



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Interaction Recording Solution Guide

Genesys Interaction Recording 8.5.2

Table of Contents

Genesys Interaction Recording Solution Guide	6
About Genesys Interaction Recording	8
Genesys Interaction Recording 8.5.2	10
Recording Methods	18
How Recording Works	21
Getting Started with Genesys Interaction Recording	30
Deploying Genesys Interaction Recording in a Single-Tenant Deployment	32
Deploying Interaction Recording Web Services (RWS)	35
Prerequisites	36
Deploying Cassandra	38
Installing and Configuring Cassandra	39
Upgrading Cassandra to 2.2	47
Upgrading Cassandra to 4.1	48
Configuring WebDAV	51
Initializing Cassandra	55
Elasticsearch	59
Installing	81
Deploying the Web Application	86
Configuring Interaction Recording Web Services	89
Configuring Security	105
Starting and Testing	124
Configuring Features	126
Configuration Options	139
Deploying Web Services and Applications for GIR	174
Deploying SIP Server for GIR	192
Deploying Interaction Concentrator for GIR	195
Deploying Recording Crypto Server	197
Deploying the Recording Plug-in for GAX	213
Deploying Recording Processor Script	220
Deploying Voice Processor	274
GIR Voice Processor deployment using Podman	288
Deploying Genesys Voice Platform for GIR	299
Encrypting and Provisioning Certificates	306
Deploying the Screen Recording Service	317
Deploying the Screen Recording Service - Advanced Configuration	332

Deploying Recording Muxer Script	359
Deploying SpeechMiner for GIR	417
Deploying Workspace Desktop Edition for GIR	424
Configuring permissions, access control, and privacy	425
Secure Transport Configuration	433
Configuring Media Lifecycle Management	437
Creating Folder Hierarchy for Recording Storage	444
Setting up the Load Balancer in a Single-Tenant Environment	447
Additional Feature Configuration	464
Genesys Interaction Recording Options Reference	466
Deploying Genesys Interaction Recording in a Multi-Tenant Deployment	491
Deploying Interaction Recording Web Services (RWS)	35
Prerequisites	36
Deploying Cassandra	38
Installing and Configuring Cassandra	39
Upgrading Cassandra to 2.2	47
Upgrading Cassandra to 4.1	48
Configuring WebDAV	51
Initializing Cassandra	55
Elasticsearch	59
Installing	81
Deploying the Web Application	86
Configuring Interaction Recording Web Services	89
Configuring Security	105
Starting and Testing	124
Configuring Features	126
Configuration Options	139
Deploying Web Services and Applications for GIR	174
Deploying SIP Server for GIR	192
Deploying Interaction Concentrator for GIR	195
Deploying Recording Crypto Server	197
Deploying the Recording Plug-in for GAX	213
Deploying Recording Processor Script	220
Deploying Voice Processor	274
GIR Voice Processor deployment using Podman	288
Deploying Genesys Voice Platform for GIR	299
Encrypting and Provisioning Certificates	306

Deploying the Screen Recording Service	317
Deploying the Screen Recording Service - Advanced Configuration	332
Deploying Recording Muxer Script	359
Deploying SpeechMiner for GIR	417
Deploying Workspace Desktop Edition for GIR	424
Configuring permissions, access control, and privacy	425
Secure Transport Configuration	433
Configuring Media Lifecycle Management	437
Creating Folder Hierarchy for Recording Storage	444
Setting up the Load Balancer in a Multi-Tenant Environment	914
Additional Feature Configuration	464
Genesys Interaction Recording Options Reference	466
Migrate Genesys Interaction Recording from a Single Tenant to a Multi-Tenant Deployment	952
Architecture and Features	964
Genesys Interaction Recording Components	965
IVR Recording	967
Multi-Site Call Transfers	972
Call Flows	977
The Call Recording Model	981
Screen Recording Architecture	985
Multiple data center locations	990
Media Lifecycle Management	997
Scalability and High Availability	1001
Security and Encryption	1004
Archiving and Metadata	1008
Access Control for Recording Users	1023
How the T-Library Works for GIR	1027
User Interfaces	1030
Geo-Location	1032
Audio Tones	1035
Reporting	1037
Appendixes	1038
Example Solution Definition SPD File	1039
Disk Storage Recommendations	1041
Sample Certificate and Key File Generation	1042
Interaction Recording Web Services (Web Services) Group Settings	1043
Agent Hierarchy and Partitioning Examples	1057

Secure Transport Configuration	433
Automated Recovery of Recordings	1069
Recovering Metadata for SpeechMiner	1085
Troubleshooting	1090
Understanding Genesys Interaction Recording	1095
Minimum Recommended Versions	1098
GIR Alarms	1102

Genesys Interaction Recording Solution Guide

The Genesys Interaction Recording (GIR) 8.5 Solution Guide provides an overview of Genesys Interactive Recording. These pages are valid for all 8.x releases of Genesys Interaction Recording.

Genesys Interaction Recording (GIR) can be deployed in a single-tenant environment or in a multi-tenant environment. To successfully deploy GIR you must follow the instructions provided, in the order that they appear:

- [Deploy GIR in a Single-Tenant Environment](#)
- [Deploy GIR in a Multi-Tenant Environment](#)

About Genesys Interaction Recording

Find out about Genesys Interaction Recording:

[Overview](#)
[How It Works](#)
[New in this Release](#)
[Getting Started](#)

*all
topics>*

Features and Architecture

Find out about the supported features:

[Core Components](#)
[Encryption](#)
[Archiving and Metadata](#)
[Media Lifecycle Management](#)

*all
topics>>*

Installation and Configuration

Find out about how to install and configure the solution:

[Web Services and Applications](#)
[SIP Server](#)
[GVP](#)
[SpeechMiner](#)

*all
topics>*

Other Information

[Disk Storage Recommendations](#)
[Troubleshooting](#)
[Transport Layer Security](#)

*all
topics>>*

About Genesys Interaction Recording

Genesys Interaction Recording (GIR) is a compliance and control platform based on Genesys SIP, the T-Lib protocol, and the Genesys proprietary event model. Fully integrated to the CIM platform, Genesys Interaction Recording provides economies and powerful recording control via a host of integrations across the suite.

The voice portion of interaction recording relies on Media Server to perform Dual Channel Recording, where Media Server captures the audio from the RTP streams and makes them available to the recording storage.

The screen portion of interaction recording relies on a Screen Recording Client running on the Agent Desktop to perform screen captures and makes them available to the recording storage via Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier).

Additional events and information are provided by the SIP Server (via ICON) for voice interactions and by the Interaction Server for non-voice interactions (like emails, chats, etc.).

This powerful solution will enable the modern contact center to record the entire customer interaction, using the Genesys Speech and Text Analytics platform, across multiple sites and interactions, allowing the contact center to meet quality or regulatory compliance requirements.

Genesys provides your organization with reliable, high-quality recordings of both audio communications and desktop screen activity. Now you can capture 100% of all interactions, even if customers are transferred multiple times to agents in geographically dispersed locations. Interaction recordings can even be shared and sent through email to your agents, managers or customers, as required. And, if there is a customer complaint or dispute, interaction metadata saves you valuable time in locating the right interaction from thousands of hours of recordings.

Integrated into the Genesys Customer Experience Platform, through native integration with the Genesys SIP communications infrastructure, you can record 100% of all interactions. Genesys Interaction Recording can analyze metadata from each interaction and evaluate which interactions must be recorded using user-defined recording rules. Employing this same metadata, you can quickly search and retrieve stored recordings, helping to resolve any customer complaints more efficiently.

Your customer service organization can benefit immediately from recording integration, as configuration and maintenance are performed within one platform. You can define recording profiles to meet internal and external policy requirements, including the ability to archive recorded sessions to separate storage locations. Single sign-on and role-based access ensures recorded interactions are viewed by authorized employees only, and sensitive data is hidden to prevent unauthorized data loss.

Core Capabilities

An example of some of the available core capabilities of the Genesys Interaction Recording solution are as follows:

- **Full-time recording**—Records every call for a specific DN through configuration.
- **Selective recording**—A decision to record a party in the call is made at a Routing Point and the recording starts as soon as the call is established.

- **Dynamic recording**—Recording Sessions are established on an as-needed basis after the Communication Session is established. T-Library recording functions are provided to allow third parties, such as Agent Desktop, to record on demand.
- A call recording can be started while supervisor monitoring is enabled.
- Real-time control of the call recording—The recording can be paused and resumed on demand by the agent, or by the workflow when the customer provides sensitive data such as a PIN. This is the same functionality as **Dynamic recording**.
- Support of secured communications such as SIPS and SRTP.
- **IVR recording**.
- **Screen recording**.
- Genesys Administrator Extension plug-in for the administration of Call Recording.
- Speech and Text Analytics direct audio support.
- **Geo Location** for WAN/Latency cost control.
- **Audio Tones** for compliance.
- **Multi-site** call recording and retrieval.
- **Multiple storage** locations at a single site.
- Full PKCS #7 **encryption**.
- Storage, Retrieval, and **Archive** support.
- Systemic monitoring and alarming.
- Unified UI across Call Recording, QM, and Speech Analytics.
- Record via attached data - "Call Type=Gold Credit Card".
- Architectures: Cloud / Premise / Hybrid.

Genesys Interaction Recording 8.5.2

New in this Release

Release **8.5.225.01** — June 09, 2022, support for Windows Server 2019 and Red Hat Enterprise Linux 8.

Release **8.5.224.00** — June 10, 2021, contains resolved issues only.

Release **8.5.223.00** — October 19, 2020, contains resolved issues only.

Some of the primary new features added in release **8.5.222.00**—April 24, 2020:

- Genesys Interaction Recording now supports SIP Cluster in deployments that are using Voice Processor.

Important

Recording Processor Script (RPS) cannot be used with SIP Cluster.

- Interaction Recording Web Services now supports authentication with Cassandra.
- The Screen Recording Service (SRS) now supports the use of a web proxy for outbound connections.
- Elasticsearch v2 schema performance is significantly improved. For existing deployments, these improvements will become increasingly evident as older interactions are purged from the system.
- With this release, Voice Processor is now generally available and no longer under shipping control by the Genesys Product Management team. For new deployments, Genesys recommends using Voice Processor instead of Recording Processor Script (RPS). For information on how to deploy Voice Processor, see [Deploying Voice Processor](#).

Some of the primary new features added in release **8.5.221.01**—April 01, 2019:

- The scan and scroll method of Elasticsearch is now supported for MLM with Elasticsearch v2 schema. For large scale deployments, enable the scan and scroll option to improve performance of MLM.
- The default number of shards for new deployments with Elasticsearch v2 schema is 12. The existing deployments with the current number of shards are also supported.

Some of the primary new features added in release **8.5.221.00**:

- January 11, 2019, contains resolved issues only.
- December 20, 2018:
A new multi-threaded microservice, Voice Processor, is introduced. Voice Processor can be used instead of the Recording Processor Script (RPS) which is currently in use. GIR needs Voice Processor to process recording metadata from Media Control Platform (MCP), combine this metadata with data collected from Genesys Info Mart (GIM), and forward the result to Interaction Recording Web Services (RWS) and SpeechMiner Interaction Receiver (SM IR). For information on how to deploy Voice Processor, see [Deploying Voice Processor](#).

Important

Customers with an existing RPS deployment can migrate to Voice Processor using the instructions provided in the [Migrating from RPS to Voice Processor](#) section.

- November 09, 2018:
Interaction Recording Web Services (RWS) now includes new parameters for a search API used by Recording Muxer Script. Recording Muxer Script uses these parameters when receiving call recordings from RWS.

Release **8.5.220.00** —August 28, 2018, contains resolved issues only.

Release **8.5.219.02**—June 04, 2018 (initially released as 8.5.219.01 on May 17, 2018), contains resolved issues only.

Important

As part of upgrading to this release, the Cassandra schema used by Interaction Recording Web Services must be upgraded. Refer to the migration steps described in [Upgrading Interaction Recording Web Services](#) for details.

Some of the primary new features added in release **8.5.218.00**—December 15, 2017:

- Interaction Recording now has a new call recording metadata attribute to indicate if a call recording has a related screen recording available ([screenRecording](#)). In addition, if a screen recording is associated with a call recording, a new screen recording metadata attribute ([callRecordingId](#)) indicates the associated call recording. These metadata values are only populated for new recordings moving forward. All older interactions will indicate that there is no associated screen recording.
- Screen Recording Service (SRS) now uses updated versions of Python and OpenSSL.
- SRS now supports a [configurable parameter](#) to ignore errors that occur during certificate verification for

screen recording encryption.

Some of the primary new features added in release **8.5.217.00**—September 27, 2017:

- A guide to using the [Screen Recording API](#) is available in the [Genesys Interaction Recording API Reference](#). This information can be used to integrate a third-party desktop with screen recording functionality.
- The following GIR components now support [configurable certificate validation](#) when making outbound TLS connections: Interaction Recording Web Services, Recording Muxer Script, Recording Plug-in for GAX, Recording Crypto Server, and Interaction Recording LVR Recovery Script.

Some of the primary new features added in release **8.5.216.01**—July 14, 2017:

- Support for automated recovery of recordings with a new Lost Voice Recording (LVR) Recovery Script component. See [Automated Recovery of Recordings](#) for details.
- Multi-site disaster recovery support for screen recording with the Screen Recording Service (SRS) and Interaction Recording Web Services (RWS) when Workspace Desktop Edition (WDE) version 8.5.118.10 or later is used.
- Support for the new **Delete Recording by ID** API in Interaction Recording Web Services. See [Genesys Interaction Recording API](#) for details.

Some of the primary new features added in release **8.5.215.00**—June 28, 2017:

- The following GIR components now support RHEL 7: Interaction Recording Web Services, Recording Muxer Script, Recording Plug-in for GAX, and Recording Crypto Server.
- You can now manage the life cycle of voice recordings by using interaction labels. For more information, refer to [Creating a Rule](#) and [Recording Lifecycle Scheduler Parameters](#).
- The Elasticsearch templates (**call_recordingv2_template.json** and **screen_recordingv2_template.json**) have been updated. Deploy the new versions of these templates to the Elasticsearch cluster so that when an index is created, the new index template is used. Re-indexing is not required after this step.
- Interaction Recording Web Services now supports two new APIs: **Get Recording by ID** and **Get Recording Media by ID**. For more information, refer to [Genesys Interaction Recording API](#).
- Interaction Recording Web Services now supports using Java Runtime Environment (JRE) 8 as an alternative to the Java Development Kit (JDK) 1.8.
- The SFDC Gplus Adapter now supports screen recording with the SR Service.
- When you delete an interaction in SpeechMiner, its associated voice recording and screen recording files are now also deleted.

Important

As part of upgrading to this release, the Cassandra schema used by Interaction Recording Web Services must be upgraded. Refer to the migration steps described in [Upgrading Interaction Recording Web Services](#) for details.

