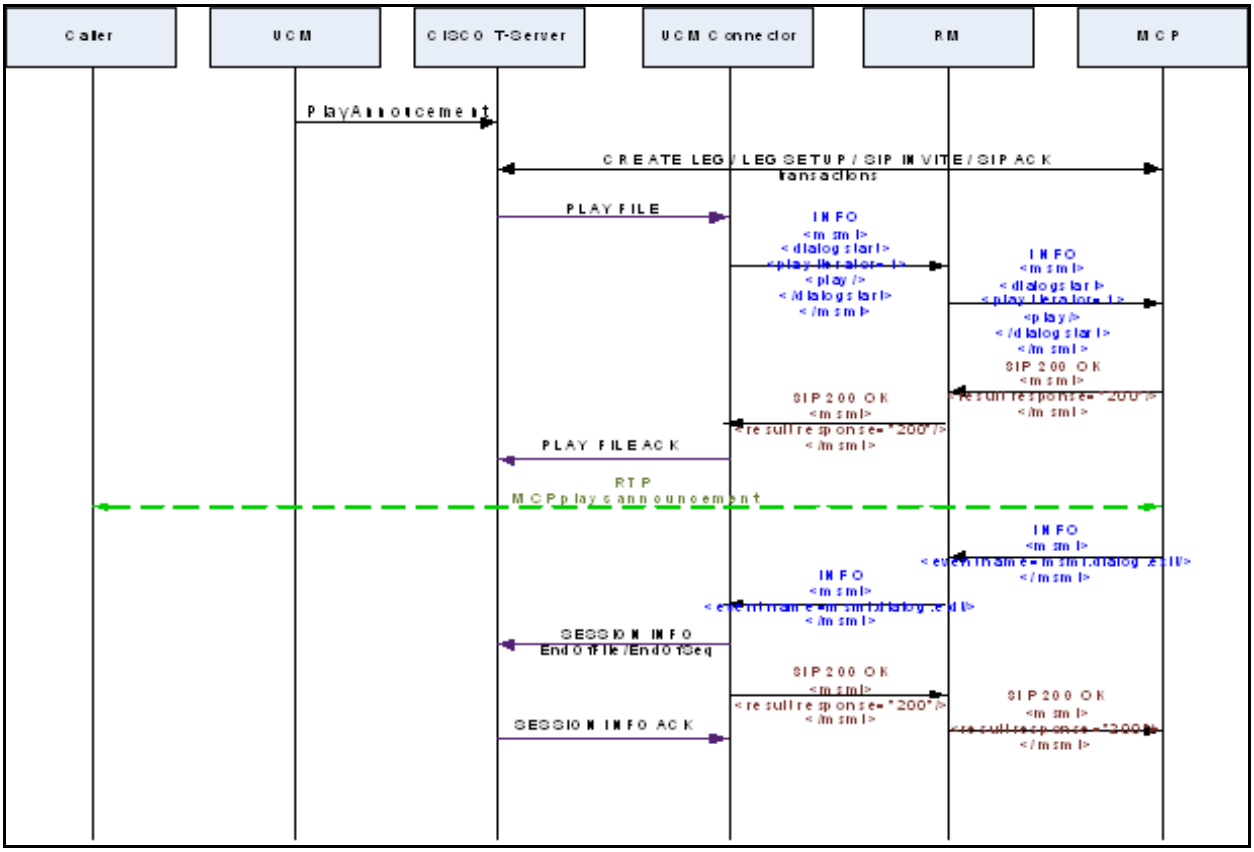

Figure 10: PLAYFILE Request—UCM Connector

RECORDFILE Request

Figure 11 illustrates how a `RECORDFILE` request is handled.

