Genesys Quality Management 8.0

Planning Guide

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. Go to www.genesyslab.com for more information.

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

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

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 75. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

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: 80gqm_planning_03-2011_8.0.480.01 v.0.01

Table of Contents

Chapter

1 Introduction

This chapter provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

- Document Purpose
- Audience
- Document Version
- Related Documents
- Conventions Used
- Expected Knowledge

Document Purpose

Before implementing an IPT Recording solution, you must understand the implications for your existing system. This planning document provides you with details of the Call Recording IPT Recording solution, and how it integrates with your environment.

This document includes an overview of Call Recording, its technical specifications, and the hardware, bandwidth and other requirements for successful call recording.

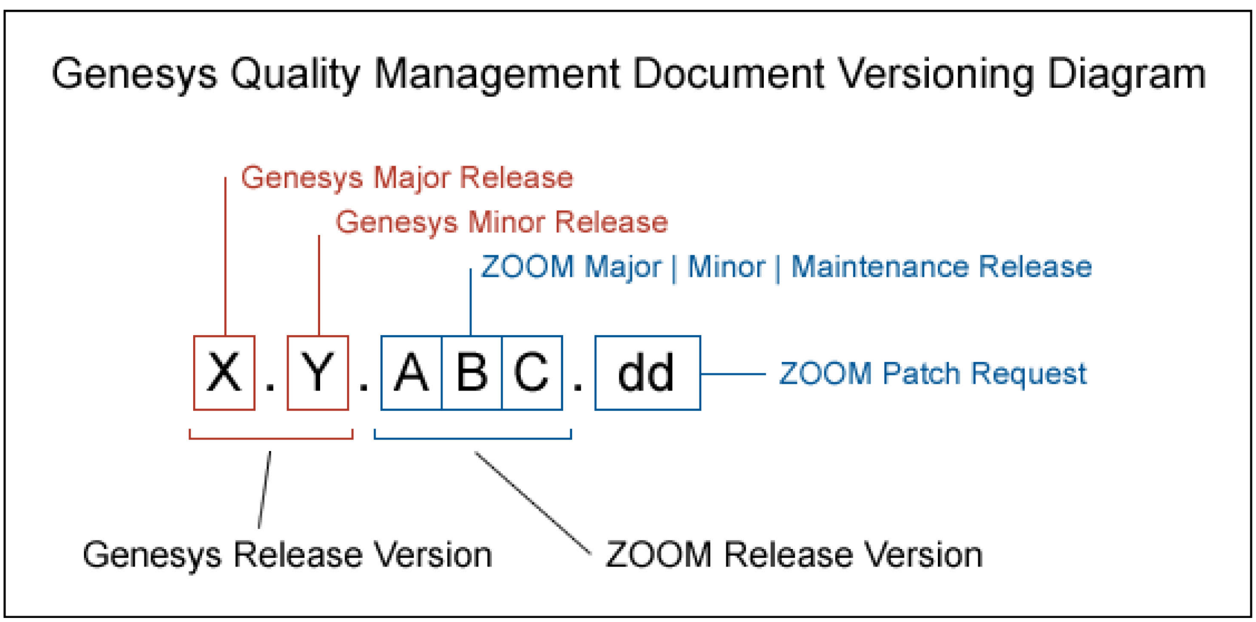
Audience

This document is targeted at Genesys Support and selected partners' technical specialists.

Document Version

The Genesys Quality Management products are provided by a partnership between Genesys and ZOOM International. The Genesys Quality Management products use a versioning format that represents a combination/joining of the versions used by these two separate entities. Although the Genesys Quality Management products and documentation use this combined versioning format, in much of the software and logs you will see the ZOOM versioning alone. You need to be aware of this, for example, when communicating with Technical Support.

The version for this document is based on the structure shown in the following diagram:



Related Documents

For other documents related to Genesys Call Recording please consult:

- *Genesys Call Recording 8.0 User Guide*
- *Genesys Quality Manager 8.0 User Guide (CC Manager, Team Leader, Agent)*
- *Genesys Call Recording 8.0 Administration Guide*
- *Genesys Screen Capture 8.0 Administration Guide*
- *Genesys Quality Manager 8.0 Administration Guide*
- *Genesys Quality Management 8.0 Installation Guide*

Conventions Used

Names of functions and buttons are in **bold**. Example: **Upload**

File names, file paths, command parameters and scripts launched from the command line are in `non-proportional font`.

```
Code is placed on gray background and bordered
```

Expected Knowledge

- Basic knowledge of Red Hat Enterprise Linux
- Unix-system administration skills
- At least intermediate network knowledge – architecture, planning etc.

Chapter

2 Call Recording Principles

The Genesys Quality Management provides you with a complete set of solutions to improve your contact center quality and performance. The suite offers interaction recording, screen capture, agent evaluation and live monitoring.

The Genesys Quality Management scales from small contact center deployments up to large distributed enterprise architectures and fully supports hosted and multitenant environments.

The information in this chapter is divided into the following topics:

General Principles of IPT Recording

Passive recording of voice calls basically means capturing and interpreting of a telephony signaling protocol, interpreting call events and based on them capturing the voice traffic as it flows between the IP endpoints. No direct interaction with a PBX or endpoints is necessary and thus no special support on the PBX side is required.

Active detection of calls requires interconnection with a softswitch, by utilizing a CTI protocol. Direct support on the softswitch is required to provide call events and to mirror voice media. Some platforms implement voice media forking at the phone or PBX level, some work as a media proxy.

IPT Recording Challenges

Detecting Calls

First, the call must be detected. This requires identifying the signaling protocol so the system can set the rest of the parameters of the call.

There are two main approaches in detection of calls – **passive** and **active** detection. The passive method is like classic wiretapping (or sniffing) – recording software attempts to identify the signaling protocol and the call data. **Active detection** makes use of direct communication with the softswitch using an available CTI protocol/API. The softswitch sends a report directly to the recording software each time a call is initiated, along with additional information based on the users' needs.

Both methods are influenced by the usual network problems– loss of packets and other errors in communication – but the danger of lost packets can be nearly eliminated by using well configured active recording.

Passive vs. Active Recording

Passive recording	Active recording
+ easy to implement	+ nearly 100% calls recorded
+ no influence on softswitch or IP phone functionality	+ lower HW and SW requirements on detection part of recording software
+ no additional functionality from softswitches needed	+ additional information captured
+ could be more protocol independent (SIP support etc.)	+ selective packet processing as main pro in wide networks
- minimally 0.5% calls missed, based on condition of network	- every monitored softswitch (or device) has to be manually added into partnership

In both Passive and Active Recording, the system must:

- Identify signalization packets
- Correctly interpret signaling commands
- Interoperate with softswitches to identify individual calls, or analyze all network traffic

Capturing Calls

In passive recording, after successful detection of call initiation, the system must capture every packet of the chosen IPT communication. These packets are taken directly from the network flow based on the call's header (sender, recipient and type of traffic). Header information is detected using a promiscuous network adapter on the recording server or through traffic mirroring by a SPAN port on a network switch.

Since IPT communication occurs in real-time, the capturing of packets has to be real-time as well. The captured packets are stored temporally and processed after the call ends.

The main problem is independence of UDP packets used for transmission of data—every packet can utilize a different path from sender to recipient, so they may arrive out of sequence. So in very wide networks, the passive recording system must monitor every possible route of the packets.

Active recording captures the call data and call stream through direct connection with the softswitch. This means that the signalization information cannot be lost, and you achieve nearly 100% capture reliability. In addition, other call data contained on the softswitch can be captured and stored in Call Recording. Of course, only IP phones that have been configured properly by the administrator can be captured through active recording.

Reconstructing Audio Streams

When a call ends, all relevant packets are stored at a temporary server location. Each packet contains its Real Time Transport Protocol (RTP) header time stamp. Recording software is able to rearrange packets in the proper sequence based on these stamps, and then reconstruct two independent data streams.

These streams are decoded by the proper codec and then again stored at a temporary server location. The decoding process requires lots of computing power (in case of many simultaneous calls), but does not have to be in real time.

In passive recording, some packets in the data stream will inevitably be lost, causing gaps in the data streams. While the human ear won't hear these small pauses, the recording software recognizes them.

In active recording, the recorder is one of the end points of the stream. Therefore packets are not lost in active recording.

Some codecs provide Packet Loss Concealment (PLC) which eliminates gaps in recording. Call Recording supports these algorithms.

Synchronizing Audio Streams

After capturing all relevant packets on a temporary server and placing them back in original order the system creates at least two independent voice streams. The streams may have variable lengths because of differing caller and called party start/end times. Therefore the streams have to be aligned, to ensure the conversation is synchronized properly.

Since the RTP protocol is real time, it does not contain external synchronization information. Therefore Call Recording must synchronize the two calls based on timestamps.

Servers

Call Recording is a platform independent application and supports any Intel and AMD based servers. Call Recording requires at least two network interface cards for recording. Other requirements vary depending on network design, planned load and especially on the count of simultaneously recorded calls.

As mentioned, servers used for recording must have at least two NIC cards. The first NIC connects with the management interface. The recorder server must have a SPAN port connected to the second NIC. The SPAN port must provide all the RTP packets related to the calls being made. If the data are not available, the system will show that the call was made, but there won't be any data to replay.

Recording Formats

Call Recording uses two different formats for storing recorded calls in permanent storage. You will be able to choose between the MP3 codec (with constant bit rate), and uncompressed WAV. Call Recording records every call as a standalone stereo audio file.

MP3

The default storage format for recorded calls in Call Recording is MP3 with constant bitrate (default value is 24 kbps, which consumes 180 kB of disk space per minute). You can choose between 8, 16, 24 and 32 kbps. The best results are attained by choosing bitrates of 24 or 32 kbps, which represent the best ratio between quality and storage demands.

WAV

Call Recording also supports uncompressed WAV (PCM codec) for storing recorded calls. Use WAV for special purposes; normally it's not optimal as a final output format due to its greater demand for storage space.

Conference Calls and Call Couples

When you plan for call recording capacity, you must take account of how many conferences and transferred calls you anticipate. This will increase the overall load on your system, and should be reflected in your Busy Hour Call Completion (BHCC) total.

Because every call is stored in stereo format (e.g. just two channels) and every caller has their own channel, conference calls cannot be stored in a single audio file. In Call Recording every call is divided into **couples** and each couple represents a two-sided conversation. A normal call has only one couple so recording produces only one stereo file.

In a conference call there is at least one couple for every pair of participants – in a conference call with four callers there are six couples and the resulting record will consist of six audio files. All these calls are mixed through the Conference Bridge, which can be either software or hardware based.

When a regular call becomes a conference call, each participant is considered to be in a separate conversation with the conference bridge. So a three way conference call would have at least five recorded streams, and potentially more.

For more information on how call flow configurations result in various call couples, see Appendix B.

Chapter

3 Call Recording and Genesys

Having looked at the general recording principles and challenges, we will look at how the Call Recording product integrates with the Genesys platform and a few of the recording methods available.

Call Recording supports passive SIP and RTP monitoring and capturing. If Genesys is deployed on top of Cisco Unified Communications Manager there is also direct integration with Cisco UCM. The result in both deployments is the same: we can capture all the communication, provide all of the calls and additional metadata including attached data. The information in this chapter is divided into the following topics:

- Recording in the Genesys environment
- Call metadata
- Protocols and interfaces

Recording in the Genesys Environment

Call Recording works through capturing SIP signaling protocol and RTP streams and integrating the Genesys T-Server and Configuration Server to get additional data about the calls.

Call Recording also supports direct support of Cisco Unified Communications Manager as a 3rd Party PBX under the Genesys Customer Interaction Management Platform.

Genesys also supports recording via Stream manager to create WAV files which can then be obtained by a 3rd party, but Call Recording does not support this method as it severely limits product functionality such as live monitoring and provides no guarantee that the call will be recorded.

Another approach is in development: Media Stream Replication which will provide a key feature of media forking.

Genesys CIM with SIP Server

The first figure illustrates the current recording method on Genesys SIP server, using SIP signaling or sniffing. Call Recording server captures the SIP signaling protocol and interprets information about running calls; the terminal addresses and captures the RTP streams. Through the T-Library, Call Recording integrates with the SIP T-Server to get data about agents, their interactions and attached data.

This method is highly dependent on the network infrastructure.