Some of the primary new features added in release **8.5.214.03**—April 28, 2017:

- Interaction Recording Web Services (RWS) now supports MLM backup in unzipped format to both the Windows and Linux file systems. To perform unzipped backup to a Windows file system, make sure the **useFullPathInMediaFileBackup** option is set to false. For more information on this option, refer to the following sections: [Media Lifecycle Management Archive Structure](#), [backgroundScheduledMediaOperationsSettings](#), and [Recording Lifecycle Scheduler Parameters](#).

Some of the primary new features added in release **8.5.214.02**—March 29, 2017:

- For new installations or new tenants in existing installations the Elasticsearch schema version 2 will be used by default. For an existing installation, this support for this version requires that you perform the migration steps described in the [Migrating an Existing Elasticsearch Deployment to Schema V2](#) section as part of upgrading to this release.
- GIR now supports the ability to tag interactions, and to protect them from being deleted, through SpeechMiner. You can also create and apply tags using the labels API, and protect recordings from deletion using the non-delete API. See [Recording Label API](#) and [Recording Non-Deletion API](#).

Important

To use the new SpeechMiner functionality the following configuration is required:

- If you are using tagging or deletion protection, Cross Site Request Forgery (CSRF) protection must be disabled in RWS. For details, refer to [CSRF Protection](#).
- The **RWS URI** field in the **SMConfig > Recording** tab must be configured. For details, refer to [Deploying SpeechMiner for GIR](#).
- The **Interaction Receiver** settings must be configured in RWS. For details, refer to [Create SpeechMiner Settings](#).
- The SpeechMiner username and password must be configured in **[recording.archive]** for each tenant. For details, refer to [Step 5 of Configuring SpeechMiner users](#).

- The Recording Lifecycle Scheduler (in Media Lifecycle Management) now supports storing backup (archive) files in unzipped format and zipped format. See the setting [in the RLS dialog box on this page](#) and a description [on this page](#).

- Genesys Interaction Recording now supports remote recording of established calls, using the recording capabilities of SIP Server version 8.1.102.55 or later with T-Server for Skype for Business version 8.5.001.17 or later.

Some of the primary new features added in release **8.5.213.04**—December 20, 2016:

- The Recording Muxer Script and Recording Processor Script now support passing password-related configuration values using environment variables, on both Windows and Linux.
- Support for registering multiple DNs with the Screen Recording Service, with desktops, via the [SRS login API](#).
Note: Required: the agent desktop must support using multiple DNs and Hot Seating with the Screen Recording Service. For additional information, refer to your Agent Desktop documentation.
- Support for Cassandra 2.2. Support continues for Cassandra 1.2, but Genesys recommends version 2.2. See [Deploying Cassandra 2.x](#).
- Support for [Setting up the Load Balancer in a Premise Multi-tenant Environment](#).

Some of the primary new features added in release **8.5.212.03**—September 27, 2016:

- Configuration support for Disposition Codes in GIR metadata filters.
- Support for a Premise Load Balancing mechanism for dedicated GIR nodes.
- Support for the following components running in the same environment:
 - Web Services and Applications
 - Interaction Recording Web Services (RWS)
- Java 8 support for Recording Plug-in for GAX and Recording Crypto Server (RCS).
- Support for Hot Seating for the SR Service with desktops via the SRS login API. **Note:** This support requires that the agent desktop supports Hot Seating with the Screen Recording Service. For additional information, refer to your Agent Desktop documentation.
- Support for Play application level Interactive Voice Response (IVR) recording.
- Support for Horizontal Scaling of the Recording Muxer Script.
- The Recording Muxer Script now supports the SR Service Nightly Upload.
- Ad hoc download of encrypted media assets.

Some of the primary new features added in release **8.5.212.02**—August 15, 2016:

- The **Encrypt Exported Interactions** feature enables you to encrypt exported interactions, so that a password is required to access the interactions. By default, exported interactions are now encrypted after upgrading SpeechMiner to 8.5.504.02.

Some of the primary new features added in release **8.5.212.01**—July 11, 2016:

- The `slowMachine` parameter is now deprecated, and is replaced by the new `vlcCloseTimeout` parameter.
- The SR Service will now retry to upload all failed recordings after the next restart.
- Support for SRS on VMware Horizon 7 using the RDP protocol.

Some of the primary new features added in release **8.5.211.01**—June 29, 2016:

- Support for a multi-tenant configuration server for GIR Screen Recording.
- Support for a multi-tenant configuration server for multi-tenant objects.
- Support for partitions per interaction segment.
- Interaction Recording Web Services replaces the Web Services and Applications prerequisites for the Genesys Interaction Recording solution. It includes all Interaction Recording related Web Services features that are available in Web Services and Applications releases up to 8.5.201.29—for example, storing and managing recording files. It does not provide API support for non-GIR related Web Services, such as Workspace Web Edition.
- Interaction Recording Web Services includes a new option (**`sessionCookieName`**). This new option can be used to define the name of the session cookie used by Interaction Recording Web Services.

Some of the primary new features added in release **8.5.210.02**—April 19, 2016:

- Support for metadata suppression for privacy and compliance.

Some of the primary new features added in release **8.5.210.01**—April 1, 2016:

- Windows Server 2012 support for Screen Recording (SR) Service Citrix.
 - Muxer can now rely on the Query Call Recording API to only return call recordings that match the searching criteria specified by the new configuration `call_recording_extra_query_string`.
 - Support for Voice Recording reports.
 - Support for enabling the user to view SR Service error messages.
 - Support for filtering metadata fields from ICON.
 - Support for a new SR Service report that summarizes all the Screen Recording client connections.
 - ADDP support for the configuration server in the Recording Crypto Server (RCS).
 - Documentation of the Recording Processor Script (RPS) error logs and actions.
-

- Sizing Tool update that now includes numbers for decryption of media through HTCC.
- The SR Service installation package is now signed.

Some of the primary new features added in release **8.5.209.01**—February 1, 2016:

- Citrix is now supported on Windows 2008 R2 for Screen Recording Service.
- Support for Screen Recording Service on Windows 10 (32-bit + 64-bit).
- Support for Screen Recording Service Windows 8 / 8.1 32-bit.
- Recording Processor Script now supports Red Hat 7.

Some of the primary new features added in release **8.5.208.01**—December 18, 2015:

- Information about minimum recommended versions was added to the GIR Solution Guide.
- API documentation about recording search and playback is now published.

Some of the primary new features added in release **8.5.207.01**—October 2, 2015:

- Support for Screen Recording Service on Windows 7 and Windows 8.
- Support for **8 kbit/s mp3 compression** for mono voice recording.
- Ability to filter on user data when scheduling purge and backup tasks.
- Ability to **audit** Media Lifecycle Management.
- Ability to **decrypt** media files.
- Ability to **capture** the entire audio of a call.

Some of the primary new features added in release **8.5.206.01**—June 30, 2015:

- Ability to filter on call type when scheduling purge and backup tasks.

Some of the primary new features added in release **8.5.205.01**—April 15, 2015:

- Support for **backing up and purging** recording files.
- Support for a single **ICON Database** configured to service multiple sites.

- Support for **screen recording client** for Workspace Desktop Edition.
- Enhanced SpeechMiner player.

Some of the primary new features added in release **8.5.204.01**—January 16, 2015:

- Support for screen recording when in **After Call Work** state.
- Support for screen recording client **authentication** for Workspace Web Edition.
- Support for MP3 files that use 16 kbit/s for bitrate compression.
- Support for dual monitor screen recording.

Some of the primary new features added in release **8.5.203.01**—November 3, 2014:

- Support for screen recording encryption.
- Support for percentage based screen recording for voice agents.
- Support for Windows 2012 (except for Web Services).
- Support for Oracle ICON databases.
- Ability to display the number of recorded segments within multi-segmented interactions.
- Ability to display the Business Terminology in SpeechMiner's Metadata Manager.
- Support for Screen Recording Client authentication with the **Screen Recording API**.

Some of the primary new features added in release **8.5.200.01**—September 18, 2014:

- Support for **screen recording**.
- Support for **multiple screen recording storage** locations within a single tenant environment.

Recording Methods

This section describes the recording methods used by Genesys Interaction Recording. For more information about configuring SIP Server settings for GIR, see [Deploying SIP Server for GIR](#).

Full-time Recording or Total Recording

To start recording based on static DN-level settings, set the **[TServer] record** parameter to `true` in any of the following:

- Extension or ACD Position DN for agent-side recording—The recording will be stopped when the call is transferred to a different agent.
- Agent Login for agent-side recording

Important

Not supported for deployments using SIP Cluster.

- Trunk Group DN to record IVR interaction—The recording will be stopped when the IVR hands-off the call to an agent.
- Voice Treatment Port DN to record GVP interaction

When the recording is paused, the recording file is padded with silence for the duration of the period.

Selective Recording

To enable selective recording, configure the following:

Configure the **TRouteCall** request in the routing strategy to include the key **record**, with the value:

- `destination`—Starts destination recording.

Recording can be stopped immediately by using the following value for the key **record**:

- `disable_destination`—Turns off destination recording.

For an inbound or outbound call, `destination` corresponds to agent-side recording.

For agent recording, recording stops when the agent transfers the call, unless recording is setup on the new routing point.

You can also add the following optional key-value pairs in the extensions:

- **id**—A string used to add an identifier to the recording session; must be globally unique. If not configured, Media Server constructs a unique identifier itself.
- **dest**—A string used to override the default location of the recording to be imported to SpeechMiner (see [Configure GVP](#) and [Configure the SpeechMiner components](#)).

When the recording is paused, the recording file is padded with silence for the duration of the period.

Dynamic Recording

A T-Library client that is registered with a DN can send a **RequestPrivateService** to start, pause, resume, and stop a recording. The client must include recording-related parameters in the **RequestPrivateService** request that it sends to SIP Server. Note that if a recording is triggered due to Full-time Recording or due to Selective Recording, it can still be controlled using the **RequestPrivateService** request.

When the recording is paused, the recording file is padded with silence for the duration of the period.

To control dynamic recording with **TPrivateService**, the request uses the following parameters:

Attribute	Value
PrivateMsgID	<p>Specifies the type of recording operation to be performed:</p> <ul style="list-style-type: none"> • GSIP_RECORD_START (3013)—Starts the recording. • GSIP_RECORD_STOP (3014)—Stop the recording. • GSIP_RECORD_PAUSE (3015)—Pause the recording. • GSIP_RECORD_RESUME (3016)—Resume the recording.
ThisDN	Specifies the DN on behalf of which the recording operation is requested. This DN must be registered by the T-Library client
ConnectionID	References the ID for the call that is currently being recorded.
Extensions	<p>Specifies key-value pairs used to control the recording session:</p> <ul style="list-style-type: none"> • record—Set to source or destination. • partitions—Set the list of partitions to be assigned for this recording. The list is comma delimited.

Attribute	Value
	These parameter will appear in the recording session.For example, AttributeExtensions... 'record' 'source' 'partitions' 'sales'
Reasons	Specifies any reasons. Processed the same as for all other T-Library requests.

Important

When an Agent is configured to capture **screen recordings**, and he/she starts, stops, pauses, and resumes voice calls, the screen recordings will do the same.

How Recording Works

Overview

Generally, the basic call flow for agent recording is as follows:

1. Call recording is initiated in one of the following ways:
 - **Static configuration**—Recording is enabled through static DN-level configuration of the agent (Extension DN or Agent Login). **Important:** Agent Login is not supported for deployments that are using SIP Cluster.
 - **Routing strategy**—The routing strategy initiates recording through the `TRouteCall` request that it sends to SIP Server.
 - **T-Library client**—A **T-Library** client initiates recording through a `TPrivateService` request that it sends to SIP Server.
2. Based on this trigger, SIP Server builds a request URI that includes key recording-related parameters. It then sends this request URI in an INVITE to Resource Manager.
3. Resource Manager determines the right Media Control Platform (MCP) to provide the service, and then forwards the INVITE to the selected MCP, to set up the service.
4. SIP Server sends additional Media Server Markup Language (MSML) instructions in SIP INFO messages, telling Media Server to start the recording.

For additional control over the established recording session, the T-Library `TPrivateService` request can be used to initiate new actions—for example, to pause or resume recording. SIP Server forwards the resulting MSML instructions in new INFO messages.

IVR recording is also supported when using VoiceXML recording control, or when using agent recording through the SIP Server.

IVR Recording

The GIR ecosystem can be used to record a GVP-based IVR application. There are two ways that a call within an IVR application may be recorded:

- The entire IVR application can be recorded using the agent recording method through SIP Server DN configuration.
- Part or all of the IVR application can be recorded based on VoiceXML application-level control. This provides the ability to pause and resume recording in order to mask sensitive information collected by or played by the IVR application.

For additional information, refer to [IVR Recording](#) .

Call Recording Interactions

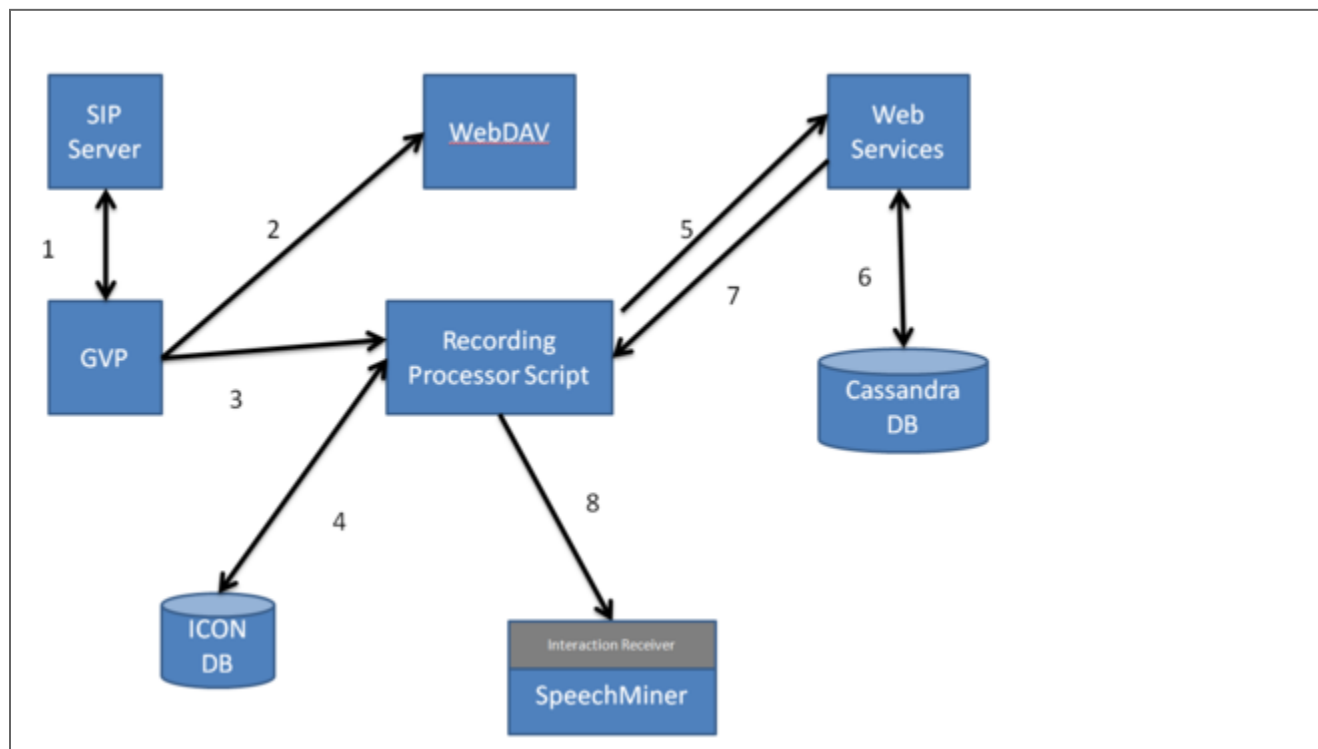
If you want help to resolve conflict, ensure that your service is consistent, or ensure that you conform to your compliance standards, consider this solution.