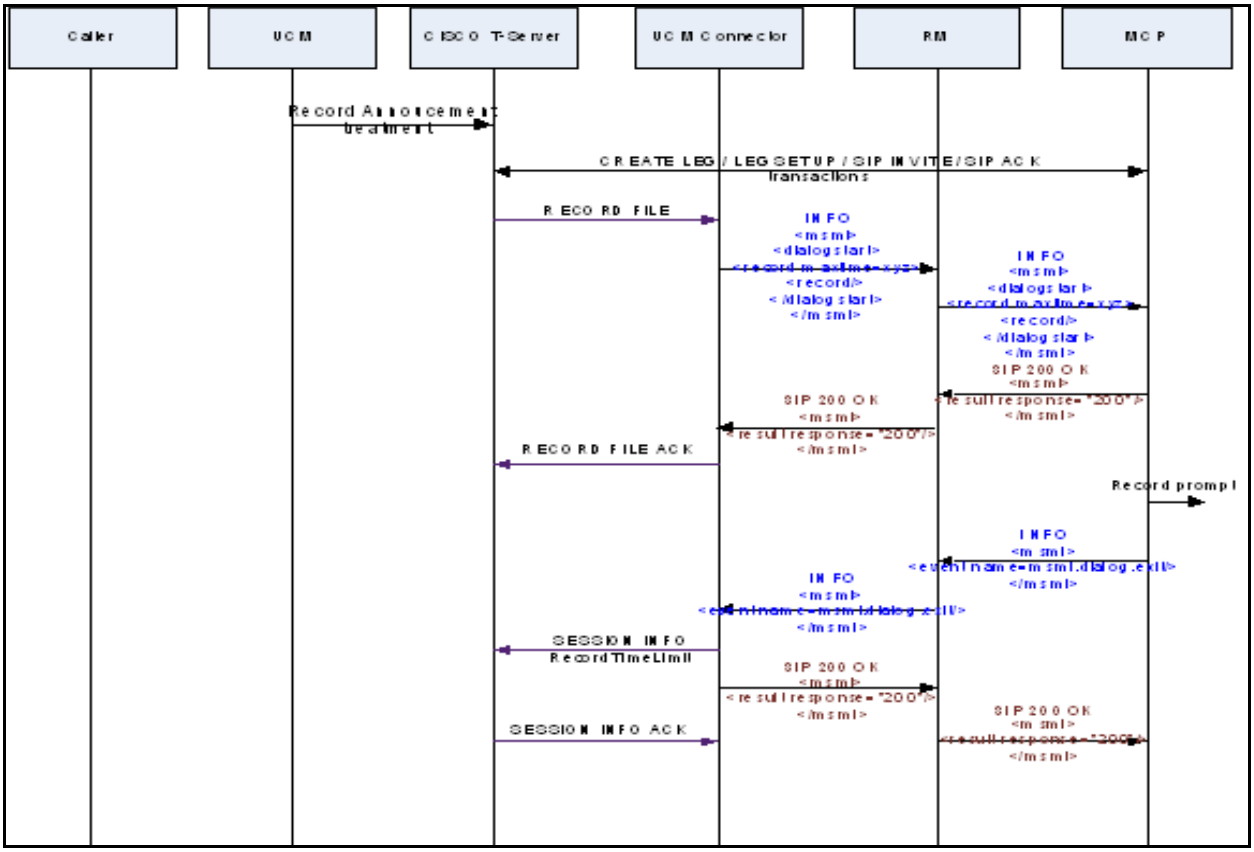

Figure 11: RECORDFILE Request—UCM Connector

Call Leg Termination

Figure 12 illustrates how the call leg is terminated.