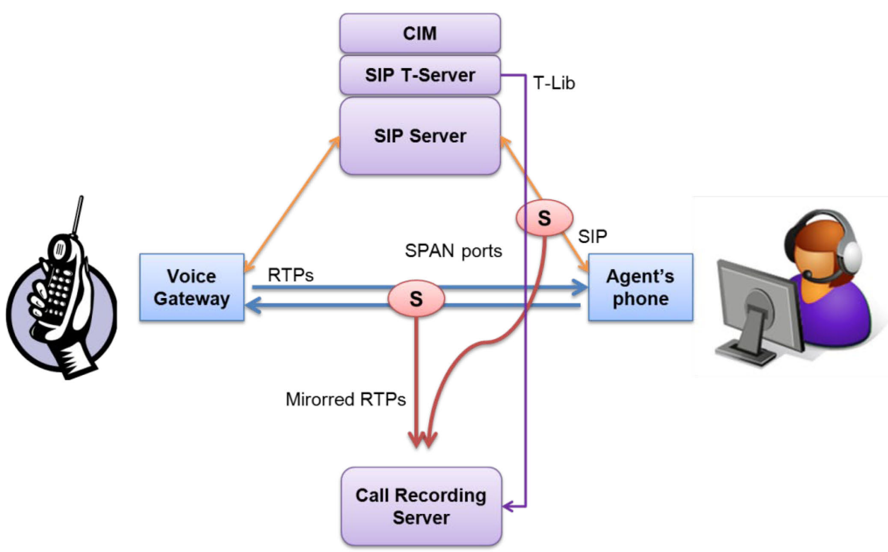


Figure 1: Genesys CIM with SIP Server

Genesys CIM with Cisco UCM

Another approach is the recording of Genesys with Cisco UCM. In this case we have the Genesys Customer Interaction Management Platform, CIM, where Cisco Communications Manager acts as the PBX. In this scenario, the Cisco T-Server is also performing integration. In addition to connecting to Cisco UCM, Call Recording directly communicates with the Cisco T-Server to get the call related data not available from the Cisco platform.

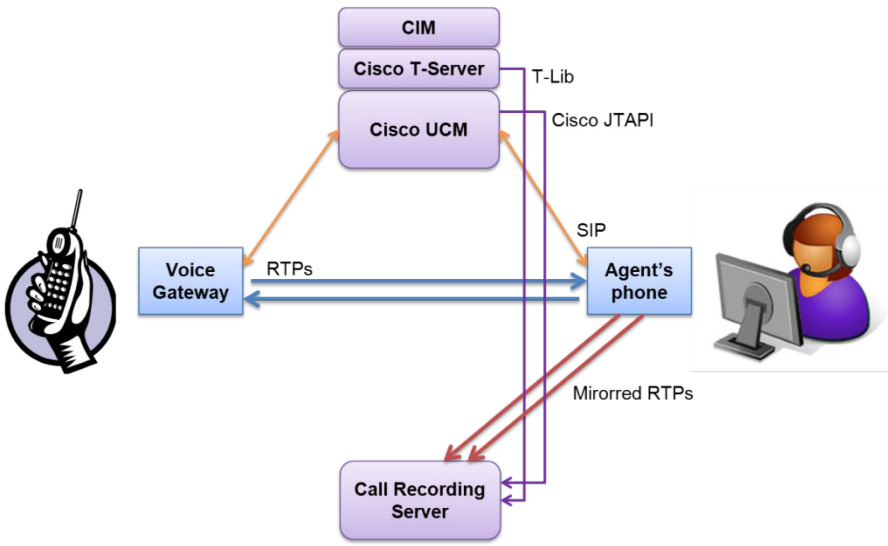


Figure 2: Genesys CIM with Cisco UCM

SIP Server Recording with Stream Manager

The next figure describing SIP server recording with Stream Manager is for illustrative purposes only, as this method is not supported in Call Recording.
Stream Manager acts as the RTP proxy. When there is the request to record it, then both end points are re-invited by Stream Manager, which captures the voice content and creates the WAV files.

This means there is no access to the voice content in real time and therefore it is not possible to use Live Monitor, pre-recording or even guarantee the functionality of recording calls.

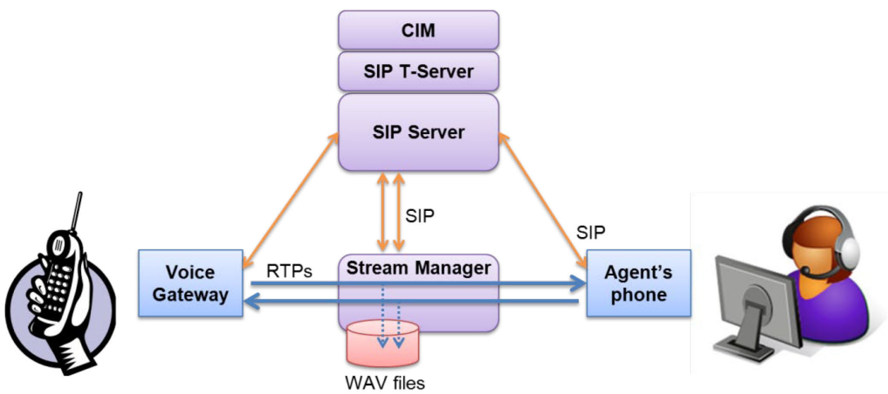


Figure 3: SIP Server recording with Stream Manager

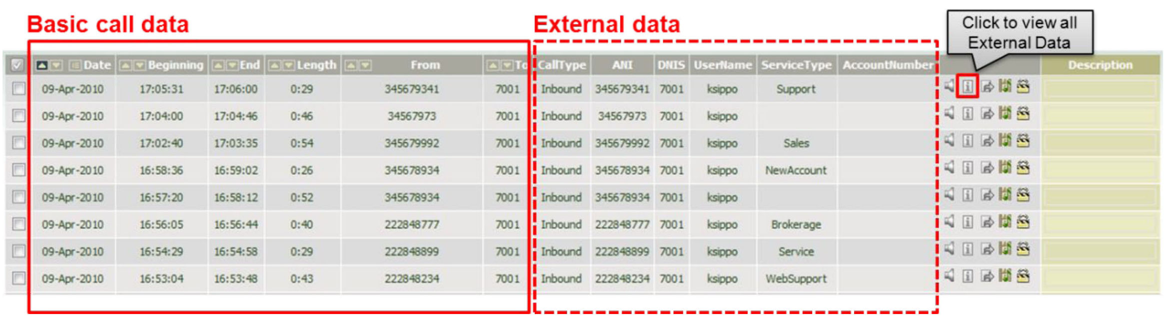
Call Metadata

The internal Call Recording database stores call metadata as two different entities:

- Basic call data
- External data

Basic call data exists for all recordings, regardless of the underlying PBX, contact center platform or 3rd party integration. The structure of data is always the same and basically represents only extension numbers, data and time and internal flags.

External Data contains platform specific information stored as key/value pairs. Virtually any number of external data can be stored with every call; however a large amount of data may have negative impact on database performance and thus on the speed of searching, sorting or any other tools handling call metadata.



Date	Beginning	End	Length	From	To	CallType	ANI	DNIS	UserName	ServiceType	AccountNumber	Description
09-Apr-2010	17:05:31	17:06:00	0:29	345679341	7001	Inbound	345679341	7001	ksippo	Support		
09-Apr-2010	17:04:00	17:04:46	0:46	34567973	7001	Inbound	34567973	7001	ksippo			
09-Apr-2010	17:02:40	17:03:35	0:54	345679992	7001	Inbound	345679992	7001	ksippo	Sales		
09-Apr-2010	16:58:36	16:59:02	0:26	345678934	7001	Inbound	345678934	7001	ksippo	NewAccount		
09-Apr-2010	16:57:20	16:58:12	0:52	345678934	7001	Inbound	345678934	7001	ksippo			
09-Apr-2010	16:56:05	16:56:44	0:40	222848777	7001	Inbound	222848777	7001	ksippo	Brokerage		
09-Apr-2010	16:54:29	16:54:58	0:29	222848899	7001	Inbound	222848899	7001	ksippo	Service		
09-Apr-2010	16:53:04	16:53:48	0:43	222848234	7001	Inbound	222848234	7001	ksippo	WebSupport		

Figure 4: Call Metadata

In general, all the External Data have different prefixes, which identify the origins of the data.

SIP protocol properties – information extracted from SIP signaling such as IP addresses and ports of both end points, SIP call ID or SDP details – all are stored with the SIP_ prefix.

T-Server properties are prefixed with GEN_TEV_ (T-Server Event), since the information gets extracted from various T-Server events. The properties available are already pre-configured. It is possible to configure which T-Server properties you can get, but it is highly recommended to keep the default settings. The majority of users find that the default values are more than sufficient.

If changes need to be made to the range of available properties, this must be done via direct access to the appropriate configuration file; this option is not configurable from the Call Recording GUI.

The configuration data from the **Configuration Server**, such as Agent Name and Employee ID, are also available, representing the static configuration of users. The difference between configuration data and T-Server properties is that the T-Server properties come from events, while configuration data is cached. The configuration server is not queried with every new event, but its contents are refreshed regularly, e.g. every 30 minutes. This can be configured during installation and also later via the graphical user interface (GUI).

Configuration data is cached and passed to every recorded call. Configuration Server data are prefixed with GEN_CFG_.

Attached Data or **User Data** is the last group of metadata. This also comes from T-Server events and is updated during the entire call. The list of User Data which should be attached to recorded calls must be configured in the Call Recording GUI; by default no data are attached. This is deliberate, since with every call there may be tens of fields of attached data, making it inefficient to store all of them by default.

User Data are saved with the GEN_USR_ prefix. The name of the field can be customized but will always have this prefix.

Protocols and Interfaces

As already mentioned, calls are recorded based on SIP signaling, which is obtained from the network switch by mirroring SPAN ports. The signaling information comes to the protocol adapter, which interprets the protocol and passes the events to Recording Core. From the SIP Server point of view, Call Recording is entirely passive.

Call Recording then interfaces with the T-Server and Configuration Server, which is accomplished by a component called the Genesys Integration Module (GIM). There are two different components on the Call Recording side which interface with the Genesys CIM platform.

The platform SDK is used as the interface for connecting to T-Server and Configuration Server, namely the Voice Platform SDK and Configuration Platform SDK.

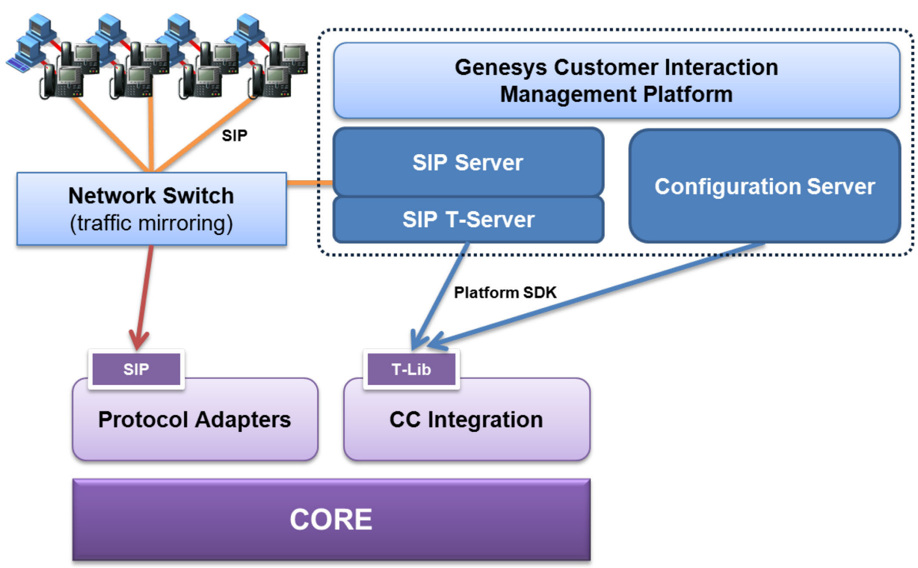


Figure 5: Protocols and Interfaces

Chapter

4 Call Recording Architecture

To properly design and configure Call Recording it is very important to understand its internal structure, modules, design and communication. This chapter focuses on the internal architecture of Call Recording as well as on the interfaces and possibility of third party integration.
The two related important topics are discussed as well – Media Lifecycle Management and Recording rules.

The information in this chapter is divided into the following topics:

- System architecture
- Third party integration
- Media Lifecycle Management
- Recording Rules

System Architecture

Call Recording has a multi-level module architecture that supports scaling for optimizing functionality in any environment.