[+] Show diagrams that will help you determine which options are best suited for your needs.

Important

In the diagrams below, Web Services and Interaction Recording Web Services (RWS) are used interchangeably. The Voice Processor can be used instead of the Recording Processor Script (RPS).

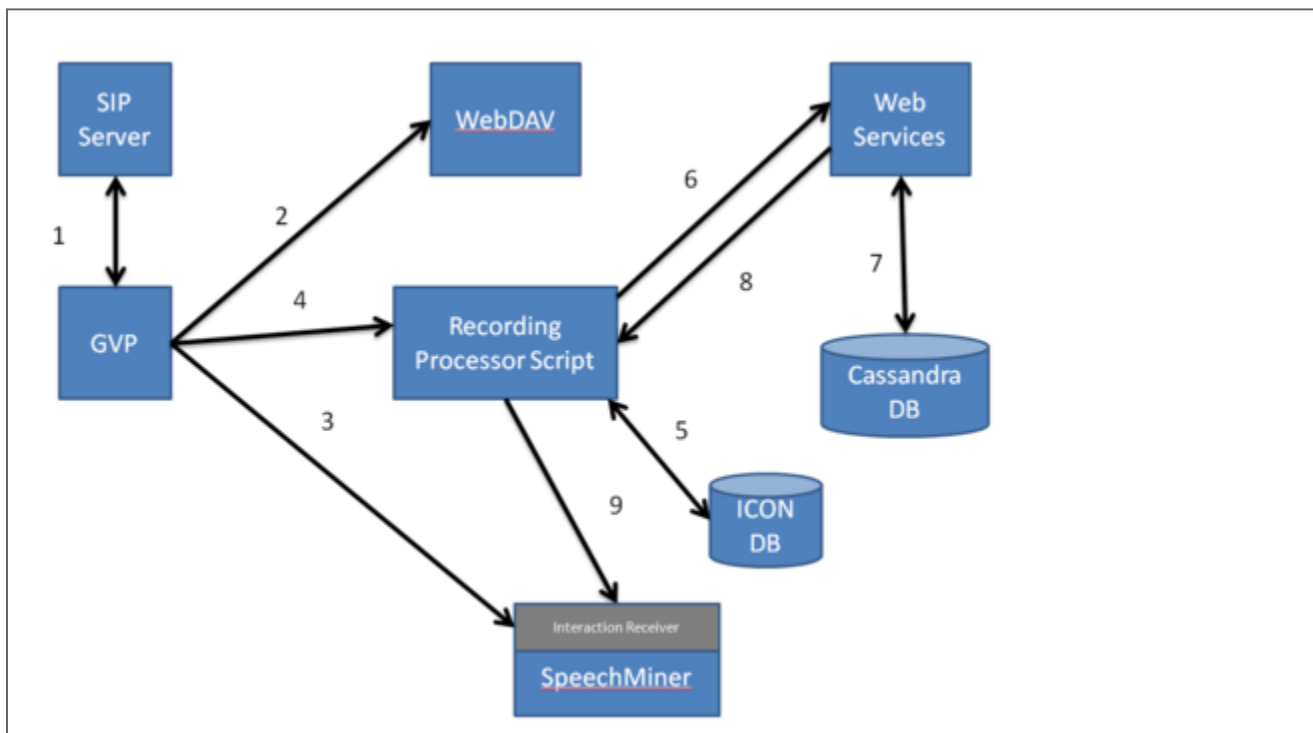
In Recording-Only Mode



1. When a call enters the contact center, SIP Server sends the call to Genesys Voice Platform (GVP) to process the recording.

2. GVP sends the mp3 recording file to WebDAV to store.
3. When GVP receives a successful notification that the file is stored, it sends the recording's metadata information to the Recording Processor Script.
4. The Recording Processor Script parses the metadata received from GVP and retrieves the corresponding metadata from the ICON database that was previously provided by the SIP Servers.
5. Once the metadata is retrieved from the ICON database successfully, the Recording Processor Script sends the information about the recording to Interaction Recording Web Services (Web Services if you're using version 8.5.210.02 or earlier).
6. Interaction Recording Web Services (Web Services) stores the recording information to the Cassandra database.
7. Interaction Recording Web Services (Web Services) tells the Recording Processor Script that the recording information is stored.
8. The Recording Processor Script sends the recording information to SpeechMiner Interaction Receiver.

In Recording and Analytics Mode

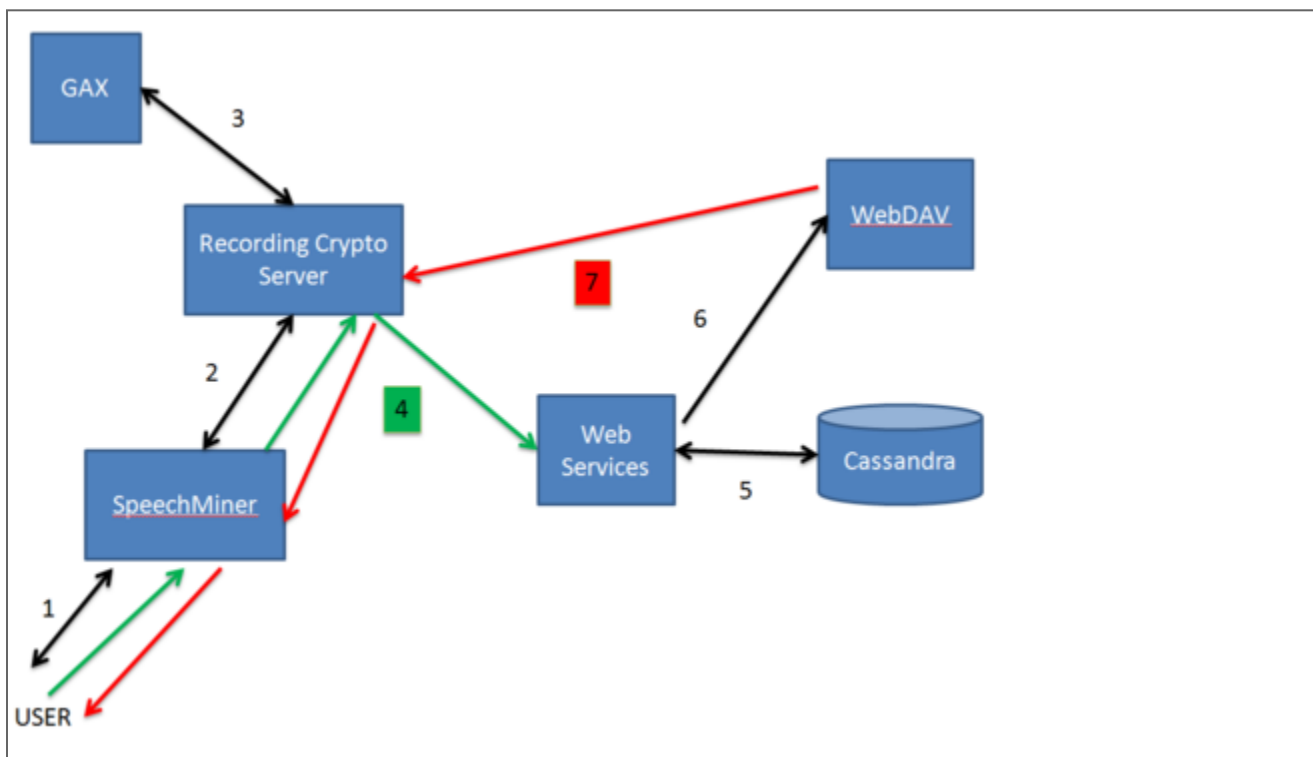


1. When a call enters the contact center, SIP Server sends the call to Genesys Voice Platform (GVP) to process the recording.
2. GVP sends the mp3 recording file to WebDAV to store.
3. GVP also sends a .wav recording to SpeechMiner's Interaction Receiver to analyse.
4. When GVP receives a successful notification that the file is stored, it sends the recording's metadata

information to the Recording Processor Script.

5. The Recording Processor Script parses the metadata received from GVP and retrieves the corresponding metadata from the ICON database that was previously provided by the SIP Servers.
6. Once the metadata is retrieved from the ICON database successfully, the Recording Processor Script sends the information about the recording to Interaction Recording Web Services (Web Services if you're using version 8.5.210.02 or earlier).
7. Interaction Recording Web Services (Web Services) stores the recording information to the Cassandra database.
8. Interaction Recording Web Services (Web Services) tells the Recording Processor Script that the recording information is stored.
9. The Recording Processor Script sends the recording information to SpeechMiner Interaction Receiver.

Playing Back the Inbound Interactions



1. Log into the SpeechMiner UI with your Genesys user credentials.
2. In the background, SpeechMiner opens a two way session with the Recording Crypto Server.
3. Then, the Recording Crypto Server checks in the Genesys Configuration database that your username and password are correct.
4. After your credentials have been successfully verified, you can search for your recording. Select your recording. In the background, SpeechMiner makes an API call to the Recording Crypto Server and tells Web Services to find the recording file.
5. Interaction Recording Web Services (Web Services if you're using version 8.5.210.02 or earlier) then searches the Cassandra database for the file you selected.

6. When the recording file is found in the database, Interaction Recording Web Services (Web Services) tells WebDAV to retrieve the file from storage.
7. WebDAV streams the recording file to you through the Recording Crypto Server and the SpeechMiner UI.

Screen Recording Interactions

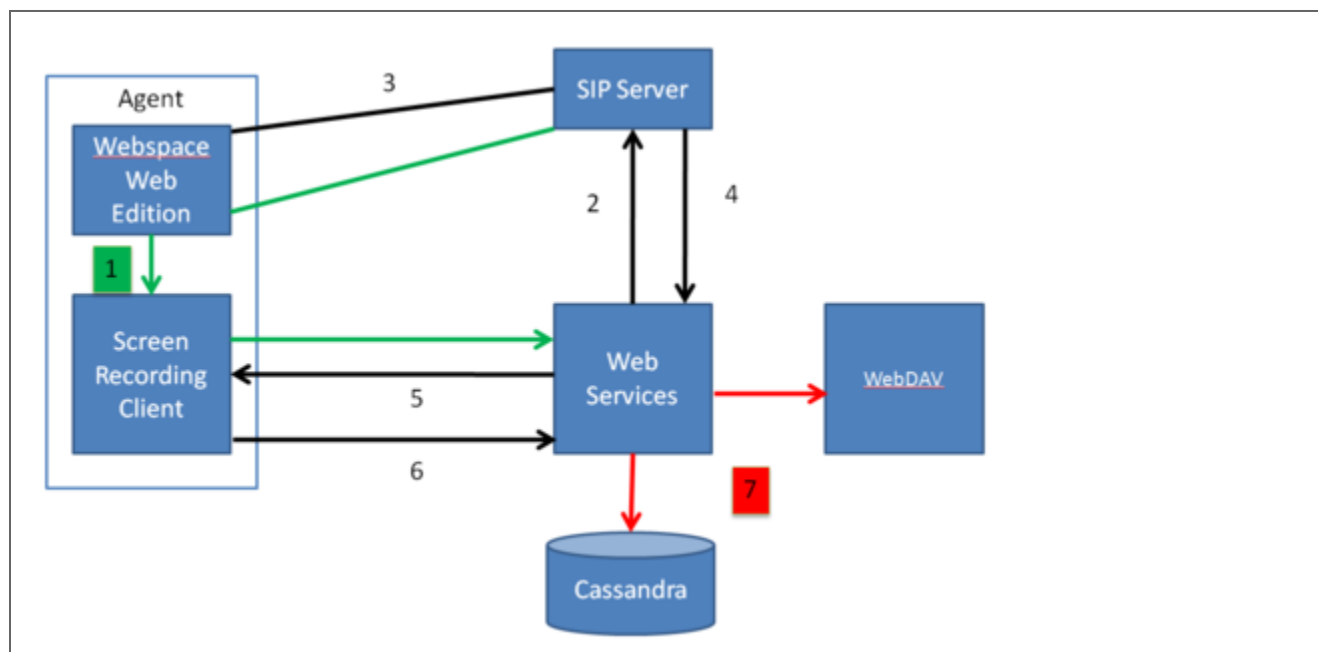
If you want help to ensure that your service is consistent consider using the screen recording solution.

[+] Show diagrams that will help you determine which options are best suited for your needs.

Important

In the diagrams below, Web Services and Interaction Recording Web Services are used interchangeably.