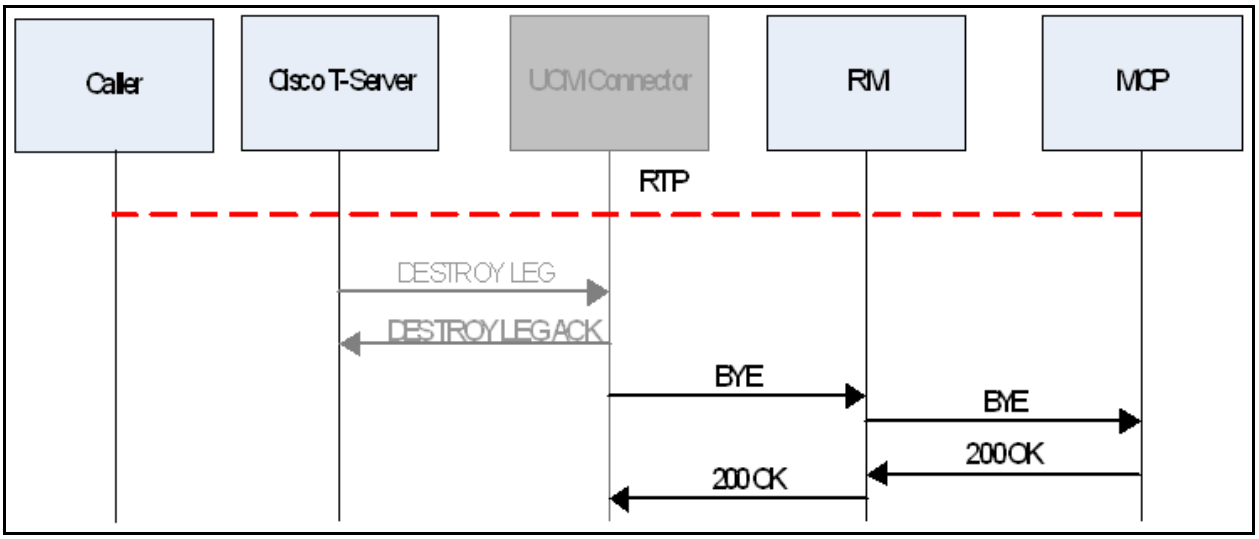

Figure 12: Call Leg Termination—UCM Connector

Genesys

Supplements

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Management Framework

- *Framework 8.1 Deployment Guide,* which provides information about configuring, installing, starting, and stopping Framework components.
- *Framework 8.1 Genesys Administrator Deployment Guide,* which provides information about installing and configuring Genesys Administrator.
- *Framework 8.1 Genesys Administrator Help,* which provides information about configuring and provisioning contact-center objects by using the Genesys Administrator.
- *Framework 8.1 Configuration Options Reference Manual,* which provides descriptions of the configuration options for Framework components.

SIP Server

- *Framework 8.1 SIP Server Deployment Guide,* which provides information about configuring and installing SIP Server.

Genesys Voice Platform

- *Genesys Voice Platform 8.5 Deployment Guide,* which provides information about installing and configuring Genesys Voice Platform (GVP).
- *Genesys Voice Platform 8.5 User's Guide,* which provides information about configuring, provisioning, and monitoring GVP and its components.

- *Genesys Voice Platform 8.1 Genesys VoiceXML 2.1 Reference Help,* which provides information about developing Voice Extensible Markup Language (VoiceXML) applications. It presents VoiceXML concepts, and provides examples that focus on the GVP Next Generation Interpreter (NGI) implementation of VoiceXML.
- *Genesys Voice Platform 8.1 Legacy Genesys VoiceXML 2.1 Reference Manual,* which describes the VoiceXML 2.1 language as implemented by the Legacy GVP Interpreter (GVPi) in GVP 7.6 and earlier, and which is now supported in the GVP 8.5 release.
- *Genesys Voice Platform 8.5 CCXML Reference,* which provides information about developing Call Control Extensible Markup Language (CCXML) applications for GVP.
- *Genesys Voice Platform 8.1 Application Migration Guide,* which provides detailed information about the application modifications that are required to use legacy GVP 7.6 voice and call-control applications in GVP 8.5.
- *Genesys Voice Platform 8.1 Troubleshooting Guide,* which provides troubleshooting methodology, basic troubleshooting information, and troubleshooting tools.
- *Genesys Voice Platform 8.5 SNMP and MIBs Reference,* which provides information about all of the Simple Network Management Protocol (SNMP) Management Information Bases (MIBs) and traps for GVP, including descriptions and user actions.
- *Genesys Voice Platform 8.5 Configuration Options Reference,* which replicates the metadata that is available in the Genesys provisioning GUI to provide information about all of the GVP configuration options, including descriptions, syntax, valid values, and default values.
- *Genesys Voice Platform 8.5 Metrics Reference,* which provides information about all the GVP metrics (VoiceXML and CCXML application event logs), including descriptions, format, logging level, source component, and metric ID.
- *Genesys Voice Platform 8.1 Web Services API wiki,* which describes the Web Services API that the Reporting Server supports.

Voice Platform Solution

- *Voice Platform Solution 8.1 Integration Guide,* which provides information about integrating GVP, SIP Server, and, if applicable, IVR Server.

Genesys

- Genesys Online Documentation at docs.genesyslab.com.

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms that are used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Customer Care website at http://genesyslab.com/support.

Information about supported operating systems and third-party software is available on the Genesys Customer Care website in the following documents:

Genesys Supported Operating Environment Reference Guide

Genesys Supported Media Interfaces Reference Manual

For additional system-wide planning tools and information, see the release-specific listings of System-Level Documents on the Genesys Documentation website.

Genesys product documentation is available on the:

- Genesys Customer Care website.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management.
- Genesys Documentation website.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. The following is a sample version number:

80fr_ref_06-2008_v8.0.001.00

You will need this number when you are talking with Genesys Customer Care about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, can sometimes contain minor spelling, capitalization, or grammatical errors. The text that accompanies and explains the screen captures corrects such errors, *except* when such a correction would prevent you from installing, configuring, or using the product successfully. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 27 describes and illustrates the type conventions that are used in this document.

Table 27: Type Styles

Type style	Used for	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Used also to indicate placeholder text within code samples or commands, in the special case in which angle brackets are a required part of the syntax (see the note about angle brackets on page 185).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for ...

Table 27: Type Styles (Continued)

Type style	Used for	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number that is specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W