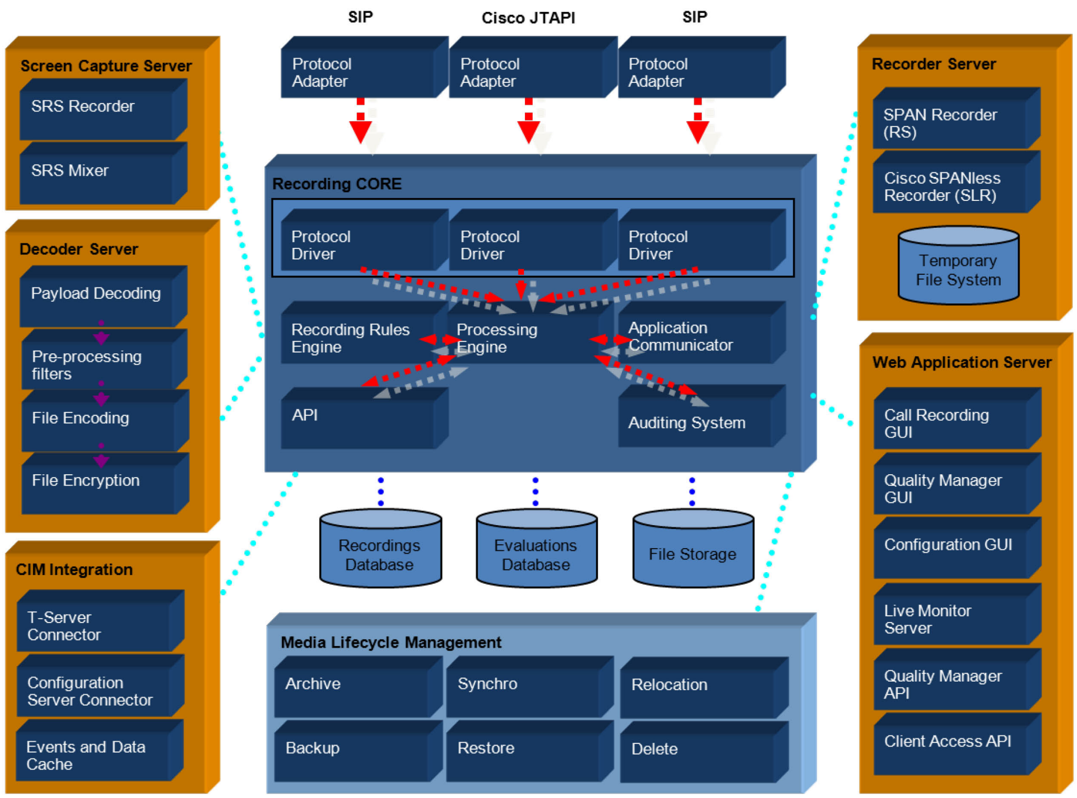


Figure 6: System architecture

Recording Core

Recording Core is the main "brain" of the solution. It is a finite automaton that receives information about signalization from different **Protocol Adapters** through its **Protocol Drivers**, managing **Recorder server**(s), **Decoder server**(s), communicates with the **Database**, **Temporary** and **Permanent File Systems**, and provides powerful **API** for third party applications**, Auditing System** and **SNMP** for monitoring.

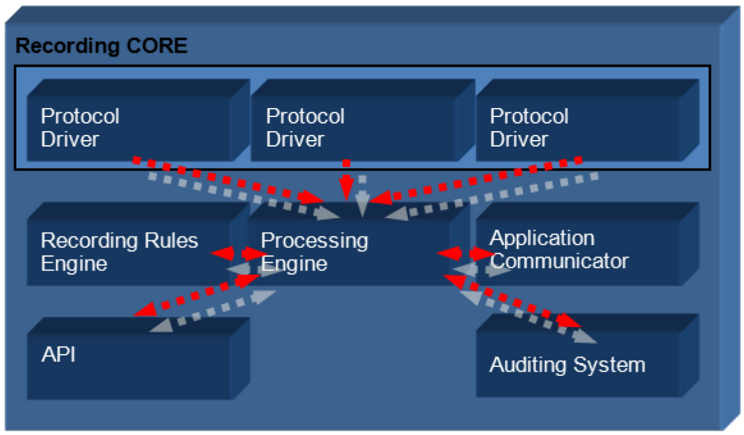


Figure 7: Recording Core

Recording Core includes an **Auditing System** which creates a log about all the actions that occurred in Call Recording (e.g. call starts, couple starts, RTP begin / end**, Recorder Server** asked to begin / end recording, **Decoder Server** asked for decoding, Database insertions, user login, call replay / export / deletion and many more). This supports installation debugging as well as auditing user interaction with the system.

Recording Core also fully supports SNMP for remote monitoring. **Recording Core** communicates with each component and periodically, or upon request, shows the status of all components attached. If any component fails or reports warnings or problems**, Recording Core** can report this by SNMP for further processing.

Protocol Adapters

Protocol Adapters provide **Recording Core** with the necessary signalization of phones so that **Recording Core** can determine the next actions required (start recording, stop recording etc.).

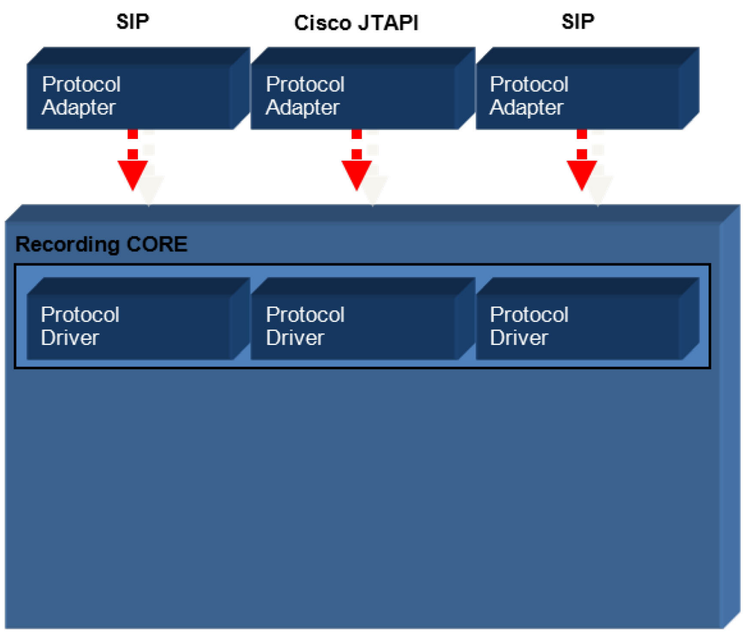


Figure 8: Protocol Adapters

Recording Core is written to fully support IP telephony recording without any hard coded signalization protocol. This allows support of new protocols as they are being introduced to IP telephony. Call Recording 8.0.480 supports 3 protocols for receiving signalization: SIP (as an industry standard protocol) and Cisco specific protocols—Cisco JTAPI and Cisco SCCP (the latter not supported in the Genesys environment, since it does not allow full integration).

The role of each **Protocol Adapter** is to translate any signalization into standard messages for the **Recording Core**, informing it about call establishment, RTP Streams begin / end, transfers, conferences, on-holds, barge calls etc. These unified messages allow, for example, the use of Call Recording for the recording of one group of phones by SIP adapter and a second group of phones by Cisco JTAPI, all as a single Call Recording system.

Multiple instances of the same **Protocol Driver** are allowed as well. This can be used for connecting by one Call Recording system to multiple softswitches in larger environments or sniffing SIP from multiple SPAN ports if remote SPAN is not supported or inconvenient to configure.

Recorder Server

There are two types of recorder server.

- SPAN Recorder Server (RS)
- SPANless Recorder Server (SLR)

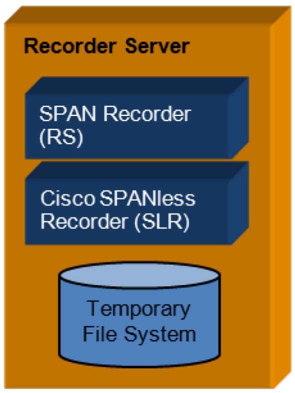


Figure 9: Recorder Server

The **SPAN Recorder Server** is fully controlled by **Recording Core** and captures specified streams and packets at a specific location (port, IP address) and sends them to temporary storage. A recorder server binds to a particular interface (like eth0 on the Linux system), which is automatically switched to promiscuous mode to receive the RTP streams. The interface has to be connected to a switch with a SPAN port configured. You can run multiple instances of the **Recorder Server** at an unlimited number of locations to support distributed IP telephony recording.

The **SPAN Recorder Server**, if requested by **Recording Core**, can also forward streams of RTP to a particular IP address and port, which is used in the live monitoring console for real-time listening to calls.

The **SPANless Recorder Server (SLR)** supports Cisco specific “Phone Media Forking” as an active recording approach. SLR communicates with both **Recording Core** and the Cisco UCM. The Core determines which calls are recorded, while the UCM provides the call data and call stream. When a call identified by **Recording Core** is detected on the SIP trunk, the call data and streams are saved as PCAP files on the recorder server.

Decoder Server

Decoder Server is used for:

- Sorting the saved RTP packets
- Decoding them from codec
- Transcoding the streams to WAV
- Creating stereo WAV out of two independent channels (calling and called party)
- Encoding the WAV audio into MP3 format

Decoder Server is always running as master with 1 or more clients (there can be a special server allocated for decoding in large environments with a lot of calls in the G.729 format).

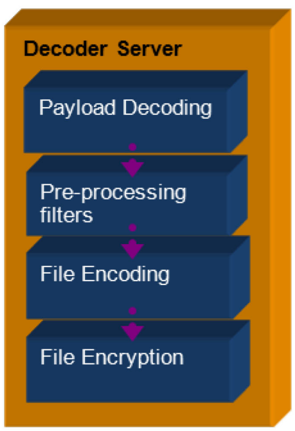


Figure 10: Decoder Server

The **Decoder Server** master keeps statistics about the decoding speed of the different registered Decoder Server slaves and intelligently manages their queues of calls to reach the maximum speed of decoding.

Key Manager

Key Manager Server is responsible for secure handling of certificates, keys, and passwords. It has the following features:

- implements a basic set of key management related features such as different key stores (PKCS12, JKS, JCEKS) and encryption algorithms (AES, DES, Blowfish)
- supports more keys with random usage (If one key of the keys gets compromised, only the corresponding part of the recordings must be re-encrypted)
- features a and re-encryption process in case the event that any of the keys gets compromised

Key Manager clients are part of Decoder Server, GUI, and APIs and provides encryption and decryption of audio and video media.

Every database entry contains the UUID of the key that was used for encryption and the MD5 (or SHA-1) hash of the encrypted file so encrypted media can be easily be verified or re-encrypted if any of the keys expires or are compromised.

Temporary File System

Unprocessed raw data captured by **Recorder Server** from the network are stored on the Temporary File System. For every recorded call the **Recorder Server** stores two files in PCAP format. PCAP files are processed right after a call ends or are cached for later processing if Pre-recording is enabled. After processing of the recorded call, temporary data are deleted. In case of decoder failure, temporary data can be processed by the repair utilities.

Permanent File System

After **Decoder Server** creates the final MP3 version of the recording, the physical file is placed on the Permanent File System and all related call information are inserted into the Database.

Database

Call Recording uses a standard SQL database (PostgreSQL), to which it connects via JDBC driver. The database stores users, groups, recording rules and information about recorded calls.

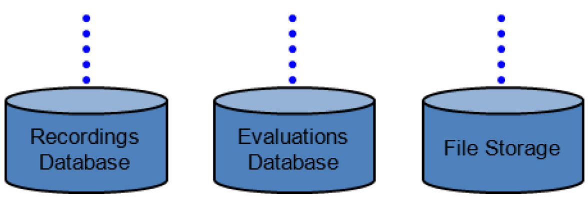


Figure 11: Database

Application Server

Call Recording has a web-based GUI which is used for both system configurations (user and group management, recording rules setup) as well as for call search & replay, live call monitoring (Live Monitor), Media lifecycle management etc. The Quality Manager GUI runs in the same application server, but in most cases as a separate application container on a dedicated server.

Figure 12: Application Server

Application Server runs on an Apache Tomcat java based application server and is communicating with **Recording Core** via its API. It also has access to both recording and evaluation databases and to file storage.

Call Recording and 3rd Party Integration

Call Recording offers three methods of interconnection, depending on planned usage and amount of control. You can use one of these methods for integration with Call Recording:

- Call Recording RMI API for low level event exchange with contact center platforms and CRM systems
- Call Recording User Access API for client access through a web application server (http/xml protocol)
- Quality Manager Web Service API enables 3rd party access to evaluations, questionnaires and user data.