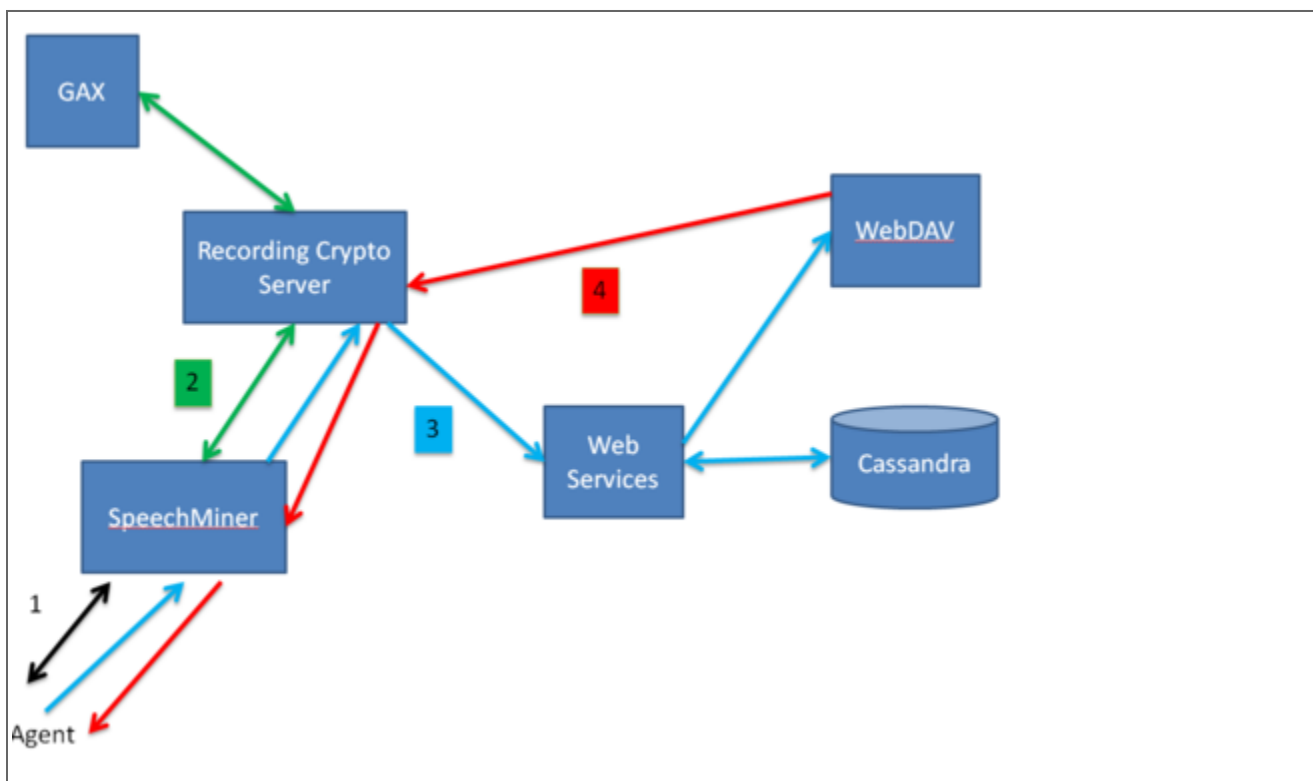
In Recording-Only Mode



1. When you log into Workspace Web Edition, Workspace Web Edition notifies Interaction Recording Web Services (Web Services if you're using version 8.5.210.02 or earlier) about the request to login (through a HTTP load balancer).
2. Workspace Web Edition initiates a connection to the Screen Recording Client.
3. When you click Record, the Screen Recording Client, through the load balancer, tells Interaction Recording Web Services (Web Services) to listen for the agent events via SIP Server.

4. SIP Server tells Interaction Recording Web Services (Web Services) that a voice recording has started, and Web services notifies the Screen Recording Client.
5. When you stop the voice recording, SIP Server notifies Interaction Recording Web Services (Web Services), and Interaction Recording Web Services (Web Services) notifies the Screen Recording Client.
6. The Screen Recording client uploads the recording to Interaction Recording Web Services (Web Services).
7. Interaction Recording Web Services (Web Services) sends the recording file to WebDAV for storage and writes the metadata to the Cassandra database.

Playing Back the Interactions



1. Log into the SpeechMiner UI with your Genesys user credentials.
2. In the background, SpeechMiner UI initiates a session with Interaction Recording Web Services (Web Services) if you're using version 8.5.210.02 or earlier) and Recording Crypto Server. Interaction Recording Web Services (Web Services) and Recording Crypto Server also checks for your user credentials and permissions.

3. After your credentials have been successfully verified, you can search for your recording using the UI's Interactions view, or Screen Recording view.
 - In the **Interactions** view—If the voice interaction has an associated screen recording, SpeechMiner UI presents a thumbnail of the screen on the media player.
 - When you play the recording, SpeechMiner UI retrieves the recording file from Recording Crypto Server if the recording is encrypted.
 - The Recording Crypto Server retrieves the recording via Interaction Recording Web Services (Web Services) and WebDAV, and decrypts the recording.
4. In the **Screen Recording** view—Search for your screen recording file. The Screen Recording Client tells Interaction Recording Web Services (Web Services) to find the recording.
 - When the file is successfully found, select the recording file to play and the Recording Crypto Server retrieves the recording if the recording is encrypted.
 - The Recording Crypto Server retrieves the recording via Interaction Recording Web Services (Web Services) and WebDAV and decrypts the recording.
5. WebDAV streams the recording file to you through the Recording Crypto Server and the SpeechMiner UI.

Supported Media File Formats

MSML-based call recording supports MP3 (8 kbit/s mono, 16 kbit/s stereo, or 32 kbit/s stereo) for playback, and WAV G.711 for SpeechMiner analytics purposes.

Important

- In a Recording Only deployment the system must be configured to store the file in MP3 format. This is the default configuration for the IVR profile. When using Analytics deployments, the IVR profile must be updated to include an additional recording setting. For details, see [Deploying Genesys Voice Platform for GIR](#).
- If a Muxed screen recording contains 16 kbit/s stereo audio, it cannot be played in Internet Explorer under Windows 7. In this scenario, you can either configure the MCP to use 32 kbit/s stereo audio or use a different Web browser for media file playback with Windows 7.

Mid-Call Control of the Recording Session

Using TPrivateService requests, T-Library clients can control, in real-time, an ongoing recording session. The client can pause, resume, or stop the recording. SIP Server translates recording-related parameters from the request to INFO messages that it sends to Genesys Media Server.

Supported mid-call actions are as follows:

- Stop the recording.
- Pause the recording.
- Resume a paused recording.

To control mid-call recording, the TPrivateService request uses the following parameters:

Attribute	Value
PrivateMsgID	Specifies the type of recording operation to be performed: <ul style="list-style-type: none">• GSIP_RECORD_STOP (3014)—Stops the recording.• GSIP_RECORD_PAUSE (3015)—Pauses the recording.• GSIP_RECORD_RESUME (3016)—Resumes the recording.
ThisDN	Specifies the DN on behalf of which the recording operation is requested. This DN must be registered by the T-Library client.
ConnectionID	References the ID for the call that is currently being recorded.
Reasons	Specifies any reasons. Processed the same as for all other T-Library requests.

Recording During Transfers and Conferences

SIP Server supports continuous recording for calls that are transferred or added to a conference. Once recording is initiated (through DN configuration, routing strategy, or by T-Library client), recording will continue for as long as one party that is set for recording remains in the call. Recording ends when no more recording-enabled parties are left.

Screen Recording

The screen recording feature for Genesys Interaction Recording allows agents to capture both voice and non-voice interactions that are currently being played on their computer monitors. The screen recording client interfaces with Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). The latter provides the following tasks:

- Authenticate and authorize screen recording clients
- Register client connections and associate with agents

-
- Coordinate recording controls
 - Receive screen recording files from clients
 - Store screen recording files to recording storage
 - Store screen recording metadata
 - Provide software updates to the clients

For more information, see [Screen Recording Architecture](#).

Important

For blended agents that are configured to support the handling of both voice and non-voice interactions, GIR will perform screen recording of voice interactions only.

Getting Started with Genesys Interaction Recording

Welcome to Genesys Interaction Recording (GIR). GIR helps to optimize your workforce performance and customer experiences by allowing you to record, save, and play back your customer interactions.

Each **product and feature** page will tell you how to install and configure the component to enable recording. Once everything is in place, you can record an interaction, then listen to it.

If you want to know more about GIR itself and where it fits into your Genesys solution, you can check out the following topics, or check out the **videos**:

- [About Genesys Interaction Recording](#)
- [How Recording Works](#)
- [Architecture and Features](#)

Genesys Interaction Recording (GIR) can be deployed in a single tenant environment or in a multi-tenant environment. To successfully deploy GIR you must follow the instructions provided in the order that they appear.

- [Deploy GIR in a Single Tenant Environment](#)
- [Deploy GIR in a Multi Tenant Environment](#)

Videos

This high level overview of the GIR Architecture talks about the components that are involved with capturing, searching for, and playing back your recordings (6:48).

Important

In the videos below, the GIR Voice Processor can be used instead of the Recording Processor Script (RPS).

[Link to video](#)

Or, you can also watch these shorter videos describing each component separately.

[Link to video](#)

[Link to video](#)

[Link to video](#)

Before You Start

The first thing you need to do is check that the following Genesys minimum versions, components and features are installed and working.

Minimum Required Versions

Before you install and configure the Genesys Interaction Recording (GIR), verify that you have the required minimum Genesys versions. For detailed information, refer to [Minimum Required Versions](#).

Genesys Components

[Interaction Recording Web Services \(RWS\)](#) (or [Web Services and Applications](#) if you're using version 8.5.210.02 or earlier)

[SIP Server](#)

[Genesys Voice Platform](#)

[Interaction Concentrator \(ICON\)](#)

[Recording Processor Script](#) or [Voice Processor](#)

[Recording Crypto Server](#)

[Recording Plug-in](#)

[Speech and Text Analytics \(SpeechMiner\)](#)

[Workspace Desktop Edition](#)

GIR Features

[Geo-Location](#)

[Audio Tones](#)

[Security \(TLS\)](#)

[Access Control](#)

[Encrypting and Provisioning Certificates](#)

[Enable Call Recording](#)

[Enable Screen Recording](#)

[Media Life Cycle Management](#)

Deploying Genesys Interaction Recording in a Single-Tenant Deployment

Installation considerations

This section provides the deployment steps required to configure GIR with a single-tenant Configuration Server.

Important

Each component should be sized according to the Tenant sizing needs.

Once the deployment steps are completed, the tenant will include the following items:

[+] Show items.

- Users (that is, only users within a tenant are allowed to access GIR).
- Permissions
- Access control of voice and screen recordings for search, view and playback.
- The ability to view the agent hierarchy.
- Recording conditions (full time recording, selective recording).
- Recording retention policies.
- Recording storage location and policies.
- Recording backup policies.
- Audit logs.
- Use of encryption keys.
- Administration of encryption keys.
- Screen recording policies.
- Quality management functionalities.

Tenant components

The following is a list of the components that can be deployed as tenant-specific components:

[+] Show tenant components.

-
- SIP Server
 - ICON
 - ICON DB
 - Recording storage – WebDAV server
 - Muxer
 - Recording Processor Script or Voice Processor
 - Recording Crypto Server
 - Interaction Recording Web Services (RWS)
 - SpeechMiner
 - Database instance
 - Tenant reporting on SQL Server Reporting Services
 - Interaction Receiver
 - UPlatform
 - SpeechMiner Web
 - Screen recording service (runs on the agent's PC)

Single-Tenant Deployment

This section provides the tasks required to install and configure the Genesys components and features for Genesys Interaction Recording.

To successfully deploy GIR in a single-tenant deployment, you must perform the following procedures in the order presented:

1. [Genesys Administrator Extension](#)
 2. [Interaction Recording Web Services \(RWS\)](#) (or [Web Services and Applications](#) if you're using version 8.5.210.02 or earlier)
 3. [SIP Server](#)
 4. [Interaction Concentrator \(ICON\)](#)
 5. [Recording Plug-in for GAX](#)
 6. [Recording Processor Script](#) or [Voice Processor](#)
 7. [Recording Crypto Server](#)
 8. [Genesys Voice Platform](#)
 9. [Screen Recording Service](#)
 10. [Recording Muxer Script](#)
 11. [User Access](#)
 12. [Workspace Desktop Edition](#)
-

13. [Speech and Text Analytics \(SpeechMiner\)](#)
14. [Security \(TLS\)](#)
15. [Media Lifecycle Management](#)
16. [Geo-Location](#)
17. [Audio Tones](#)
18. [Encrypting and Provisioning Certificates](#)
19. [Enable Call Recording](#)
20. [Enable Screen Recording](#)
21. [Recording Storage Folder Hierarchy](#)
22. [Load Balancing](#)

Deploying Interaction Recording Web Services (RWS)

Warning

The content on this page only applies to versions of Genesys Interaction Recording newer than 8.5.210.02. If you're using an earlier version, you'll need to install Web Services and Applications instead. See [Deploying Web Services and Applications for GIR](#) for details.

If you upgrade to Interaction Recording Web Services (RWS), it does not provide API support for non-GIR related Web Services, such as Workspace Web Edition.

Genesys Interaction Recording (GIR) needs the Interaction Recording Web Services component to store and manage recording files.

Complete the steps below to install and configure Interaction Recording Web Services and its supporting components:

1. [Review the Prerequisites](#). Make sure all supporting components are installed and configured.
2. [Deploy Cassandra](#)
3. [WebDAV Requirements](#)
4. [Initialize Cassandra](#)
5. [Elasticsearch](#)
6. [Install Interaction Recording Web Services \(RWS\)](#)
7. [Deploy the web application](#)
8. [Configure Interaction Recording Web Services](#)
9. [Configure additional security \(optional\)](#)
10. [Start and test Interaction Recording Web Services](#)
11. [Configure your required features](#)

Prerequisites

Before deploying and configuring Interaction Recording Web Services, make sure your system meets the following minimum requirements:

OS Requirements

See the [Genesys Interaction Recording](#) page in the *Genesys Supported Operating Environment Reference* for more detailed information and a list of all supported operating systems.

Java Requirements

- From Interaction Recording Web Services version 8.5.205.69 (or higher), you have installed the latest JDK 17 (64-bit for Linux). Alternatively, you can also install the latest standalone JRE 17 (64-bit for Linux). You can choose to download the software from an OpenJDK version of the software.
- From Interaction Recording Web Services version 8.5.205.65 (or lower), you have installed the latest JDK 8 (64-bit for Linux). Alternatively, you can also install the latest standalone JRE 8 (64-bit for Linux). You can choose to download the software from Oracle or obtain an OpenJDK version of the software.

Cassandra Requirements

Interaction Recording Web Services stores information about call and screen recordings in a Cassandra database. For each contact center, distinct column families with unique names exist for storing recording information. These column families are created when the contact center is created, and deleted if the contact center is deleted.

Important

- Interaction Recording Web Services deletes column families only if they do not contain any call recordings; otherwise they should be deleted manually from Cassandra using the `cqlsh` utility tool.
- Interaction Recording Web Services and Web Services and Applications share the same Cassandra instance within the same deployment. If you are using Interaction Recording Web Services with Web Services and Applications in the same environment, verify that your Cassandra version is the same for both components and all nodes.

Interaction Recording Web Services requires that your environment includes Cassandra 1.2 or 2.2. Genesys recommends Cassandra version 2.2. Complete the steps in these procedures below to install and configure Cassandra 2.2:

- [Deploying Cassandra 2.2](#)
 - [Installing and Configuring Cassandra 2.2](#)
 - [Upgrading to Cassandra 2.2](#)

Interaction Recording Web Services 8.5.500.17 (or higher) requires that your environment includes Cassandra 4.1. Complete the steps in these procedures below to upgrade and configure Cassandra 4.1:

- [Upgrading Cassandra to 4.1](#)

Interaction Recording Web Services 8.5.500.30 (or higher) supports install of Cassandra 4.1. Complete the steps in these procedures below to install Cassandra 4.1:

- [Installing and Configuring Cassandra 4.1](#)

Genesys Environment

Before installing and configuring the [Interaction Recording Web Services](#) 8.5.500.17 (or higher), you must have the following prerequisites:

- A [Recording Crypto Server](#) 8.5.095.22 (or higher) instance to decrypt the encrypted recordings.
- A [Recording Plugin for GAX version](#) 8.5.500.08 (or higher) for certificate updation and MLM configuration.

For more information about the required Genesys environment for GIR, refer to the [Minimum Recommended Versions](#).

Next Step

- [WebDAV Requirements](#)

Deploying Cassandra

Before you start installing and configuring Genesys Interaction Recording (GIR), you must first install and configure Cassandra. GIR supports Cassandra version 1.2 or 2.2 or 4.1.

For new deployments, use Cassandra 4.1. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the [Cassandra 4.1 documentation](#).

For general instructions and guidelines, select one of the following links:

- [Installing and Configuring Cassandra 4.1](#)
- [Upgrading to Cassandra 4.1](#)

Installing and Configuring Cassandra

Cassandra 4

Installing Cassandra4

Complete this procedure for each Cassandra node.

Prerequisites

- For new deployments, we recommend Cassandra 4.1.x. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the [Cassandra 4.1 documentation](#).
- You have installed the latest supported openJDK 8.

Start

1. [Download the latest 4.1.x version of Cassandra](#).
2. Copy the Cassandra archive to the installation directory. For example, **/usr/local**
3. Use a tar utility to extract the files. For example, `tar -zxvf apache-cassandra-4.1.9-bin.tar.gz`
4. Add directories for data, commitlog, and saved_caches. You can create these directories anywhere or in the default locations configured in the ***Cassandra_install_dir/conf/cassandra.yaml*** file. For example:
 - **/var/lib/cassandra/data**
 - **/var/lib/cassandra/commitlog**
 - **/var/lib/cassandra/saved_caches**

End

Configuring Cassandra4

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development: 1 Cassandra node (appropriate for a development or lab environment)
- Single Datacenter: 1 datacenter with a minimum of three Cassandra nodes
- Two Datacenters: 2 datacenters with a minimum of three Cassandra nodes

Important

For more complex Cassandra deployments, please consult with Genesys

Prerequisites

- [Installing Cassandra4](#)

Follow one of the procedures that matches your deployment scenario.

Important

The files modified in this procedure are typically found in the ***Cassandra_install_dir/conf*** directory.

Development

Configuring Cassandra (1 Cassandra node)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set `cluster_name` to the desired name.
 - b. Set `seeds` to the list of host name of the node. For example: `- seeds: "127.0.0.1"`
 - c. Set `listen_address` and `broadcast_address` to the IP of the node.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra4](#).
 - e. Set the `endpoint_snitch` parameter to `SimpleSnitch` (for development only).
6. Save your changes and close the file.

End

Single Datacenter

Configuring Cassandra (1 datacenter)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set the `cluster_name`. It must be the same name on all nodes.
 - b. Set `seeds` to the list of host names of all nodes. For example: `- seeds: "node1, node2, node3"`

- c. Set `listen_address` and `broadcast_address` to the IP of the node.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra4](#).
 - e. Set the `start_rpc` parameter to `true`.
 - f. Change `endpoint_snitch` to `GossipingPropertyFileSnitch`.
7. Save your changes and close the file.
 8. Open the **`cassandra-rackdc.properties`** file and update for your cluster topology. For each node in your cluster, add the following line:

```
dc=[datacenter] rack=[rackname]
```

Where:

- `[datacenter]` is the name of the datacenter for this node.
- `[rackname]` is the name of the rack for this node.

The following is a sample **`cassandra-rackdc.properties`** file for a Single Datacenter scenario:

```
dc=datacenter1 rack=rack1
```

9. Save your changes and close the file.

End

Two Datacenters

Configuring Cassandra (2 datacenters)

Start

1. Modify the **`cassandra.yaml`** file:
 - a. Set the `cluster_name`. It must be the same name on all nodes.
 - b. Set `seeds` to the list of host names of all nodes. For example: `- seeds: "node1, node2, node3, node4, node5, node6"`
 - c. Set `listen_address` and `broadcast_address` to the host name.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra4](#).
 - e. Change `endpoint_snitch` to `GossipingPropertyFileSnitch`.
6. Save your changes and close the file.
7. Open the **`cassandra-rackdc.properties`** file and update for your cluster topology. For each node in your cluster, add the following line:

```
dc=[datacenter] rack=[rackname]
```

Where:

- `[datacenter]` is the name of the datacenter for this node.

- *[rackname]* is the name of the rack for this node.

The following is a sample **cassandra-rackdc.properties** file for a Two Datacenter scenario:

For datacenter1 nodes:

```
dc=datacenter1 rack=rack1
```

For datacenter2 nodes:

```
dc=datacenter2 rack=rack1
```

8. Save your changes and close the file.

End

Verifying the Cassandra4 installation

Prerequisites

- [Configuring Cassandra](#)

Start

1. Start all Cassandra nodes using the following command: *Cassandra_install_dir/bin/cassandra -R*
2. Use the nodetool utility to verify that all nodes are connected by entering the following command: *Cassandra_install_dir/bin/nodetool status*

The following is sample output for a Single Datacenter scenario with three Cassandra nodes:

```
/genesys/apache-cassandra-4.1.4/bin$ ./nodetool status
Address      DC           Rack  Status  State  Load      Owns      Token
192.0.2.10   datacenter1  rack1 Up       Normal 14.97 MB  100.00%   -9223372036854775808
192.0.2.11   datacenter1  rack1 Up       Normal 14.97 MB  100.00%   -3074457345618258603
192.0.2.12   datacenter1  rack1 Up       Normal 14.97 MB  100.00%   3074457345618258602
```

The following is sample output for a Development scenario with a single Cassandra node:

```
/genesys/apache-cassandra-4.1.4/bin$ ./nodetool status
Address      DC           Rack  Status  State  Load      Effective-
Ownership Token
127.0.0.1     datacenter1  rack1 Up       Normal 1.89 MB
100.00%       76880863635469966884037445232169973201
```

End

Cassandra 2

Installing Cassandra2

Complete this procedure for each Cassandra node.

Prerequisites

- For new deployments, we recommend Cassandra 2.2. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the [Cassandra 2.2 documentation](#).
- You have installed the latest [Java SE Development Toolkit 8](#). For more information, refer to the [Java documentation](#).

Start

1. [Download the latest 2.2.x version of Cassandra](#).
2. Copy the Cassandra archive to the installation directory. For example, `/usr/local`
3. Use a tar utility to extract the files. For example, `tar -zxvf apache-cassandra-2.2.7-bin.tar.gz`
4. Add directories for data, commitlog, and saved_caches. You can create these directories anywhere or in the default locations configured in the **`Cassandra_install_dir/conf/cassandra.yaml`** file. For example:
 - `/var/lib/cassandra/data`
 - `/var/lib/cassandra/commitlog`
 - `/var/lib/cassandra/saved_caches`

End

Configuring Cassandra2

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development: 1 Cassandra node (appropriate for a development or lab environment)
- Single Datacenter: 1 datacenter with a minimum of three Cassandra nodes
- Two Datacenters: 2 datacenters with a minimum of three Cassandra nodes

Important

For more complex Cassandra deployments, please consult with Genesys

Prerequisites

- [Installing Cassandra4](#)

Follow one of the procedures that matches your deployment scenario.

Important

The files modified in this procedure are typically found in the ***Cassandra_install_dir/conf*** directory.

Development

Configuring Cassandra (1 Cassandra node)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set seeds to the list of host name of the node. For example: -seeds: "127.0.0.1"
 - b. Set listen_address and rpc_address to the host name.
 - c. Set data_file_directories, commitlog_directory, and saved_caches_directory to the directories you created in Step 4 of [Installing Cassandra](#).
 - d. Set the start_rpc parameter to true.
5. Save your changes and close the file.

End

Single Datacenter

Configuring Cassandra (1 datacenter)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set the cluster_name. It must be the same name on all nodes.
 - b. Set seeds to the list of host names of all nodes. For example: -seeds: "node1, node2, node3"
 - c. Set listen_address and rpc_address to the host name.
 - d. Set data_file_directories, commitlog_directory, and saved_caches_directory to the directories you created in Step 4 of [Installing Cassandra](#).
 - e. Set the start_rpc parameter to true.
 - f. Change endpoint_snitch to PropertyFileSnitch.
7. Save your changes and close the file.
8. Open the **cassandra-topology.properties** file and update for your cluster topology. For each node in your cluster, add the following line:

```
[node]=[datacenter]:[rack]
```

Where:

- *[node]* is the IP address of the node.
- *[datacenter]* is the name of the datacenter for this node.
- *[rack]* is the name of the rack for this node.

The following is a sample **cassandra-topology.properties** file for a Single Datacenter scenario:

```
192.0.2.10=datacenter1:rack1
192.0.2.11=datacenter1:rack1
192.0.2.12=datacenter1:rack1
```

9. Save your changes and close the file.

End

Two Datacenters

Configuring Cassandra (2 datacenters)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set the `cluster_name`. It must be the same name on all nodes.
 - b. Set seeds to the list of host names of all nodes. For example: `-seeds: "node1, node2, node3, node4, node5, node6"`
 - c. Set `listen_address` and `rpc_address` to the host name.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra](#).
 - e. Set the `start_rpc` parameter to `true`.
 - f. Change `endpoint_snitch` to `PropertyFileSnitch`.
7. Save your changes and close the file.
8. Open the **cassandra-topology.properties** file and update for your cluster topology. For each node in your cluster, add the following line:

```
[node]=[datacenter]:[rack]
```

Where:

- *[node]* is the IP address of the node.
- *[datacenter]* is the name of the datacenter for this node.
- *[rack]* is the name of the rack for this node.

The following is a sample **cassandra-topology.properties** file for a Two Datacenter scenario:

```
192.0.2.10=datacenter1:rack1
192.0.2.11=datacenter1:rack1
192.0.2.12=datacenter1:rack1
```

```
198.51.100.10=datacenter2:rack1
198.51.100.11=datacenter2:rack1
198.51.100.12=datacenter2:rack1
```

9. Save your changes and close the file.

End

Verifying the Cassandra2 installation

Prerequisites

- [Configuring Cassandra](#)

Start

1. Start all Cassandra nodes using the following command: *Cassandra_install_dir/bin/cassandra*
2. Use the nodetool utility to verify that all nodes are connected by entering the following command: *Cassandra_install_dir/bin/nodetool -h Cassandra_host ring*

The following is sample output for a Single Datacenter scenario with three Cassandra nodes:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool ring
Address      DC           Rack  Status  State  Load      Owns      Token
192.0.2.10   datacenter1 rack1  Up      Normal  14.97 MB  100.00%   -9223372036854775808
192.0.2.11   datacenter1 rack1  Up      Normal  14.97 MB  100.00%   -3074457345618258603
192.0.2.12   datacenter1 rack1  Up      Normal  14.97 MB  100.00%   3074457345618258602
```

The following is sample output for a Development scenario with a single Cassandra node:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool ring
Address      DC           Rack  Status  State  Load      Effective-
Ownership Token
127.0.0.1    datacenter1 rack1  Up      Normal  1.89 MB
100.00%      76880863635469966884037445232169973201
```

End

Upgrading Cassandra to 2.2

Genesys Interaction Recording (GIR) supports Cassandra versions 2.2 and 1.2. If you are using Cassandra 1.2, you can maintain this version or upgrade to Cassandra 2.2.

Directly upgrading from 1.2 to 2.2 is not supported, therefore you need to upgrade your Cassandra versions in several steps. For example 1.2 > 2.0 > 2.1 > 2.2. For more information about upgrading Cassandra, see the **Upgrading Apache Cassandra** section in [Cassandra Upgrade Guide](#).

Important

No Screen Recordings are made during the upgrade of Cassandra.

To minimize the risk of losing screen recordings, upgrade Cassandra during off-hours. During the migration, GIR still operates with voice recording capabilities. After the Cassandra cluster is back in service, you can process the metadata information for voice recordings that was saved during the migration by initiating the recovery procedure of the Recording Processor Script—as described in step 5 below.

1. Stop all Interaction Recording Web Services nodes.
2. Perform Cassandra upgrade according to the Cassandra Upgrade Guide.
3. When configuring Cassandra 2.2 according to the Datastax instructions, you must enable the thrift interface. Set the **start_rpc** parameter to true in the **cassandra.yaml** file
4. Start all Interaction Recording Web Services nodes.
5. After the upgrade, run the [LVR Recovery Script](#) to recover recordings from the **Recording Processor failed** folder and repost the recordings to Interaction Recording Web Services, as failed voice recordings accumulate while the Cassandra cluster is unavailable.

Upgrading Cassandra to 4.1

Important

- If you are using GIR with Workspace Web Edition, refrain from using a shared deployment of Cassandra. This is because Web Services and Applications support Cassandra 2.2 only and do not support Cassandra 4.1. For details, see: [Installing and configuring Cassandra](#).
- Genesys Interaction Recording Web Services (GIR RWS) version 8.5.500.17 (or higher) supports Cassandra 4.1 only.

If you are using Cassandra 2.2, you must upgrade to Cassandra 4.1. Upgrading directly from 2.2.x to 4.1 is not supported. You must upgrade your Cassandra versions incrementally for schema portability, for example, from version 2.2.x -> 3.11.11 -> 4.1.x. Requires Python 3.6 (or higher) and Java 8 for Cassandra 4.1.

For more information about upgrading Cassandra, refer the **Upgrade to Cassandra 4.1** section.

Important

No screen recordings are made during the upgrade of Cassandra.

To minimize the risk of losing screen recordings, upgrade Cassandra during off-hours. It's a best practice to perform a cluster upgrade during a scheduled maintenance window that occurs when application loads are typically lower. During the migration, GIR still operates with voice recording capabilities. After the Cassandra cluster is back in service, you can process the metadata information for voice recordings that was saved during the migration by initiating the recovery procedure of the Recording Processor Script.

Pre-requisites:

1. Ensure that the Recording Processor Script (RPS) or GIR Voice Processor (GIR VP), and Recording Web Services (RWS) nodes debug level logs are stored for any debugging or troubleshooting issues related to migration.
2. Back up all your Cassandra configuration files.
3. Take snapshots of all keyspaces on each existing Cassandra node. For more information, refer [Cassandra Documentation](#).
4. Confirm that your Cassandra environment satisfies all of the necessary [prerequisites](#).