Call Recording RMI API

Call Recording RMI API allows you to connect custom applications to the Call Recording platform, monitor activities and system core objects, change settings, add custom information to calls, and so on. Core operation monitoring is implemented by means of connecting an observer to specific parts of the Call Recording API that subsequently report core changes through these observers.

- Call Recording notifies applications (connected to the Call Recording RMI API) of selected events
- Through the Call Recording RMI API external applications can:
  - Access a list of registered and currently recorded calls
  - Dynamically modify recording rules
  - Request the storage of additional information about the call (agent ID, skill group, customer ID etc.)
  - Request the storage of a specific call being pre-recorded
- 3rd party applications can search and replay calls without using the standard Call Recording user interface

Call Recording User Access API

- The Call Recording User Access API allows you to integrate third party applications with Call Recording. The applications have secure access to the recording functionality of Call Recording through the http/xml protocols
- 3rd party applications can search and replay calls without using the standard Call Recording GUI
- The Web Application Server allows access of 3rd party applications using the http/xml protocol
- Protected access – authorization, https encryption
- Choice of encoding: octet-stream (default) or base64 (see example URI below)
- The service is called by the http protocol, which can be simulated by pasting the following sequence of URLs (with correct parameters) into a web browser address bar:

```
http://localhost:8080/Call
Recording/loginservlet?loginname=admin&password=admin
(this will provide the session id)

http://localhost:8080/Call
Recording/audiodata;jsessionid=232B0EFFFC198B53C520F88F67442396?isCoupl
e=1&externalData=CiscoID$!$1&typeQuery=externalData&encoding=base64
(this will return calls with a Couple Id)

http://localhost:8080/Call
Recording/audiodata;jsessionid=14D1B2F3AE7671B79CB1315C8760BEAB?isCoupl
e=1&dbId=7&type=1&encoding=base64
(returns the file in base64 format)
```

Quality Manager Web Service API

Quality Manager Web Service is an API that allows 3rd party developers and system integrators to access Quality Manager evaluations, questionnaires and user data, and then export the results to other systems and applications.

Quality Manager Web Service provides a set of data which can be used by other business applications such as Workforce Management Platforms, Business Intelligence Applications, and so on.

Capabilities

- Provide evaluation results from Quality Manager by Web Service interface
- Provide questionnaire data via Web Service
- Provide Quality Manager user and agent data via Web Service

Media Lifecycle Management

The Media Lifecycle Management tools offer archiving, backup and restore, deletion and replication of calls with many configurable options.

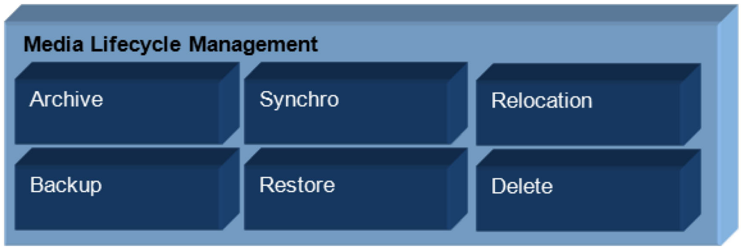


Figure 13: Media Lifecycle Management

Call Recording has a built-in module for archive and backup. It supports any shared file systems accessible from a Linux server: Windows share (SMB protocol), UNIX share (NFS protocol) and Apple share (AppleTalk protocol) or directly accessible storage by a SCSI interface or FC card installed on the server. Also direct integration with IBM Tivoli Storage Manager software is available as a set of ready to use scripts. By integrating with TSM, Call Recording can save or restore archive packages directly to/from a TSM server.

Media Lifecycle Management is fully configurable. Different storage and export policies can be applied based on certain criteria, for example: incoming numbers, called party, call length, notes attached, etc.

Media Lifecycle Management Example:

1. Leave calls on the Call Recording server for three months.
2. After three months, move them to a shared file system, where they remain accessible online for the next nine months.
3. After nine months, export them for storage on an encrypted DVD and delete them from the system. The encrypted calls can be accessed when necessary from the DVD.

Export can automatically create “packages” that contain directories with exported MP3 files and an HTML index of the calls with direct links to the files together with a CSV export of call related data for additional processing.

Full call details can be saved in the Call Recording database to enable complete searches using the GUI, even after the calls have been archived and deleted.

Reports on archive and backup results, detailing available space and the expected number of days to fill the storage space available, can be configured to trigger corresponding warnings and alarms.

Recording Rules

Recording Rules determine which calls are recorded. Basic recording rules are based on the phone number(s) or on the IP address of the phone. Additionally, rules can be created using any other available metadata. For example, to support free seating in call centers, Call Recording also offers the option to set up recording rules based on Agent ID as obtained from the T-Server and stored in Call Recording external data.

Note: At least one recording rule has to be defined, otherwise no calls will be recorded.

Recording rules are always associated with groups of users, and identify which calls to record for those users. Where there are multiple user groups, their hierarchy also applies to recording rules, so recording rules associated with higher privileged users take precedence over recording rules for users with lower privileges.

Using Wild Cards

Recording rules can be set for a single number as well as for a range of numbers. Different wild cards are available when creating recording rules.

- **Setting the range**: 200? selects the numbers from 2000 to 2009; 20?? selects the numbers from 2000 to 2099, etc.
- **Setting all numbers**: “2*” selects all phone numbers which start with the number 2. “*2” means to record all phone numbers which end with the number 2.
- **Incoming and outgoing**: ”>“ sets the range for specifying incoming or outgoing phone calls. For example: 2005> selects all calls made from the number 2005, whereas >2005 selects all calls which were made to number 2005.
- **From-To**: ”=” specifies calls made between two phone numbers. For example 2005=3000 means calls made between 2005 and 3000.

Note: Wild cards can be combined. For example 20**??>** selects all outgoing calls from numbers 2000 to 2099.

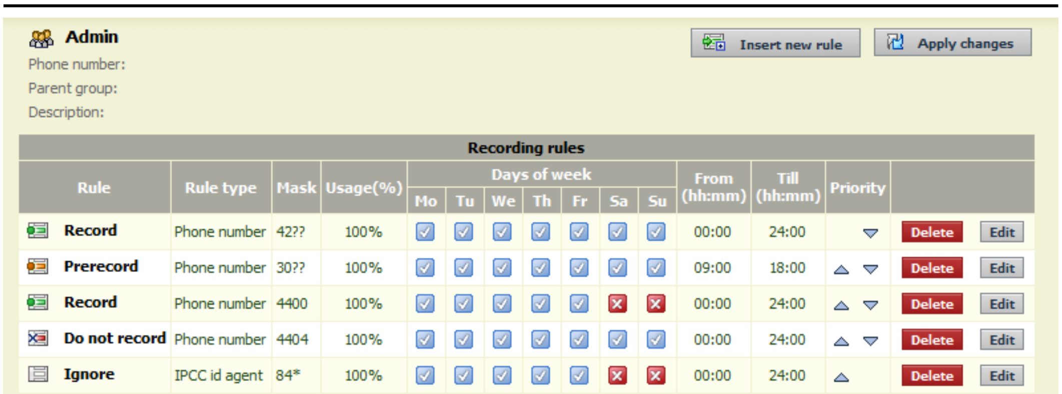


Figure 14: Using Wild Cards

Types of Recording Rules

- **Record**: The system records incoming and outgoing calls from the specified number, range of phone numbers or IP addresses.
- **Do not record**: The system does not record any calls to or from the specified number, range of phone numbers or IP addresses.
- **Pre-record**: The system records the calls, but does not save the recording unless the user requests it.
- **Ignore**: The system stops processing recording rules that have lower priority and are assigned to the same group. It continues checking the following groups. This rule is superior to any other rule.

External Data Recording Rules

Recording rules can be based on any value available in external data. In such a case the system waits until a certain external data key/value pair is available and then makes a decision. The call is always being recorded until a decision is made; if such a call should not be recorded, recording stops immediately and all captured data are discarded.

Note: Using External Data Recording rules may impact Call Recording system performance. A number of calls may be temporarily recorded before the system can make an informed decision!.

Rules Order

Recording rules are applied from top to bottom. Order the recording rules so that the "**No**" rules (denial of permission) are positioned above the "**Yes**" rules (granting permission). To move rules up or down, use the up and down arrow buttons ▲▼.

Hierarchical Recording Rules

Recording rules can be defined in every Call Recording group, and groups are arranged in a hierarchy. Higher group recording rules affect subordinate groups.

Random recording

Another option Call Recording provides is the ability to record a random selection of calls. While setting up recording rules, you can set the percentage of calls to be randomly recorded.

Random recording can be also independently configured for Screen Capture. Call Recording can for example record all the voice interactions, but only some percentage of agent desktop screen interactions will be captured.

Chapter

5 Call Recording Technical Specification

The information in this chapter is divided into the following topics:

- Supported standards
- Supported Operating Systems
- Hardware and software requirements

Supported Standards

Call Recording records the following models of interconnection:

- IP phone to IP phone.
- IP phone to Voice GW.
- IP phone to CTI port.

Voice GW to CTI port (On a specific CTI port there usually reside specialized IPT applications like IVR (Interactive Voice Response) or Voice Mail; CTI port is supported only with trunk side recording).

The decoder module is able to process most modern codecs; G.722 (PCL), G.711 (PCL) and G.729 (including ‘a’ and ‘b’ annexes).

Supported Operating Systems

All modules of Call Recording are Linux-based and interoperate with **Red Hat Enterprise Linux**.

Linux was chosen for its high stability for demanding call recording, easy remote administration, superior uptime without restarts and no hidden additional costs. The Red Hat Enterprise Linux distribution includes everything that Call Recording requires (e.g. database server, management tools etc.) and supports a wide range of server hardware (Intel as well as AMD based servers).

Red Hat Enterprise Linux as a commercial Linux distribution offers operating system support from Red Hat with pricing dependent on a Service Level Agreement.

Hardware and Software Requirements

Hardware

Call Recording supports both processor platforms used in common servers – AMD and Intel based servers. The hardware requirements depend on planned load and architecture. Although there are no minimum requirements, the following minimal hardware configuration is recommended:

Single site installation (all modules on single server):

- Processor: Pentium IV or equivalent, 2GHz
- RAM: 2 GB
- Storage: 60 GB (minimum 25 GB), RAID 5+1 recommended
- Network: 2x 1Gbit NIC

Multi server installation (dedicated servers for some modules):

- Processor: Pentium IV or equivalent, 1GHz
- RAM: 3 GB
- Storage: 60 GB (minimally 25 GB), RAID 5+1 recommended
- Network: 2x 1Gbit NIC

Notes

- If Quality Manager 8.0.470.00 will be installed on the same server, a minimum of 4GB RAM is required.
- Pentium III processors are not compatible or supported with Genesys Quality Management (8.0.460.00 and higher).

Important:

When you are planning to use file systems that will later be upgraded to larger than 2TB, it is necessary to use GPT/EFI partitions. This prevents losing data during an upgrade. It is not possible to migrate between two types of partitions without recreating your file system.

Software

Server

The Quality Management installation CD / ISO requires a customer-installed instance of the Red Hat Enterprise Linux operating system. No further server-side software is required for a standard installation.

User Access

Web Browser

Both the main Call Recording and Quality Manager products are heavily reliant on a standard web browser-based user interface for most administration tasks, data management and all user-controlled features.

The following web browsers are supported on the Microsoft Windows and Apple Mac OS desktop platforms:

- Microsoft Internet Explorer 7+
- Mozilla Firefox 1.5+
- Apple Safari 3+
- Google Chrome 4+
- Opera 9+

Note: Genesys Quality Management versions above 8.0.460 no longer support the Microsoft Internet Explorer 6 browser. If other supported browsers are not available, the Google Chrome Frame plug-in for this browser may be a workable solution.

- Web browser requires media player plug-in (Windows Media Player 9+ on the Windows platform, QuickTime on the Mac) for audio and video media review.
- Web browser requires Adobe Flash Player plug-in for rendering Reports.

Desktop Client

If the Screen Capture screen recording feature is required, a 4MB client installer application (available on the product CD and downloadable from the installed server product) must be run on each agent desktop PC. This Screen Capture Client software is currently available for Windows-based PCs only.

No other third party software is required for Screen Capture recording functionality.

Chapter

6 Integrating Call Recording with Your Network

Call Recording applications are always dependent on network architecture and very sensitive to all configuration and topology changes. This chapter describes various dependencies and important configuration procedures which help you to properly integrate Call Recording with your network infrastructure.