Upgrade to Cassandra 4.1:

The following section provides the instructions for performing in-place upgrade from Cassandra 2.2.X to Cassandra 4.1.X. The recommendation procedure should be followed for quicker rollback and

minimal downtime.

Recommendation:

Warning

Genesys is not responsible for migration of existing data to the replicated Cassandra cluster on premise environments.

1. Create a replica of existing Cassandra 2.2 cluster with the data. For more information refer to [Cassandra Documentation](#).
2. Ensure that replicated Cassandra 2.2 cluster data is validated with existing RWS nodes by updating the following parameters in `application.yaml`.

`nodes`: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>] and `backup_nodes`: [ToBeChangedOrRemoved: <CASSANDRA_BACKUP_DC_NODES>] to respective replicated nodes. For more information, check [RWS Options for cassandraCluster](#).

1. Execute Migration plan on replicated Cassandra 2.2 new Cluster.

Migration Plan:

1. Stop all Interaction Recording Web Services nodes.
2. Run the schema upgrade application. This will be packaged with the RWS IP installation files. This application need to be run against one of the Cassandra 2.2 cluster nodes.

```
java -jar upgrade.jar [options]
```

options:

 - cassandraHost <host> - required, no default
 - thriftPort <port number> - optional, default=9160
 - nativePort <port number> - optional, default=9042
 - keyspace <keyspace name> - optional, default=sipfs
 - datacenter <datacenter name> - optional, default=datacenter1. The required value can be found via "nodetool status".
 - cassandraUser <username> - optional, but required if -cassandraPassword is set in `cassandra.yaml`
 - cassandraPassword <password> - optional, but required if -cassandraUser is set in `cassandra.yaml`
 - useSSL - optional, but required if cassandra is SSL enabled in `cassandra.yaml`
 - truststore <truststore> - optional, but required if -truststore is set in `cassandra.yaml`
 - storePassword <password> - optional, but required if -storePassword is set in `cassandra.yaml`

Example for SSL enabled cassandra: `java -jar upgrade.jar -cassandraHost <hostname> -datacenter DC1 -useSSL -truststore <truststore> -storePassword <password>`

Example for non-SSL enabled cassandra: `java -jar upgrade.jar -cassandraHost <hostname> -datacenter DC1`

Earlier versions of RWS communicated with Cassandra 2 using the thrift protocol. Cassandra has discontinued support for the thrift protocol, so the new version of RWS uses CQL. The upgrade application will make schema changes to the `call_recording_by_contact_center` table and all call/screen recording and log tables referenced by this table so that the data inserted via thrift will be available via CQL.

3. Perform Cassandra upgrade according to the [Cassandra Upgrade Guide](#).
4. When configuring Cassandra 4.1, you must set the following both parameters in the `cassandra.yaml` file:
 - **batch_size_warn_threshold** to 200KiB
 - **batch_size_fail_threshold** to 250KiB
 - Save and apply all the changes.
 These parameters are related to the size of CQL batches. The default for these parameters is quite

low and need to be increased to accommodate RWS inserts and update batches.

5. Upgrade the Interaction Recording Web Service nodes to the IP version 8.5.500.17 (or higher).
6. Configure Cassandra 4.1 parameters in **application.yaml** of Interaction Recording Web Service nodes. For more information, see [Configuring Interaction Recording Web Services](#).
7. Start all Interaction Recording Web Service nodes. Validate the health of RWS and Cassandra.
8. After the upgrade,
 - If using Recording Processor Script(RPS), run the **LVR Recovery Script** to recover recordings from the each **Recording Processor Script failed** folder and repost the recordings to Interaction Recording Web Services, as failed voice recordings accumulate while the Cassandra cluster is unavailable.
 - If using GIR Voice Processor (GIR VP), the GIR Voice Processor automatically retries the recording posts for up to 40 days.

Rollback:

If we are seeing unresolved errors at any phase of the migration, please rollback back the Interaction Recording Web Service nodes to the previous RWS version and previous existing Cassandra 2.2 cluster until the issue is resolved.

Limitation:

If rollback is performed once the migration procedure is done and new voice recordings are directed to Cassandra 4.1, the following scenarios will result in errors :

- Playback of the new recordings
- Backup and purge of the new recordings
- Querying of the new recordings through API.

Configuring WebDAV

Interaction Recording Web Services relies on a Web Distributed Authoring and Versioning (WebDAV) server to store and manage the GIR recording files. WebDAV is an extension of the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on World Wide Web servers. A working group of the Internet Engineering Task Force (IETF) defined WebDAV in RFC 4918.

The following information represents examples of what can be done for WebDAV. Follow these procedures to get a better understanding of what needs to be done when you use a Red Hat Enterprise Linux machine with the Apache HTTP Server.

Important

- This document provides you with basic guidelines on configuring WebDAV on RHEL. If you wish to configure WebDAV on other operating systems or if you have additional questions regarding WebDAV on RHEL, refer to the official documentation from the operating system provider.
- It is recommended that you do not install WebDAV on the same machine as Interaction Recording Web Services (RWS), since numerous deployments already install Cassandra and Elasticsearch on the same host. These are critical components for the operation of RWS. If an additional process such as WebDAV is run on the same machine as RWS, disk I/O operations will be limited and the stability of RWS may be negatively impacted.
- Authentication must be configured on the WebDAV server. This is required to ensure proper storage and management of GIR recording files.

Deploying the WebDAV Server

1. Install Apache HTTP Server and run the following command:

```
yum install httpd
```

2. Edit the **/etc/httpd/conf/httpd.conf** file, and append the following to the end of the file:

```
Alias /recordings /mnt/recordings
<Directory /mnt/recordings>
    Options Indexes MultiViews FollowSymLinks
    EnableSendfile off
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
<Location "/recordings">
```

```
DAV On
AuthType Basic
AuthName "user"
AuthUserFile /var/www/htpasswd
Require valid-user
</Location>
```

3. Open the firewall.

Because Apache HTTP Server is an HTTP server, the incoming default HTTP and/or HTTPS ports (80 and/or 443) must be open to the server.

Important

It is possible to use custom ports by changing the permitted incoming ports in the firewall, the virtual host configuration file, and the URL used to reach the Apache HTTP server.

4. Create the directory to keep the recording files, and set the permission to apache, using the following commands:

```
mkdir /mnt/recordings
chown apache:apache /mnt/recordings
chcon -R -t httpd_sys_content_t /mnt/recordings
```

Important

Due to performance concerns, Genesys does not recommend using a remote directory for WebDAV.

5. Create an Apache HTTP Server user for httpd, and configure the password. The following example creates a user called "user":

```
htpasswd -cm /var/www/htpasswd user
```

Warning

If the Recording Muxer is deployed for screen recording, make sure all WebDAV storages of the same contact center region are using the same username and password.

6. Configure the httpd to start on boot up (and start it now) using the following commands:

```
chkconfig --levels 235 httpd on
service httpd start
```

7. Test the Apache HTTP Server installation:

a. Upload a hello.world file to the Apache HTTP Server using the following command:

```
curl -T hello.world -u user:password http://myserver/recordings/hello.world
```

- b. Using a browser, open the `http://myserver/recordings/hello.world` URL. The browser will request for user credentials.

8. The Apache HTTP Server is installed.

Configuring TLS for the WebDAV Server

To configure TLS for the Apache HTTP Server on RHEL6:

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: `/etc/pki/tls/certs/localhost.crt`
 - Key: `/etc/pki/tls/private/localhost.key`
2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **`/etc/httpd/conf.d/ssl.conf`** file and modify the following lines:
 - `SSLCertificateFile /etc/pki/tls/certs/localhost.crt`
 - `SSLCertificateKeyFile /etc/pki/tls/private/localhost.key`
 3. Restart httpd by running the following command:

```
service httpd restart
```

TLS is enabled on the default HTTPS port 443.

Important

If you're using a self-signed certificate and migrating from Web Services to Interaction Recording Web Services, you'll need to complete a few more steps. See [Re-importing the Certificate for WebDAV](#) for details.

Changing storage location

You can use any one of the following methods to expand the available storage or to migrate the recording files to a new WebDAV server.

Method 1

1. Leave the existing WebDAV server in place and point the Storage Destination in IVR Profile to the new

WebDAV server. Ensure that the recordings storage destination folder in new WebDAV server has write access.

2. Configure the new WebDAV storage path in RWS Storage settings for voice recordings. For details, see [Configure the Storage Credentials for Interaction Web Services](#).
3. If the tenant has Screen Recordings enabled, configure the new WebDAV storage path in Screen Recording storage settings and set the active property to true. For more details, see [Screen Recording Storage Settings](#).
4. Update the active property to false for the existing WebDAV storage path in Screen Recording storage settings.

Important

If the WebDAV servers are load balanced using the Load Balancer, add the new WebDAV servers to the load balancer as a separate balanced address and follow Steps 1 to 4. Retain the existing WebDAV load balancer configuration to ensure that old recordings are still accessible.

Method 2

1. Copy all the existing recording files to the new WebDAV server and make sure the file path is maintained as in the old WebDAV server.
2. Make the necessary changes to the new WebDAV server to take over the IP or FQDN of the old WebDAV server.

Next Step

- [Initialize Cassandra](#)

Initializing Cassandra

Make sure you have [installed and tested Cassandra](#) before completing the procedures below.

Creating the Cassandra Keyspace

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development—one Cassandra node (appropriate for a development or lab environment)
- Single Datacenter—one datacenter with a minimum of three Cassandra nodes
- Two Datacenters—two datacenters with a minimum of three Cassandra nodes in each datacenter

Important

For more complex Cassandra deployments, please consult with Genesys.

Select a tab below for the procedure that matches your deployment scenario.

Development

Creating the Cassandra Keyspace (1 Cassandra Node)

Start

1. Copy the **ks-schema-local.cql** file from **installation_CD/data** to the Cassandra node host.
2. Set the replication factor to 1 (the default), if needed. Since this is a single node deployment, you don't need to modify this value. Refer to the [Cassandra documentation](#) for more information about replication factors.

```
replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}
```

3. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f ks-schema-local.cql
```

...where *cassandra_host* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Single Datacenter

Creating the Cassandra Keyspace (1 Datacenter)

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **ks-schema-prod.cql** file from *installation_CD/data* to the Cassandra node host.
2. For fault tolerance, Genesys recommends that you use at least 3 Cassandra nodes and set the replication factor to 3. Refer to the [Cassandra documentation](#) for more information about replication factors. To modify this value, change the following line:

```
replication = {'class': 'SimpleStrategy', 'replication_factor': '<replication-factor-in-your-environment>'}
```

3. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f ks-schema-prod.cql
```

...where *cassandra_host* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Two Datacenters

Creating the Cassandra Keyspace (2 Datacenters)

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **ks-schema-prod_HA.cql** file from *installation_CD/data* to the Cassandra node host.
2. Modify the following content:

```
replication = {'class': 'NetworkTopologyStrategy', 'AZ1': '3', 'AZ2': '3'}
```

- a. Add the datacenter name. You can use nodetool to find the name of the datacenter by examining

the output of `nodetool status` (the tool is located in the `bin` directory of Cassandra). The following is sample output from the `nodetool`:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns (effective) \
UN 192.0.2.10    4.58 MB    256        100.0%           \
UN 192.0.2.11    2.3 MB     256        100.0%           \
UN 192.0.2.12    4.11 MB    256        100.0%           \
                               Host ID      Rack
                               dab220f6-7744-4709-b2ce-d18629076a76 rack1
                               922a3442-63f9-43f7-af08-2cd62f02e28b rack1
                               913f77c3-7dc2-4d93-b643-9e0c514314d1 rack1
Datacenter: datacenter2
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns (effective) \
UN 198.51.100.10 4.16 MB    256        100.0%           \
UN 198.51.100.11 2.24 MB    256        100.0%           \
UN 198.51.100.12 4.19 MB    256        100.0%           \
                               Host ID      Rack
                               cd92c658-176a-453b-b118-9b952f78f237 rack1
                               c4afb92-59c8-450f-b9b1-79b3454c04a2 rack1
                               d6fd07b4-8f6c-487e-a574-43d6f5980ac8 rack1
```

- b. Add the replication factor. Refer to the [Cassandra documentation](#) for more information about replication factors.

Based on the `nodetool` output above, your line might be:

```
replication = {'class': 'NetworkTopologyStrategy', 'datacenter1': '3', 'datacenter2': '3'}
```

3. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f ks-schema-prod_HA.cql
```

...where *cassandra_host* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Creating the Column Families

Complete the following procedure on one node in your Cassandra cluster.

Cassandra 4

Start

1. Copy the **cf-schema-c4-8.5.500.30.cql** file from **installation_CD/data** to the Cassandra node host.
2. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f cf-schema-c4-8.5.500.30.cql  
...where cassandra_host is the host name (fully qualified domain name) or IP address of the  
Cassandra node
```

End

Cassandra 2

Start

1. Copy the **cf-schema.cql** file from **installation_CD/data** to the Cassandra node host.
2. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f cf-schema.cql  
...where cassandra_host is the host name (fully qualified domain name) or IP address of the  
Cassandra node
```

End

Next Step

- [Elasticsearch](#)

Elasticsearch

Elasticsearch 1.x (deprecated)

Interaction Recording Web Services uses [Elasticsearch](#) — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that is separate from your Interaction Recording Web Services nodes. See [Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster](#) for details. It's possible to set up a co-located Elasticsearch cluster, which means that Elasticsearch is included in your Interaction Recording Web Services nodes. See [Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster](#) for details.

Important

- If you are using GIR with Workspace Web Edition, a shared deployment of Elasticsearch should be used. Ensure you also review the Web Services and Applications documentation for Elasticsearch. For details see: [Elasticsearch](#).
- The Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 1.x version of Elasticsearch](#).