The information in this chapter is divided into the following topics:

- Call Recording interconnection with LAN
- Configuration procedures for direct integration with Cisco UCM
- SIP traffic filtering

Call Recording Interconnection with LAN

Call Recording as a single server implementation, as well as in scaled environments, requires network connections through two or **more Network Interface Cards** (NIC). The primary NIC is used for management and the rest of the available NICs are used for recording and wiretapping (in the case of a passive recording model) of network traffic. The number of required NICs depends on the architecture.

A Switched Port Analyzer (SPAN) mirrors defined network traffic and sends it directly to the second (and next) NIC on the Call Recording server. There are three variants of SPAN; basic SPAN, remote SPAN and VLAN SPAN:

- **Basic SPAN** - The Switched Port Analyzer (SPAN) feature, sometimes called port mirroring or port monitoring, sends copies of packets matching defined criteria on a selected device—for example a Network Probe or Call Recording server. With SPAN configuration every configured switch must be connected directly to the Call Recording server.
- **Remote SPAN** - RSPAN has all the features of SPAN, plus support for source ports and destination ports that are distributed across multiple switches, allowing you to monitor any destination port located on the RSPAN VLAN. You can monitor the traffic on one switch using a device on another switch.
- **VLAN SPAN** - With VLAN-based SPAN (VSPAN), the user can choose to monitor traffic on all the ports belonging to a particular VLAN. This is the setup most commonly used by Call Recording.

RSPAN and VSPAN technologies depend on the version of the switch’s operating system and it is recommended to use the same version of CatOS/IOS (if Cisco Catalyst switches are used) for every switch with this feature enabled.

Note: For direct integration with Cisco UCM and recording based on Cisco JTAPI and SPANless, Call Recording does not require these settings since it connects directly with the Cisco UCM.

Configuration Procedures for Direct Integration with Cisco UCM

For direct integration with Cisco UCM and recording based on Cisco JTAPI adapter, Call Recording requires a user account on Cisco UC Manager with associated phone numbers configured. Only associated phone numbers in the specified network will be monitored and every new or moved phone must be added or updated in the Call Recording database. Normally, with a good numbering plan and JTAPI interconnection, system administrators only add or update numbers within the UC Manager – there are no changes in Call Recording.

For a Device Based (SPANless) call recording approach, the following configuration steps are also required:

- Turn on IP phone BIB (Built-In Bridge) to allow monitoring or recording.
    1. Add a user for the monitoring or recording application.
    2. Add the user to groups that allow monitoring and recording.
    3. Configure tones for monitoring or recording.
    4. Configure DNS for a monitoring calling search space.
    5. Enable recording for a line appearance.
    6. Create a recording profile.
    7. Create a SIP trunk that points to the recorder.
    8. Create a route pattern for the recorder.
    9. Configure recorder redundancy.

For a detailed configuration procedure please refer to Cisco Unified Communications Manager Features and Services Guide, chapter 29 – Monitoring and Recording.

Note: The current implementation of SPANless call recording requires setting up the **Automatic Call Recording Enabled** option for every recorded line. Selective recording is supported, but it is handled completely on the Call Recording side.

During selective recording Cisco UCM is forced to start a forked call, then Call Recording accepts it and in the case that the call is NOT to be recorded, it is terminated. This may cause a higher load than for UC Manager-driven selective recording, but the impact on the system is not significant.

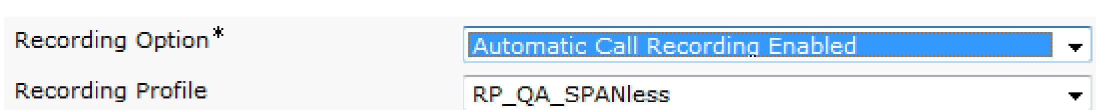


Figure 15: Configuration procedures for direct integration with Cisco UCM

SIP Traffic Filtering

For every call, there are two signaling legs. SIP Server is the main component that controls telephony signaling and the switching is connected by SIP trunk to a voice gateway. This signaling leg is called Trunk side. The second signaling leg is between the SIP Server and the agents. This signaling leg is called Agent side.

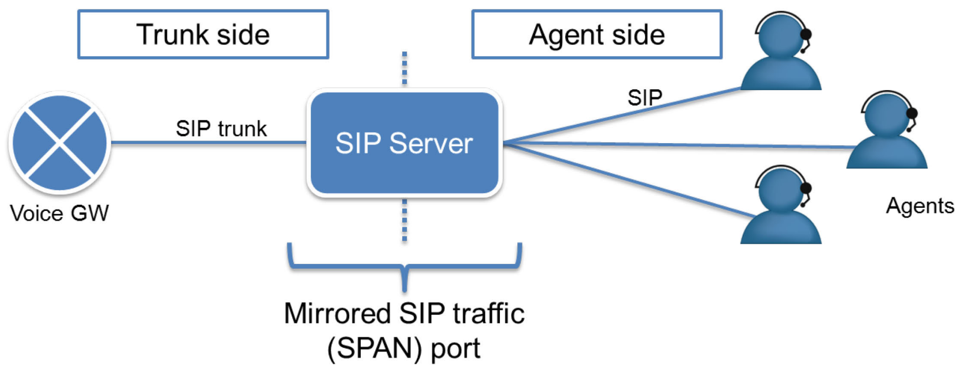


Figure 16: SIP traffic filtering

When there is a call coming through the voice gateway to an agent, the VG sends the signaling information to the SIP Server with the protocol handshakes. Together with other components, SIP server decides the call will be transferred to a specific agent. There is another signaling between the SIP server and the agent. Both sides represent one call, but this signaling leg carries slightly different information than the first.

In most cases the SPAN port mirrors both signaling legs – agent and trunk side. Call Recording is not always able to match both legs of one call and properly identify the whole signaling path. It may happen that only one signaling leg is analyzed and used as a source of call events. Therefore captured recording and associated metadata may not provide the expected results.

The filtering of the unwanted signaling leg solves this issue and ensures that the system records only (and all) desired interactions.

To record the exact interaction we want, we are usually only concerned with the Agent side. The **Agent side** signaling leg carries the most important call events, while it also enables proper detection of all transfers, conferences and other operations.

Filtering must be configured at the highest level, which is on the SIP adapter, which gets all the traffic from the SPAN port. Different filters may apply for different network connections as more SIP adapters can process data from more network sources.

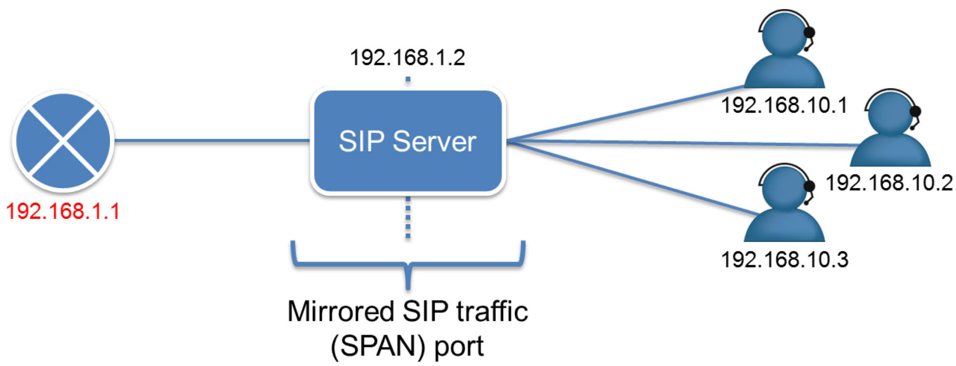


Figure 17: SIP traffic filtering

This configuration is performed in the `callrec.conf` file, which must be modified at the OS command line.

There are two basic options; you can either exclude unwanted IP addresses or whole network subnets (representing voice gateways or other border devices), or include agent network subnets or particular addresses. The combination of both of these is also supported.

The SIP Adapter filter mechanism uses the same format as popular network analyzing tools such as Tcpdump or Wireshark.

Chapter

7 Deployment Scenarios and Sizing

Call Recording is flexible in deployment. The following scenarios illustrate the various ways you can implement Call Recording in your IPT system.

The information in this chapter is divided into the following topics:

Deployment Models

The choice of the right IPT deployment model depends on many factors. The deployment of recording depends on the same factors.

With Call Recording you can record calls in any environment. For various network architecture models, there is a corresponding Call Recording deployment model. However, there are situations where it is better to optimize your existing model before adding Call Recording.

When choosing your deployment model for UC Manager as well as Call Recording, you must consider:

- Number of users, endpoint devices and types
- Geographical allocation of users and devices
- Required resistance (disaster recovery, redundancy etc.)
- Planned features and required type of administration
- Forecasting of expansion and required scalability

Based on best practices of implementation of Call Recording there are four sample scenarios with recommended hardware and software specifications. Scenarios are scaled by the maximum of simultaneous calls recorded by Call Recording. All scenarios are designed as single site installations. In the case of multisite environments, simply estimate the amount of simultaneous calls for every site and for each use values from the corresponding single site model.

Single Server

The simplest design is a single server solution. All the Call Recording modules are on the same server and most of the configuration files point to `localhost`. This is the basic design suitable for small sites with 50 – 100 simultaneous calls, and a maximum of 100 agents.

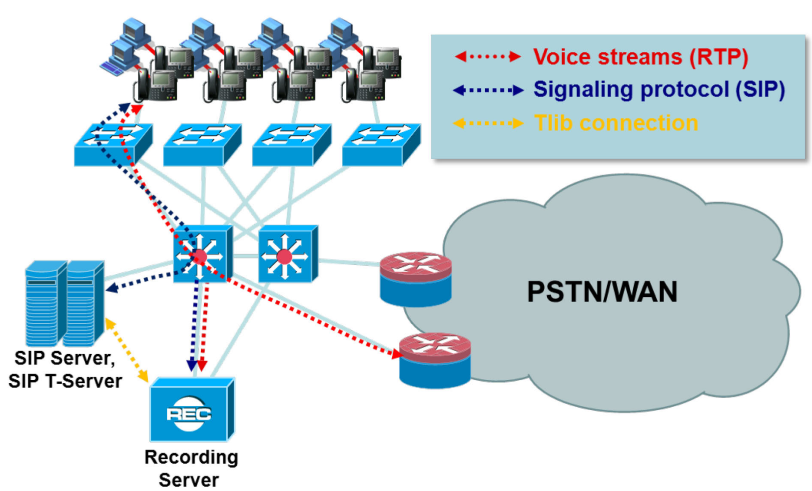


Figure 18: Single server

Redundant Installation

For critical environments, Call Recording can be deployed in a fully redundant configuration. Two servers are then configured identically; both are running and recording the same traffic. Automatic synchronization is automatically checking recorded interactions on one server against the other. In the event of server failure, any calls missing on the local machine get synchronized from the other server.

This active-active approach, compared to cold-standby or hot-standby, guarantees not losing any calls in the event of one server failure. After a malfunctioning server is repaired and placed back in operation, the redundancy synchronization automatically copies the missing call recordings and call metadata.

Redundant servers can also be configured to synchronize all the recorded calls to a dedicated Replay server, which is very popular in larger architectures with many branch offices.

For a fully redundant solution based on SPAN recording all server requirements are duplicated for the second server.

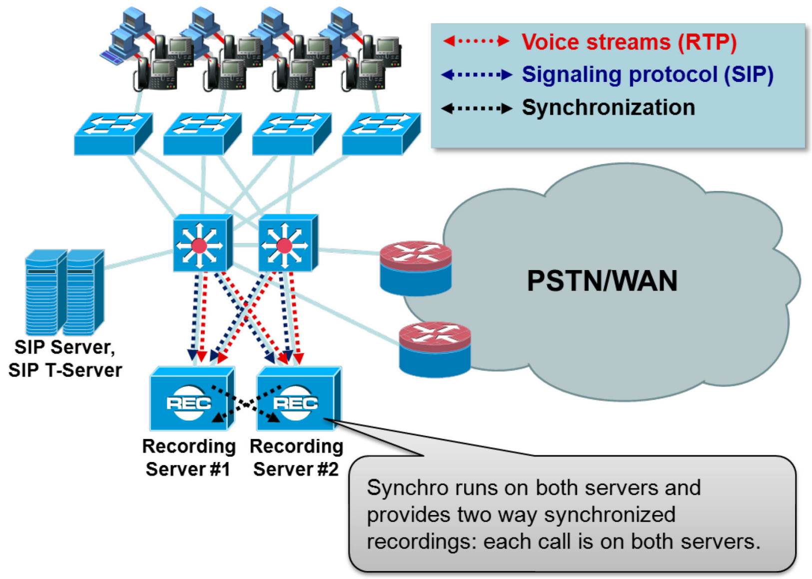


Figure 19: Redundant installation

The basic installation is quite similar to cluster installation. First, prepare all of the servers and install the basic system. Then configure the necessary connections and dependencies. Please refer to the Call Recording Administration Guide for installation and configuration details.

Replay Server

When Call Recording is deployed as a redundant solution or in larger architectures with many branches, using a Replay server is highly recommended.

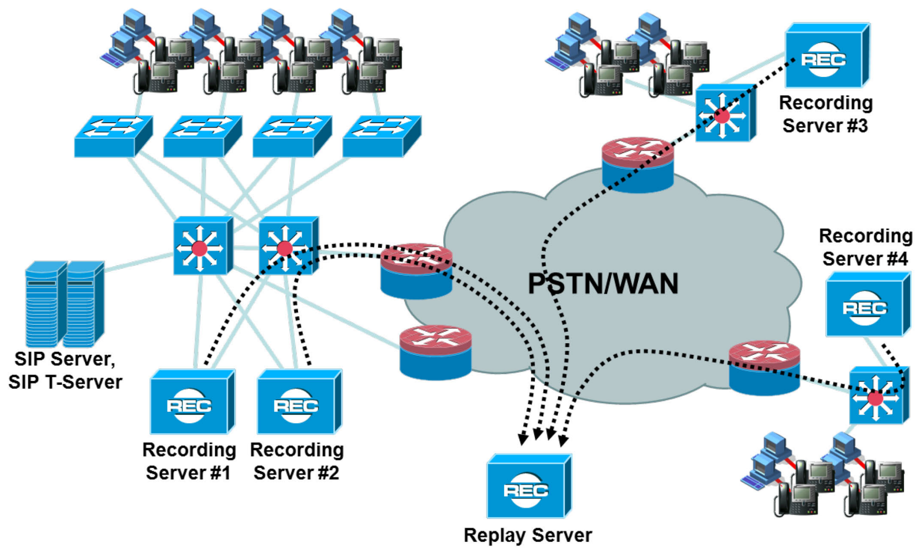


Figure 20: Replay server

The Replay server is used to access the calls, database, user management, archiving and backup, all in a central location. Replay server does not record any calls.

System users then work only with the Replay server. All the calls from deployed Call Recording servers are synchronized to the Replay server in real-time. In the event of failure of a single Call Recording server, users do not notice any interruption, since the calls are synchronized to the Replay server from the second Call Recording server that is currently recording.

Distributed Architecture

Call Recording can be deployed to record calls at multiple locations while providing the full benefits of a centrally managed system.

In a distributed architecture, the Call Recording core server takes care of handling call signalization (SIP or other protocol adapter), post-processing of captured data, the GUI, the database of calls, users, recording rules and central management of archiving and backup.

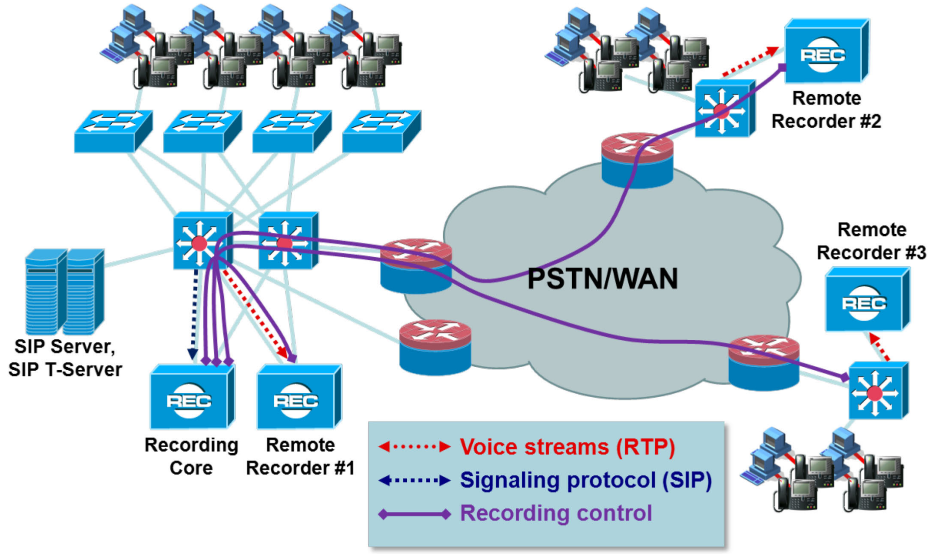


Figure 21: Distributed architecture

A Call Recording Remote Recorder is then deployed at the remote locations, and the Remote Recorders are centrally managed and controlled by the Core server. After a Remote Recorder finishes recording a particular call, it provides the Core server with data for further processing.

Cluster Installation

Cluster installation enables the recording of large telephony installations, load balancing, the possibility of recording in geographically distributed networks, and so on.

Call Recording uses clustering mainly for recording and decoding servers in environments with **more than 100 simultaneous calls**. In these cases there are two or more physical recording (decoding) servers configured as one homogenous server, and interconnected with the rest of the modules on one or more servers. This means that all Call Recording servers act like a single server installation.

Installation of a cluster solution requires:

- Solution design, description of the roles of particular servers
- Scaling the solution, scaling the individual server parameters
- Description of individual server properties; installed components, partitioning, network connections, file system sharing, etc.
- Installation of individual servers.
- Configuration of individual servers, network connection setup, file system sharing setup

Please refer to the *Call Recording Administration Guide* for the necessary configuration procedures and post-installation steps.

Hardware Sizing Models

The following models illustrate the components and hardware requirements for increasingly complex call recording scenarios.

For reference, Call Recording components are referred to by the following acronyms:

- CORE: central Call Recording control module
- RS: SPAN recorder
- DECODER: decoder server
- CC IM: Contact Center Integration Module (e.g. GIM for Genesys CIM)
- TOOLS: Media Lifecycle Management tools (e.g. Archive utility)
- WEB UI: Call Recording Web-based user/administration interface
- DATABASE: main call information database
- RTS: real-time protocol sniffer (e.g. JTAPI, SIP, Skinny)

Model 0 – VMWare – less than 10 simultaneous calls

The VMWare Virtual Appliance is a full product distribution, enabling customers to evaluate a particular product release in their network environment using an extendable 30 day evaluation license. The VA supports VMWare-ESX 3.5 and 4.0, and is configurable for Cisco JTAPI or generic SIP call signaling.

Model 1 – less than 100 simultaneous calls

This is the simplest standard scenario and does not need any special multi-server solution. In the case that the maximum number of simultaneous calls is below 100, a single server has enough resources to handle all demands. In this scenario, all Call Recording components can be installed on a single server.

The recommended server hardware configuration is as follows:

Call Recording Only

Server 1	
Components	**CORE, RS, DECODER, CC IM, TOOLS, WEB UI, DATABASE, RTS**
CPU	**Intel Quad Core 55XX 2.0Ghz+ or higher**
MEM	**4GB RAM (8GB RAM if both RS and SLR will be active at the same time)**
HDD	**2x72GB SCSI/SAS**

Table 1: Less than 100 calls, Call Recording only

Call Recording + Quality Manager

Server 1	
Components	CORE, RS, DECODER, CC IM, TOOLS, WEB UI, DATABASE, RTS
CPU	Intel Quad Core 55XX 2.0Ghz+ or higher
MEM	8GB RAM (12GB RAM if both RS and SLR will be active at the same time)
HDD	2x72GB SCSI/SAS

Table 2: Less than 100 calls, Call Recording + Quality Manager

Model 2 – less than 250 simultaneous calls

For recording more than 100 and fewer than 250 simultaneous calls, two independent servers should be deployed. In this case the installation of Call Recording is divided into two parts – the first server handles only the recording, while all other modules reside on server two. This mode requires small changes in the connectivity schema, but the functionality remains unchanged.

The recommended server hardware configuration is as follows:

Call Recording Only

Server 1	
Components	CORE, RS, RTS
CPU	Intel Quad Core 55XX 2.3Ghz+ or higher
MEM	8GB RAM
HDD	4x72GB SAS 10k rpm for RAID 10 partition (optimized write performance) Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance
Server 2	
Components	DECODER, CC IM, TOOLS, WEB UI, DATABASE
CPU	Intel Quad Core 55XX 2.3Ghz+ or higher
MEM	8GB RAM
HDD	4x72GB SAS 10k rpm RAID 10 (optimized read/write performance) Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance.

	The amount of memory and HDD capacity on the Replay server should reflect the amount of calls that will be stored in the database and the response time requirement. Use the database size calculator for a more precise recommendation!

Table 3: Less than 250 calls, Call Recording only

Call Recording + Quality Manager

Server 1	
Components	**CORE, RS, RTS**
CPU	**Intel Quad Core 55XX 2.3Ghz+ or higher**
MEM	**8GB RAM**
HDD	**4x72GB SAS 10k rpm for RAID 1+0 partition (optimized write performance)** Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance
Server 2	
Components	**DECODER, CC IM, TOOLS, WEB UI, DATABASE**
CPU	**Intel Quad Core 55XX 2.3Ghz+ or higher**
MEM	**16GB RAM**
HDD	**4x72GB SAS 10k rpm RAID 10 (optimized read/write performance)** Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance. The amount of memory and HDD capacity on the Replay server should reflect the amount of calls that will be stored in the database and the response time requirement. Use the database size calculator for a more precise recommendation!

Table 4: Less than 250 calls, Call Recording + Quality Manager

Model 3 – less than 500 simultaneous calls

To extend recording capacity up to 500 simultaneous calls requires three servers: one server for the Decoder, one server for the Database & Web Interface, and one server for the other modules.

The Decoder module needs the most computing power, so it requires a standalone server (Server 1).

On Server 2 it is recommended to install the Database module and the Call Recording Web Interface – this ensures enough resources for database functions and user access.

All remaining components are installed on Server 3. Optionally, a second instance of the Recorder can also be installed on this server, with RTP traffic distributed between the two recorders.

The recommended server hardware configuration is as follows:

Call Recording Only

Server 1

Components	**CORE, RS, RTS**
CPU	**Intel Quad Core 55XX 2.4Ghz+ or higher**
MEM	**8GB RAM**
HDD	**4x72GB SAS 10k rpm for RAID 10 partition (optimized read/write performance)** Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance

Server 2

Components	**DECODER, CC IM**
CPU	**2x Intel Quad Core 55XX 2.4Ghz+ or higher**
MEM	**8GB RAM**
HDD	**2x146GB SAS 15k rpm for RAID 1 partition for QM**

Server 3

Components	**TOOLS, WEB UI, DATABASE**
CPU	**2x Intel Quad Core 55XX 2.4Ghz+ or higher**

MEM	**16GB RAM**
HDD	**2x146GB SAS 15k rpm for RAID 1 partition for QM** Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance. The amount of memory and HDD capacity on the Replay server should reflect the amount of calls that will be stored in the database and the response time requirement. Use the database size calculator for a more precise recommendation!

Table 5: Less than 500 calls, Call Recording only

Call Recording + Quality Manager

Server 1	
Components	**CORE, RS, RTS**
CPU	**Intel Quad Core 55XX 2.4Ghz+ or higher**
MEM	**8GB RAM**
HDD	**4x72GB SAS 10k rpm for RAID 10 partition (optimized read/write performance)** Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance
Server 2	
Components	**DECODER, CC IM**
CPU	**2x Intel Quad Core 55XX 2.4Ghz+ or higher**
MEM	**8GB RAM**
HDD	**2x146GB SAS 15k rpm for RAID 1 partition for QM**
Server 3	
Components	**TOOLS, WEB UI, DATABASE**
CPU	**2x Intel Quad Core 55XX 2.4Ghz+ or higher**
MEM	**16GB RAM**
HDD	**2x146GB SAS 15k rpm for RAID 1 partition for QM** Note: 4 physical HDDs are recommended, because Call Recording is demanding on HDD performance. The amount of memory and HDD capacity on the Replay server should reflect the amount of calls that will be stored in the database and the response time requirement. Use the database size calculator for a more precise recommendation!

Table 6: Less than 500 calls, Call Recording + Quality Manager

Model 4 – more than 500 simultaneous calls

For a deployment requiring more than 500 simultaneous calls, it is strongly recommended to contact Genesys Support for consultation.

Please bear in mind that for geographically distributed architectures, every location should be sized as standalone and then all recordings should be consolidated to a central Replay Server.

Additional Requirements for Screen Capture

Screen Capture deployments will require the following hardware specifications in addition to the Call Recording or Call Recording + Screen Capture scenarios outlined earlier.

The following figures are calculated using the assumption that the estimated bandwidth required for one Screen Capture session is 400kbit/s.