Prerequisites

Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 1.x version of Elasticsearch. Refer to the [Elasticsearch documentation](#) for details. Note that the Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Complete the following steps for each Elasticsearch node

1. Copy the **elasticsearch.yml.sample** file from the **installation_CD/config-templates/** folder, to the Elasticsearch configuration folder on a standalone machine, and rename it to **elasticsearch.yml**. If you use **.rpm** for Elasticsearch, use **/etc/elasticsearch/** as the configuration folder. If you use the **gzipped tarball**, use **\$installDir/config**.
2. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
index.analysis.analyzer.whitespace_lowercase.tokenizer: whitespace
index.analysis.analyzer.whitespace_lowercase.filter: lowercase
index.analysis.analyzer.mediaPartitionAnalyzer.tokenizer: path_hierarchy
threadpool.index.queue_size: -1
threadpool.bulk.queue_size: -1
bootstrap.mlockall: true
indices.fielddata.cache.size: 75%
indices.breaker.fielddata.limit: 80%
path.conf: <Elasticsearch configuration path>
path.data: <Elasticsearch installation path>/esdata
node.name: ToBeChanged: <name of the Elasticsearch node. Set uniquely for each node>
cluster.name: ToBeChanged: <name of the Elasticsearch cluster>
transport.tcp.port: 9300
http.port: 9200
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ToBeChanged: <comma separated list of Elasticsearch nodes>
discovery.zen.minimum_master_nodes: ToBeChanged: <set to the minimum number of master nodes>
gateway.recover_after_nodes: ToBeChanged: <calculate based on the number of Elasticsearch nodes with rule: '<NUMBER_ES_NODES> / 2 + 1'>
gateway.recover_after_time: 1m
gateway.expected_nodes: ToBeChanged: <set to the number of Elasticsearch nodes>
```

3. Copy the **installation_CD/elasticsearch/templates** folder, along with its **.json** file contents, to a new templates folder under the configuration folder of Elasticsearch (for example, **/etc/elasticsearch/templates** if you use **.rpm** for Elasticsearch, or **\$installDir/config/templates** if you use the **gzipped tarball**) on each node.
4. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/current/file-descriptors.html>.

Important

The Elasticsearch engine requires a large Metaspace space. To increase the Metaspace space, pass the following argument to the JVM used to run Elasticsearch:

```
"-XX:MaxMetaspaceSize=512m"
```

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a co-located Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 1.x version of Elasticsearch](#).

Complete the following steps for each Interaction Recording Web Services node that you want to host Elasticsearch

1. Install Elasticsearch using the latest stable 1.x version of Elasticsearch.
2. Copy the **elasticsearch.yml.sample** file from the **installation_CD/config-templates/** folder, to the Elasticsearch configuration folder on the Interaction Recording Web Services node, and rename it to **elasticsearch.yml**. If you use **.rpm** for Elasticsearch, use **/etc/elasticsearch/** as the configuration folder. If you use the **gzipped tarball**, use **\$installDir/config**.
3. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file: **Note:** **<Elasticsearch installation path>** refers to the location on which Elasticsearch has been installed.

```
index.analysis.analyzer.whitespace_lowercase.tokenizer: whitespace
index.analysis.analyzer.whitespace_lowercase.filter: lowercase
index.analysis.analyzer.mediaPartitionAnalyzer.tokenizer: path_hierarchy
threadpool.index.queue_size: -1
threadpool.bulk.queue_size: -1
bootstrap.mlockall: true
indices.fielddata.cache.size: 75%
indices.breaker.fielddata.limit: 80%
path.conf: <Elasticsearch configuration path>
path.data: <Elasticsearch installation path>/esdata
node.name: ToBeChanged: <name of the Elasticsearch node. Set uniquely for each node>
cluster.name: ToBeChanged: <name of the Elasticsearch cluster>
transport.tcp.port: 9300
http.port: 9200
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ToBeChanged: <comma separated list of Elasticsearch
```

```
nodes>
discovery.zen.minimum_master_nodes: ToBeChanged: <set to the minimum number of master
nodes>
gateway.recover_after_nodes: ToBeChanged: <calculate based on the number of
Elasticsearch nodes with rule: '<NUMBER_ES_NODES> / 2 + 1'>
gateway.recover_after_time: 1m
gateway.expected_nodes: ToBeChanged: <set to the number of Elasticsearch nodes>
```

Important

Do not forget to update **<Elasticsearch installation path>** to the appropriate value.

4. Copy the **installation_CD/elasticsearch/templates** folder, along with its **.json** file contents, to a new templates folder under the configuration folder of Elasticsearch (for example, **/etc/elasticsearch/templates** if you use **.rpm** for Elasticsearch, or **\$installDir/config/templates** if you use the **gzipped tarball**) on each node.
5. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/current/file-descriptors.html>.
6. Set the **crClusterName** option in the **application.yaml** file to the name of the cluster, as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
7. Set the **elasticSearchSettings** option in the **application.yaml** file to the appropriate values for your environment.

Important

The Elasticsearch engine requires a large Metaspace space. To increase the Metaspace space, pass the following argument to the JVM used to run Elasticsearch: **"-XX:MaxMetaspaceSize=512m"**

Migrating an Existing Elasticsearch Deployment to Schema V2

Important

The following procedure should only be performed once per contact center (that is, for each contact center tenant in a multi-tenant deployment). This procedure should not be performed for a new GIR installation.

Perform the following steps while your system is running, without service interruption.

1. Copy the **call_recordingv2_template.json** and **screen_recordingv2_template.json** files from the **installation_CD/elasticsearch/templates/** folder to the **templates** folder in each node in your Elasticsearch cluster.

2. Perform a rolling restart of each node in your Elasticsearch cluster. Stop and restart each node and wait until it is restarted and is operational before stopping and restarting the next node.
3. Prepare a new dedicated **Interaction Recording Web Services** node as follows:
 - a. Install Interaction Recording Web Services in the same way a regular **Interaction Recording Web Services** node is installed. Do not add this node to the Interaction Recording Web Services Load Balancer.
 - b. Edit the Interaction Recording Web Services **application.yaml** file, by adding the following configuration. Verify that you add lines under nodes for all the existing Interaction Recording Web Services nodes in your deployment:

```
elasticSearchSettings:
  useTransportClient: true
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: true
  ignoreClusterName: true
  pingTimeout: 10000
  nodesSamplerInterval: 10000
  enableIndexVerificationAtStartup: false
  indexPerContactCenter: true
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

3. Increase the **Hystrix** timeout for **RecordingOperationApiTaskV2** on the new Interaction Recording Web Services node by adding the following line to the Hystrix configuration:

```
hystrix.command.RecordingOperationApiTaskV2.execution.isolation.thread.timeoutInMilliseconds=<max time acceptable in milliseconds>
```

Important

<max time acceptable in milliseconds> should exceed the time that the re-indexing operation is expected to take. This value varies depending on how you elect to divide the re-indexing iterations. If you expect each re-indexing operation to take approximately one hour, then set this parameter to a value such as 7200000.

4. Determine the Contact Center ID using the following command:

```
curl -u <ops-user>:<ops-pass> http://< selected-web-services-node>: <web-services-listening-port>/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://< selected-web-services-node>: <web-services-listening-port>/api/v2/ops/contact-centers/<contact-center-id>"]}
```

Migrate Call Recording Index

1. Start the migration process for call recording by issuing the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json" "http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/features" -d '{ "uris":["schema-elasticsearch-migration-to-v2-call-recording"] }'
```

2. Immediately after performing step #1, note the current time and initialize the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json" "http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/recordings" -d '{ "operationName":"forceIndex", "from": <start-range-in-milliseconds>, "to": <stop-range-in-milliseconds>, "purgeOld":<value> }'
```

Genesys recommends to perform re-indexing in multiple iterations depending on how many records exist in Cassandra. The key aspect when determining "**from**" and "**to**" values is to use these parameters to specify the number of records to be re-indexed at a time. A reasonable estimate for the time taken to re-index can be 5,000,000 records in one hour, although this is dependent on your Cassandra and Elasticsearch deployment. Therefore, depending on the number of records in your deployment, this could be accomplished with a single iteration, where the "**from**" and "**to**" values specified cover the entire time range of content within Cassandra.

Important

purgeOld is set to **true** initially, and to **false** for all subsequent invocations.

3. Repeat the command from step #2 varying the “from” and “to” values to completely cover all recordings that exist (up to and including the time noted at the beginning of step 2), so that they are included in the new index.
For each iteration, ensure that **purgeOld** is set to **false** so that the newly created index is not removed.
4. Once the **forceIndex** commands are completed (so that the entire set of recordings have been re-indexed), configure Interaction Recording Web Services to use the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-v2-call-recording"]
}'
```

5. Verify that the Search functionality is working properly using the **GetRecordings API**. For additional information refer to [Genesys Interaction Recording API](#).

Important

SpeechMiner cannot be used to perform this validation since it uses a different mechanism to search for call recordings.

Once this procedure is completed both the old index and the new index are maintained and the new index is used for all searches.

Migrate Screen Recording Index

1. Start the migration process for screen recording by issuing the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-screen-recording"]
}'
```

2. Immediately after performing step #1, note the current time and initialize the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/screen-recordings" -d '{
  "operationName":"forceIndex",
  "from": <start-range-in-milliseconds>,'
```

```
"to": <stop-range-in-milliseconds>,  
"purgeOld":<value>  
'}
```

Genesys recommends to perform re-indexing in multiple iterations depending on how many records exist in Cassandra. The key aspect when determining "**from**" and "**to**" values is to use these parameters to specify the number of records to be re-indexed at a time. A reasonable estimate for the time taken to re-index can be 5,000,000 records in one hour, although this is dependent on your Cassandra and Elasticsearch deployment. Therefore, depending on the number of records in your deployment, this could be accomplished with a single iteration, where the "**from**" and "**to**" values specified cover the entire time range of content within Cassandra.

Important

purgeOld is set to **true** initially, and to **false** for all subsequent invocations.

3. Repeat the command from step #2 varying the "from" and "to" values to completely cover all recordings that exist (up to and including the time noted at the beginning of step 2), so that they are included in the new index.
For each iteration, ensure that **purgeOld** is set to **false** so that the newly created index is not removed.
4. Once the **forceIndex** commands are completed (so that the entire set of recordings have been re-indexed), configure Interaction Recording Web Services to use the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"  
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-  
center-id>/features" -d '{  
  "uris":["schema-elasticsearch-v2-screen-recording"]  
'}
```

5. Verify that the Search functionality is working properly against the full range of screen recordings, by using the SpeechMiner Screen Recording grid.

Once this procedure is completed both the old index and the new index are maintained and the new index is used for all searches.

Completing the Migration

Once you have migrated both the Call Recording and Screen Recording indexes, both the old index and the new index are updated for every new recording. This process consumes additional disk space. To avoid the use of additional disk space, perform the following steps to remove the old indexes once testing has confirmed that the new indexes are fully operational:

Important

Once the following steps are performed, it will not be possible to roll back the migration.

1. Turn off the schema migration feature flag for the index being migrated, by using the following command:

For **Call Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-call-recording"]
}'
```

For **Screen Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-screen-recording"]
}'
```

2. Delete the old indexes by using the following command:

```
curl -XDELETE http://<es-node>:9200/<index-name>
```

Where:

- **<es-node>** is one of the Elasticsearch nodes in the cluster.
- **<index-name>** is the index name for the original schema:
 - **{contact-center-id}** for call recording when an embedded Elasticsearch cluster is used, or **call-recording-{contact-center-id}** for a standalone Elasticsearch cluster deployment. For example, **f3eec6cb-f624-4ac2-975e-6a60e0ebf878** or **call-recording-f3eec6cb-f624-4ac2-975e-6a60e0ebf878**.
 - **screen-recording-{contact-center-id}** for screen recording. For example, **screen-recording-f3eec6cb-f624-4ac2-975e-6a60e0ebf878**.

If you are unsure, the index names in use on Elasticsearch can be determined by using the following command (where **<es-node>** is one of the Elasticsearch nodes in the cluster):

```
curl -XGET http://<es-node>:9200/_cat/indices?v
```

Important

At this point the new Interaction Recording Services node that was used for these migration steps is no longer required and can be shut down or re-purposed.

Rolling Back the Migration

In the event of a problem with the index migration, perform the following steps to implement the old (previous) index and remove the new index.

1. If the new index was enabled as the default index, run the following command to use the old index:

For **Call Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-v2-call-recording"]
}'
```

For **Screen Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-v2-screen-recording"]
}'
```

2. Stop updates to the new index by turning off the schema migration feature flag, using the following command:

For **Call Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-call-recording"]
}'
```

For **Screen Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
```

```
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-screen-recording"]}
}'
```

3. To delete the new index, run the following command:

```
curl -XDELETE http://<es-node>:9200/<index-name>
```

Where:

- **<es-node>** is one of the Elasticsearch nodes in the cluster.
- **<index-name>** is the index name for the new schema:
 - **call-recording-v2-{region}-{contact-center-id}** for call recordings.
 - **screen-recording-v2-{region}-{contact-center-id}** for screen recordings.
 - **region** is the value of the **crRegion** parameter specified in the **application.yaml** file in the Interaction Recording Web Services node that was used to perform the index migration process for both the call recording index and the screen recording index.

Elasticsearch 7.16.3

Interaction Recording Web Services uses **Elasticsearch** — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster](#). You can also set up a co-located Elasticsearch cluster, which means that Elasticsearch is included in your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster](#).

Important

- If you are using GIR with Workspace Web Edition, refrain from using a shared deployment of Elasticsearch. This is because Web Services and Applications support Elasticsearch 1.x only and do not support ES 7.16.3. For details see: [Elasticsearch](#). This is applicable if you are installing Web Services and Applications version 8.5.201.09 or earlier.
- The Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 7.16.3 version of Elasticsearch](#).

Prerequisites

- Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 7.16.3 version of Elasticsearch. Refer to the [Elasticsearch documentation](#) for details. Note that the Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.
- Interaction Recording Web Services deployment version should be 8.5.204.16 or higher.
- Interaction Recording Web Services supported Elasticsearch 7.16.3 installed on RedHat 8/9 and Java 11.

Limitations of Elasticsearch 7.16.3

- Elasticsearch 7.16.3 only supports schema V3.
- Genesys is not responsible for migration of existing data to the latest version of Elasticsearch on premise environments.
- Interaction Recording Web Services does not support scan and scroll functionality on Elasticsearch 7.16.3.

Complete the following steps for each Elasticsearch node

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 7.16.3.
2. Open the following **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
```

```
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.16/file-descriptors.html>.

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a co-located Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone as co-located is discontinued after RWS version 8.5.205.69. For additional information, refer to the **latest stable 7.16.3 version of Elasticsearch**.

Complete the following steps for each Interaction Recording Web Services node that you want to host Elasticsearch

1. Refer to the **Elasticsearch documentation** for installing and configuring Elasticsearch 7.16.3.
2. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
```

```
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.16/file-descriptors.html>.
4. Set the **crClusterName** option in the **application.yaml** file to the name of the cluster, as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
5. Set the **elasticSearchSettings** option in the **application.yaml** file to the appropriate values for your environment.