Screen Capture Sessions	CPU	MEM	HDD
Up to 500	1x Quad (2.5GHz+)	8GB RAM	2xHDD (RAID 1)
Up to 1000	2x Quad (2.6GHz+)	8GB RAM	4xHDD (RAID 10)

Table 7: Additional requirements for Screen Capture

General Sizing Limits

The following sections indicate safe deployment limits for Call Recording-based installations.

Single Server Deployment

Architecture limitations:
- Max. 1M calls in database
- Max. 25 concurrent users
- High availability tools not installed (otherwise lower performance by 10%)

The limits above ensure that the deployed system will be suitable for both recording and user access.

- Overall system limit on a single server (all components running on the same HW) without Screen Capture: **100 concurrent calls**
- Overall system limit on a single server (all components running on the same HW) with Screen Capture: **75 concurrent calls/30 concurrent screens**

Multi-server Deployments

Absolute limits

- Please contact Genesys Support for more information.

Replay Server

Absolute limits

- Overall database limit for centralized recording storage: **100.000.000 recorded calls**

If more than 100M recorded calls need to be stored on the Replay Server, the database must be split into two physical instances and Synchro Tool will be configured to move oldest recordings from first database (primary) into the second (historical).

Calculating Disk Space Usage

Note: A Media Lifecycle Management (MLM) storage planning calculator (in the form of an Excel workbook) is available as a free download from Genesys Support. The following information is an overview of the calculations involved in estimating storage facilities required.

The amount of disk space needed on your storage server depends on three parameters:

- The codec and the bitrate used for storing calls
- The lifetime of stored calls
- The count of calls per day and their average length

Using default settings (24 kbps), one minute of MP3 recording consumes 180 kB of disk space.

The formula for this calculation is: chosen bitrate transformed from bits to bytes gives you required disk space for one second of recording.

$$(\textbf{bitrate}\ (\textbf{kbps})/\textbf{8}) * \textbf{average length of call} * \textbf{count of calls} = \textbf{space required}\ (\textbf{kB})$$

So, for example 500 daily calls lasting 3 minutes (180 seconds) transcoded into MP3 with a bitrate of 96 kbps will require:

$$(\textbf{96kbps}/\textbf{8}) * \textbf{180s} * \textbf{500} = \textbf{1 080 000 kB}$$

You can now multiply this value by the assumed lifetime of stored calls to get the required storage server capacity.

Finally, it is recommended to add at least **10% reserve**.

Note: When running Call Recording Tools as a daemon, additional memory and fast storage are utilized. If you are planning to run all tools as daemons simultaneously, add at least 2 GB of memory to the server where tools are running.

Chapter

8 Appendix A: Introduction to IP Telephony (IPT)

The information in this chapter is divided into the following topics:

How IPT works

Voice over Internet Protocol (VoIP) is a general term for a family of transmission technologies for delivery of voice communications over IP networks such as the Internet or other packet-switched networks. Other terms frequently encountered and synonymous with VoIP are *IP telephony*, *Internet telephony*, *voice over broadband* (VoBB), *broadband telephony*, and *broadband phone*.

Internet telephony consists of all communications services—voice, facsimile, and/or voice-messaging applications—that are transported via the Internet, rather than the public switched telephone network (PSTN). The basic steps involved in originating an Internet telephone call are the conversion of the analog voice signal to digital format and compression/translation of the signal into Internet protocol (IP) packets for transmission over the Internet; the process is reversed at the receiving end.

VoIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. Codec use is varied between different implementations of VoIP (and often a range of codecs are used); some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs.

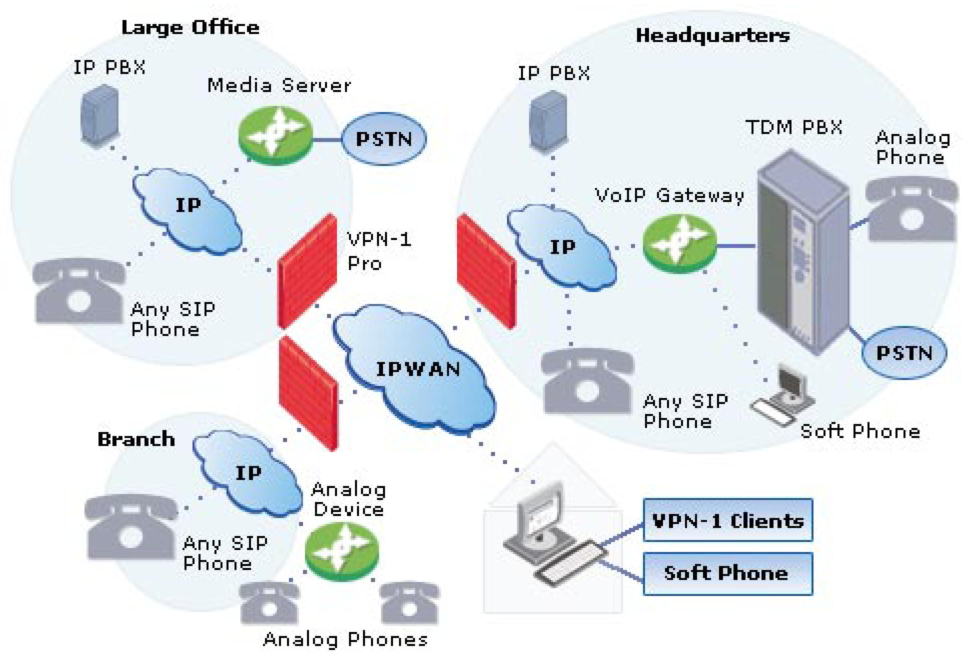


Figure 22: How IPT works
(source: http://en.wikipedia.org/wiki/IP_Telephony)

Shortly: IP telephony is voice communication based on the IP network. Voice is recoded and transcoded into digital data and transmitted as packets via the Internet Protocol (IP) and after transmission again transformed into voice.

There are three basic models of IPT:

- **IP-to-IP** communication which is using as transfer media exclusively IP network.
- **IP-to-PSTN** (or ISDN) and vice versa. For the first part of voice transfer, the same communication model is used as above, but a voice gateway (VG) is also used for interconnection between the IP network and “classic” telephone network.
- **PSTN-to-PSTN via IP** network is used for toll by-pass (dramatically reducing price of long-distance calls).

IPT Benefits

IPT is more than only voice communication. IPT technology offers many additional services, which are not possible (or very limited) in classic telephony. For example these features of Cisco UC Manager are all supported by Call Recording:

Conference calls – Easy to setup and add new callers.

Call waiting - A second (and third etc.) call can be received while on a current call.

Call forwarding - Calls can be forwarded to a voice mail or another phone across the IP network as well as PSTN/ISDN.

Call transfers – Connected calls can be easily transferred to another phone in the network (not only in the LAN)

Parked and picked up calls – Callers can “hold” already connected calls and assign an identifier, then the call can be picked again up on any endpoint after entering the identifier.

Barged calls - Barging allows a caller to get added to another active call that is on a shared line.

Caller ID – IP phones display a variety of associated information to indentify callers, such as information from a CRM or other databases.

Portability - Fully portable functionality wherever your IPT solution has coverage.

Shared lines – One number shared by more endpoints (IP phones) with on-the-fly changes of configuration.

Customer service – IPTs enable help desk and other customer related services. Calls can be directed to CTI ports and applications according to applications like IVR.

Cisco UC Manager

Cisco Unified Communications Manager (formerly Cisco Unified CallManager) provides the call-processing component of the Cisco Unified Communications system, in addition to being the foundation for advanced capabilities including video, mobility, and presence-based services. It is a scalable, distributable, and highly available enterprise IP telephony call-processing solution.

Cisco Unified Communications Manager extends enterprise telephony features and capabilities to packet telephony network devices such as IP phones, media processing devices, voice over IP (VoIP) gateways, and multimedia applications. Additional services, such as unified messaging, multimedia conferencing, collaborative contact centers, and interactive multimedia response systems, are made possible through open telephony APIs.

Cisco Unified Communications Manager is installed on the Cisco 7800 Series Media Convergence Servers and includes the following features:

- Highly scalable, supporting up to 30,000 lines per server cluster
- Able to support a full range of communications features and applications, including SIP-based applications
- Highly available for business continuity, supporting multiple levels of server redundancy and survivability
- Support for a broad range of phones to suit varying user requirements
- Choice of operating system environments: Windows server-based implementation or Linux-based appliance model implementation
- Available in an easy-to-manage single-server solution, Cisco Unified Communications Manager Business Edition, that combines call processing and messaging

Voice Codecs

Codecs control how audio data is transformed and compressed before its transmission over an IP network.

The required bandwidth and quality of voice is based on the codec used. The most frequently implemented codecs are G.723, G.729 and uncompressed G.711, but you can use other codecs supported by Cisco UC Manager. The main differences between them are packet size, voice quality, and complexity. Both sides of an IPT call must use the same codec with identical parameters. The signaling protocol handles this when negotiating call initiation.

The following table compares codes by bit rate and quality for various codecs supported by Call Recording.

Codec	Compression	Bit rate	Quality
G.711	*PCM – Pulse Code Modulation*	64 kbps	4,1
G.723.1	*MP-MLQ – Multipulse Multilevel Quantization*	5,3 kbps	3,9
G.723.1	*ACELP – Algebraic Code Excited Linear Prediction*	6,4 kbps	3,65
G.726	*ADPCM – Adaptive Differential PCM*	16, 24, 32, 40 kbps	3,85
G.728	*LD-CELP – Low Delay - Code Excited Linear Prediction*	16 kbps	3,61

G.729	*CS-CELP - Conjugate Structure - CELP*	8 kbps	3,81
G.722	*ADPCM – Adaptive Differential PCM*	48, 56, 64 kbps	4,1
G.722.1	*ADPCM – Adaptive Differential PCM*	24, 32 kbps	4,2

Signaling and Control Protocols

Signaling protocols set standards for IPT communications. The protocols cover the process of setting up and tearing down a call, as well as the resources and coding negotiations between the participants in IPT calls.

Cisco UC Manager uses three signaling and control protocols Skinny (or SCCP), SIP and MGCP. There are also other protocols for IPT, but they have no support in Cisco UC Manager.

SCCP (Skinny)

The “Skinny” client control protocol is the Cisco standard for real-time calls and conferencing over IP. UC Manager utilizes the Skinny Client Control Protocol (SCCP) as a communications protocol for signaling the hardware endpoints of the system - as a messaging set between a skinny client and the Cisco UC Manager. Other hardware vendors implement SCCP into their products – such as IP phones or soft-switches (which are important parts of IPT architecture and useful for IPT recording).

Skinny’s main benefits come from its lightweight design using a code based set of defined commands. The disadvantage is its relatively bigger demand of computing power on the client side compared with other protocols, but this issue isn’t so significant with today’s powerful processors.

SIP

SIP stands for “Session Initiation Protocol” and one of its biggest advantages is open specifications and simple implementation. SIP is supported by Cisco UC Manager from version 4.0 and by Call Recording from version 3.2.5. In UC Manager SIP is used to pass call signaling to gateways.

In RCF 3261 SIP is defined as a signaling protocol for IPT calls, multimedia distribution and multimedia conferences. From the beginning SIP was designed as a proxy-based protocol, using proxies to help route requests to the user's current location. SIP also provides a registration function that allows users to indicate their current location. Based on this information a SIP network can select the best available proxy.

In comparison with SCCP, SIP is extremely lightweight – it consists of only six standard methods and is completely text-based. SIP is transport independent because it can use not only UDP, but also TCP and so on. SIP is easy to implement and is able to pass through firewalls without additional configuration.

MGCP

The Media Gateway Control Protocol is based on distributed architecture which consists of Call Agents (or Media Gateway Controllers), Media Gateways, and Signaling Gateways. Media Gateways convert media signals between circuits and packets. Signaling Gateways control interconnection with PSTN circuits. MGCP is used by UC Manager to pass call signaling to gateways.

Chapter

9 Appendix B: Call Couple Examples

The number of recorded call files created by Call Recording depends on the type of call. Conference Calls, Transferred Calls or Barged Calls are all handled differently, resulting in different combinations and lengths of recorded call files.

Call Recording allows you to play any individual call by simply clicking on it. You can also select groups of related calls to be played back together. When you play back a group of related call recordings in the Advanced Player, you see and hear all call streams in their proper sequence.

The information in this chapter is divided into the following topics:

- Conference Calls
- Transferred Calls
- Barged Calls

Conference Calls

- Caller A connects with Caller B.
- Caller C joins the conversation in the middle, and leaves it before Caller A and Caller B finish their conversation.
- The result is six different files.

Caller A calls Caller B. File 1 is created.
Caller A calls Caller C, and invites them into a conference call. This is stored as File 2.
When Caller C joins the conference, and Files 3, 4, and 5 are created.
When Caller C leaves the conference before its end, the call reverts to a classic two-sided call, and the remainder of the conversation is stored as File 6.

	A -> B	A -> C	A+B+C	A+B
A	File 1	File 2	File 3	File 6
B			File 4	
C		File 2	File 5	

When you listen to a Conference call in Advanced player, you see and hear all of the call recording files.

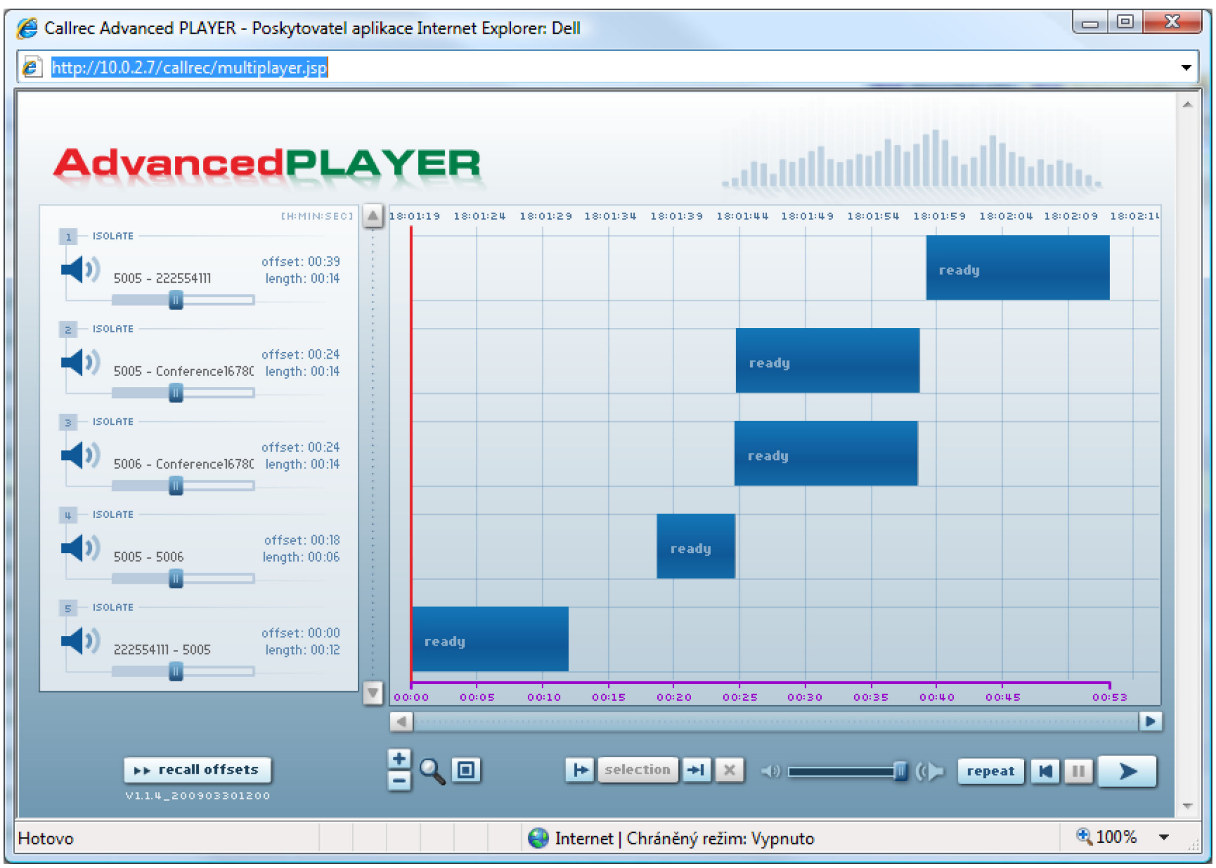


Figure 23: Conference Calls

Transferred Calls

- Caller A connects with Caller B.
- Caller B connects with Caller C, requesting a transfer.
- Call is accepted by Caller C, and Caller B hangs up.
- Caller A connects with Caller C.
- The result is three different files.

Caller A connects with Caller B. File 1 is created.
Caller B connects with Caller C. File 2 is created.
Caller A connects with Caller C. File 3 is created.

<table>
<tr><th></th><th>A -> B</th><th>B transfers to C</th><th>A+B</th></tr>
<tr><td>A</td><td rowspan="2">File 1</td><td>File 2</td><td>File 3</td></tr>
<tr><td>B</td><td>hanging-up</td><td></td></tr>
<tr><td>C</td><td></td><td>File 2</td><td>File 3</td></tr>
</table>

When you listen to a Transfer call in Advanced player, you see and hear all three call recording files.

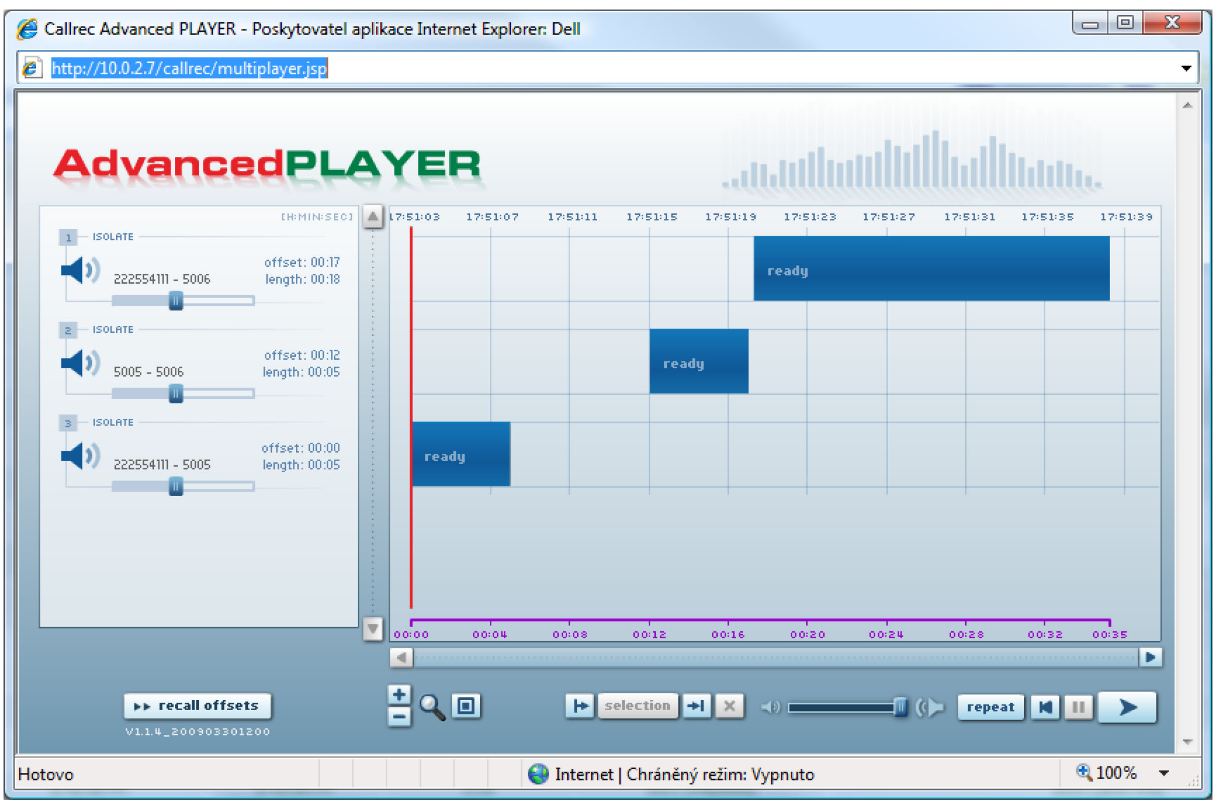


Figure 24: Transferred Calls

Barged Calls

- Caller A connects with Caller B.
- Caller C listens to the conversation between Caller A and Caller B.
- The result is two different files.

Caller A connects with Caller B. File 1 is created.
Caller C listens to conversation. File 2 is created, containing only the portion of the call that Caller C hears.

	A -> B	C listening
A	File 1	
B		
C		File 2

When you listen to a barge call in the advanced player, you see and hear both calls.

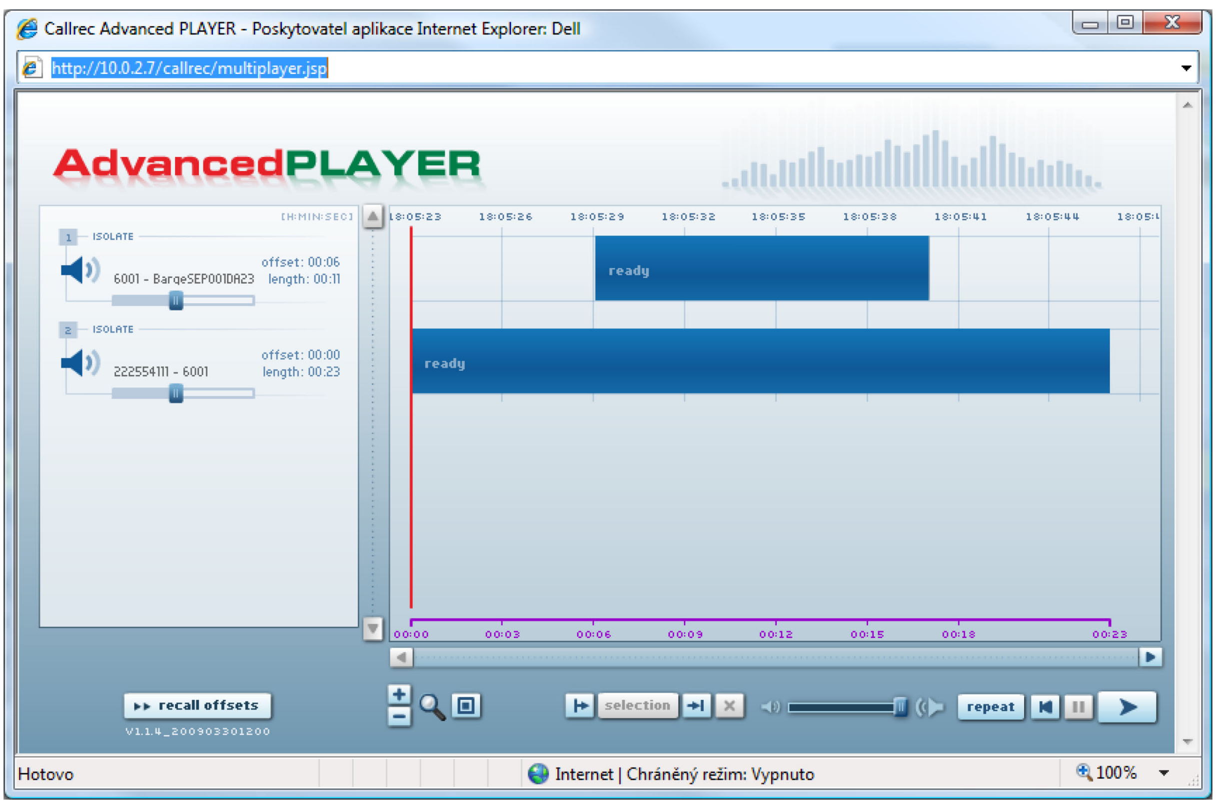


Figure 25: Barged Calls

Chapter

10 Glossary

Call Recording – Genesys Call Recording IP telephony call recording solution

CUCM – Cisco Unified Communications Manager

IPT – IP telephony, IP telephony system

NAS - Network Attached Storage – a storage device directly attached into network (mostly LAN) used only for data storing and related operations

RSPAN - Ability to remotely SPAN traffic across a cascade of switches

SAN - Storage Area Network – independent data network mainly based on cable infrastructure or fiber optic consisting of data storage devices and active network components. It is used only for data storage traffic.

Softswitch - A softswitch is a central device in a telephone network which connects calls from one phone line to another, entirely by means of software running on a computer system. This work was formerly carried out by hardware, with physical switchboards to route the calls.

SPAN - Ability of switches to copy specific traffic to a defined port

VLAN - Virtual Local Area Network – secured independent network inside a classic LAN network

Chapter

11 Requesting Technical Support

Technical Support from VARs

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

Technical Support from Genesys

If you have purchased support directly from Genesys, contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North America and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868	support@genesyslab.com.au
India	000-800-100-7136 (toll-free) +91-(022)-3918-0537	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp

Before contacting Genesys technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.