Perform the following steps in each Elasticsearch node while your system is running, without service interruption

1. Create call recording schema V3:
 - a. Copy the **call_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
 - b. Create call-recording schema V3 using the following command:


```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/call-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@call_recording_v3_template.json
```
3. Create screen recording schema V3:
 - a. Copy the **screen_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
 - b. Create screen-recording V3 schema using the following command:


```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/screen-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@screen_recording_v3_template.json
```

Configure Interaction Recording Web Services using Elasticsearch 7.16.3

1. Update the **application.yaml** file on each Interaction Recording Web Services node.
2. update **useTransportClient** to be "false".
3. Add a new property **useRestClient** as follows:

```
elasticSearchSettings:
  useTransportClient: false
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: false
  ignoreClusterName: false
  pingTimeout: 5000
  nodesSamplerInterval: 5000
  useRestClient: true
```



```
restClient:
  nodes:
    - {host: <elastic-search-node1>, port: 9200}
    - {host: <elastic-search-node2>, port: 9200}
    - {host: <elastic-search-node3>, port: 9200}
waitToIndexTimeout: 5000
scanReadTimeoutSeconds: 60
scrollTimeoutSeconds: 240
countReadTimeoutSeconds: 60
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

Elasticsearch 7.17.15

Interaction Recording Web Services uses **Elasticsearch** — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster](#). You can also set up a co-located Elasticsearch cluster, which means that Elasticsearch is included in your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster](#).

Important

- If you are using GIR with Workspace Web Edition, refrain from using a shared deployment of Elasticsearch. This is because Web Services and Applications support Elasticsearch 1.x only and do not support ES 7.17.15. For details see: [Elasticsearch](#). This is applicable if you are installing Web Services and Applications version 8.5.201.09 or earlier.
- The Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster

by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 7.17.15 version of Elasticsearch](#).

Prerequisites

- Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 7.17.15 version of Elasticsearch. Refer to the [Elasticsearch documentation](#) for details. Note that the Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.
- Interaction Recording Web Services deployment version should be 8.5.205.32 or higher.
- Interaction Recording Web Services supported Elasticsearch 7.17.15 installed on RedHat 8/9 and Java 11.

Limitations of Elasticsearch 7.17.15

- Elasticsearch 7.17.15 only supports schema V3.
- Genesys is not responsible for migration of existing data to the latest version of Elasticsearch on premise environments.

Complete the following steps for each Elasticsearch node

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 7.17.15.
2. Open the following **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/file-descriptors.html>.

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a co-located Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone as co-located is discontinued after RWS version 8.5.205.69. For additional information, refer to the [latest stable 7.17.15 version of Elasticsearch](#).

Complete the following steps for each Interaction Recording Web Services node that you want to host Elasticsearch

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 7.17.15.
2. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/file-descriptors.html>.
4. Set the **crClusterName** option in the **application.yaml** file to the name of the cluster, as specified by **cluster.name** in the **elasticsearch.yml** configuration file.

- Set the **elasticSearchSettings** option in the **application.yaml** file to the appropriate values for your environment.

Perform the following steps in each Elasticsearch node while your system is running, without service interruption

- Create call recording schema V3:

- Copy the **call_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- Create call-recording schema V3 using the following command:

```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/call-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@call_recording_v3_template.json
```

- Create screen recording schema V3:

- Copy the **screen_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- Create screen-recording V3 schema using the following command:

```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/screen-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@screen_recording_v3_template.json
```

Configure Interaction Recording Web Services using Elasticsearch 7.17.15

- Update the **application.yaml** file on each Interaction Recording Web Services node.
- update **useTransportClient** to be "false".
- Add a new property **useRestClient** as follows:

```
elasticSearchSettings:
  useTransportClient: false
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: false
  ignoreClusterName: false
  pingTimeout: 5000
  nodesSamplerInterval: 5000
  useRestClient: true
  restClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9200}
      - {host: <elastic-search-node2>, port: 9200}
      - {host: <elastic-search-node3>, port: 9200}
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: 60
  scrollTimeoutSeconds: 240
  countReadTimeoutSeconds: 60
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

Elasticsearch 8.18.2

Interaction Recording Web Services uses **Elasticsearch** — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Interaction Recording Web Services nodes. For more details, see **Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster** section below.

Important

Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Workspace Web Edition deployment and SpeechMiner deployment.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the **latest stable 8.18.2 version of Elasticsearch**.

Prerequisites

- Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 8.18.2 version of Elasticsearch. Refer to the **Elasticsearch documentation** for details.

- Interaction Recording Web Services deployment version should be 8.5.500.24 or higher.
- Interaction Recording Web Services supported Elasticsearch 8.18.2 installed on RedHat 8/9 and Java 17.

Limitations of Elasticsearch 8.18.2

- Elasticsearch 8.18.2 only supports schema V3.
- Genesys is not responsible for migration of existing data to the latest version of Elasticsearch on premise environments.

Complete the following steps for each Elasticsearch node

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 8.18.2.
2. Open the following **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
# Security settings
xpack.security.enabled: [ToBeChangedOrRemoved: true|false]
xpack.security.enrollment.enabled: [ToBeChangedOrRemoved: true|false]
xpack.security.http.ssl:
  enabled: [ToBeChangedOrRemoved: true|false]
  keystore.path: [ToBeChangedOrRemoved: <KEYSTORE_PATH>]
xpack.security.transport.ssl:
  enabled: [ToBeChangedOrRemoved: true|false]
  verification_mode: [ToBeChangedOrRemoved: certificate]
  keystore.path: [ToBeChangedOrRemoved: <KEYSTORE_PATH>]
  truststore.path: [ToBeChangedOrRemoved: <TRUSTSTORE_PATH>]

indices.id_field_data.enabled: true
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/8.18/file-descriptors.html>.

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Perform the following steps in each Elasticsearch node while your system is running, without service interruption

1. Create call recording schema V3:

- a. Copy the **call_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- b. Create call-recording schema V3 using the following command:

```
$ curl -XPUT -u <USERNAME>:<PASSWORD> -H "Content-Type: application/json"
-d@<call_recording_v3_template.json> --cacert <PATH_TO_CA_CERT>
https://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP
port>/_template/call-recording-v3-template
```

3. Create screen recording schema V3:

- a. Copy the **screen_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- b. Create screen-recording V3 schema using the following command:

```
$ curl -XPUT -u <USERNAME>:<PASSWORD> -H "Content-Type: application/json"
-d@<screen_recording_v3_template.json> --cacert <PATH_TO_CA_CERT>
https://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP
port>/_template/screen-recording-v3-template
```

Configure Interaction Recording Web Services using Elasticsearch 8.18.2

1. Update the **application.yaml** file on each Interaction Recording Web Services node.
2. update **useTransportClient** to be "false".
3. Add a new property **useRestClient** as follows:

```
elasticSearchSettings:
  retriesOnConflict: 3
  useRestClient: true
  restClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9200}
      - {host: <elastic-search-node2>, port: 9200}
      - {host: <elastic-search-node3>, port: 9200}
    authentication:
      type: [ToBeChangedOrRemoved: none|basic]
      userName: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_USER_NAME>]
      password: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_PASSWORD>]
      useSSL: [ToBeChangedOrRemoved: true|false]
      truststore: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PATH>]
      truststorePassword: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PASSWORD>]
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: 60
  scrollTimeoutSeconds: 240
  countReadTimeoutSeconds: 60
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

Next Step

- [Install Interaction Recording Web Services.](#)

Installing

To install Interaction Recording Web Services (RWS), first you need to set up the following two application objects it uses in the Genesys configuration environment:

- Cluster Application
- Node Application

If RWS is being deployed along with Web Services (GWS), then GWS must be installed first, and the Cluster Application created during the GWS installation is shared between both components.

Interaction Recording Web Services

Interaction Recording Web Services (RWS)

Creating the Application Templates

Using Genesys Administrator Extension, complete the steps below to create application templates to use for your IRWS_Cluster and IRWS_Node applications.

Start

1. To create the Genesys Generic Server template, navigate to **Configuration > Environment > Application Templates**.
2. Select **New...** and configure the properties of the template as shown below:
 - Name: IRWS_Cluster_Template
 - Type: Genesys Generic Server
 - Version: 8.5
 - State: Enabled
3. Click **Save & Close**.
4. To create the Genesys Generic Client template, select **New...** again and configure the properties of the template as shown below:
 - Name: IRWS_Node_Template
 - Type: Genesys Generic Client

- Version: 8.5
- State: Enabled

5. Click **Save & Close**.

End

Creating the IRWS Cluster Application

Start

1. Navigate to **Configuration > Environment > Applications** and click **New...**
2. In the **General** section, configure the properties of the application as shown below:
 - Name: IRWS_Cluster
 - Template: IRWS_Cluster_Template (this is the template you made in [Creating the Application Templates](#))
 - State: Enabled
 - Working Directory: .
 - Command Line: .
 - Command Line Arguments: .

Important

You need to add a "." to the Working Directory, Command Line, and Command Line Arguments fields, as shown above. These values are mandatory for all applications and must be entered to save the application object. Interaction Recording Web Services does not use these values, so the "." is used as a placeholder.

3. Choose a Host object. See [Create Host](#) in the *Management Framework Deployment Guide* for more information about Host objects.
4. Add the following connections:
 - Configuration Server (you can add CS Proxy using the `[csproxy].proxy-writable=true` option.)
 - **Interaction Server** (if supporting multimedia)
 - T-Server/SIP Server (when supporting voice)

Important

When working with dual data centers, RWS requires a connection to each Interaction Server Application Cluster (in both data centers), in order to provide support for disaster recovery with an Interaction Server between the data centers. Use the following application parameters to provide the connection between RWS and each Interaction Server Application Cluster:

- `siteName=DC1;clusterType=eservices #For DC1 Interaction Server`

Application cluster

- `siteName=DC2;clusterType=eservices` #For DC2 Interaction Server Application cluster

5. In the **Tenants** section, select a Tenant:

1. Click **Add**.
2. Choose the Environment tenant (or any other tenant that has a connection to your Configuration Server).
3. Click **OK**.

Important

This step is for adding a single tenant only. For information about multi-tenant deployments, see [Deploying Genesys Interaction Recording in a Multi-Tenant Deployment](#).

6. Add a default Listening Port:

1. Click **Add**.
2. Enter the application's Port. For instance 7000.
3. Click **OK**.

End

Creating the RWS Node Application

Start

1. Navigate to **Configuration > Environment > Applications** and click **New....**
2. In the **General** section, configure the properties of the application as shown below:
 - Name: IRWS_Node
 - Template: IRWS_Node_Template (this is the template you made in [Creating the Application Templates](#))
 - State: Enabled
3. Add the following connections:
 - Cluster application that was configured in the previous procedure.
4. Click **Save & Close**.

End

Interaction Recording Web Services with Web Services

Interaction Recording Web Services (RWS) with GWS

Creating the Application Template

Using Genesys Administrator Extension, complete the steps below to create an application template to use for your IRWS_Node applications.

Start

1. To create the Genesys Generic Client template, navigate to **Configuration > Environment > Application Templates**.
2. Select **New...** and configure the properties of the template as shown below:
 - Name: IRWS_Node_Template
 - Type: Genesys Generic Client
 - Version: 8.5
 - State: Enabled
3. Click **Save & Close**.

End

Creating the RWS Node Application

Start

1. Navigate to **Configuration > Environment > Applications** and click **New....**
2. In the **General** section, configure the properties of the application as shown below:
 - Name: IRWS_Node
 - Template: IRWS_Node_Template (this is the template you made in [Creating the Application Template](#))
 - State: Enabled
3. Add the following connections:
 - Cluster application that was configured as part of the GWS installation. For additional information, refer to the [Creating the Web Services \(WS\) Cluster Application](#) section.
4. Click **Save & Close**.

End

Next Step

- [Deploy the web application](#)

Deploying the Web Application

The final deployment step is to install Interaction Recording Web Services as a service. Complete the following steps for each Interaction Recording Web Services node.

Deploy Using Red Hat Enterprise Linux 8/9

Start

1. Create a new folder on your Interaction Recording Web Services node. For example, **ir-web-services**. This is the home folder for the web application.
2. Copy the **gir.jar** file from the installation CD to your new Interaction Recording Web Services home folder.
3. Create a new folder **/usr/lib/systemd/system/gir.service.d**
4. Copy the following files to the specified folders on your Interaction Recording Web services host:
For Red Hat Enterprise Linux 8/9
 - **installation_CD/rhel/usr/lib/systemd/system/gir.service.d/gir.conf** to the folder **/usr/lib/systemd/system/gir.service.d**
 - **installation_CD/rhel/usr/lib/systemd/system/gir.service** to the folder **/usr/lib/systemd/system**
5. Create a new folder **/usr/libexec/initscripts/legacy-actions/gir**
6. Copy the following files to the specified folders on your host:
For Red Hat Enterprise Linux 8/9
 - **installation_CD/rhel/usr/bin/gir** to the folder **/usr/bin**
 - **installation_CD/rhel/usr/libexec/initscripts/legacy-actions/gir/config** to the folder **/usr/libexec/initscripts/legacy-actions/gir**
 - **installation_CD/rhel/usr/libexec/initscripts/legacy-actions/gir/version** to the folder **/usr/libexec/initscripts/legacy-actions/gir**
7. Open **/usr/lib/systemd/system/gir.service.d/gir.conf** on your host and update the following environment variables to values appropriate for your Interaction Recording Web Services node:
 - **WorkingDirectory**—The Interaction Recording Web Services home folder you created in Step 1.
 - **Environment=GIR_TEMP**—The location where you want Interaction Recording Web Services to store temp files.
 - **Environment=GIR_CONF**—The location where you want to store the Interaction Recording Web Services configuration files. If you do not specify a value for **GIR_CONF**, Interaction Recording Web Services uses **WorkingDirectory/config**.

Important

If you are installing RWS 8.5.205.10, ensure that the environment variable `Environment=GIR_CONF` is either specified as **RWS home folder/config** or not specified.

8. Create the `GIR_CONF` folder you specified in Step 7. If you didn't set `GIR_CONF`, create a folder called **config** in **WorkingDirectory**— for example, **ir-web-services/config**.
9. Create the following configuration files in the folder you created in Step 8. You can simply copy the files from **installation_CD/config-templates** and remove the **.sample** extension. You'll learn more about the settings in these files as you go through the configuration steps for Interaction Recording Web Services and its features later in this guide.
 - **application.yaml**
 - **hystrix.properties**
 - **logback.xml**

Important

Ensure that only a single copy of the **application.yaml** file is deployed across all the GIR file locations described above.

10. Create the user group **gir**.
11. Create the **gir** user in the **gir** user group and provide the user with ownership, and read and write permissions for the following folders:
 - The folders defined in the `WorkingDirectory`, `GIR_TEMP`, and `GIR_CONF` environment variables.
 - The folder defined in the path configuration item within the logging section in the **application.yaml** file (**/var/log/jetty9** by default).
12. Set executable permissions on:
 - **/usr/bin/gir**
 - **/usr/libexec/initscripts/legacy-actions/gir/config**
 - **/usr/libexec/initscripts/legacy-actions/gir/version**
13. Use the following commands to register the new service on your host:

```
systemctl daemon-reload
systemctl enable gir.service
```

End

Next Step

- [Configuring Interaction Recording Web Services](#)

Configuring Interaction Recording Web Services

You'll need to update the **application.yaml** file on each of your **Interaction Recording Web Services** nodes to provide the basic configuration. You created this file (or Interaction Recording Web Services created it for you) as part of [Deploying the Web Application](#). In later topics, you'll learn more about modifying this file to configure additional [features](#) and [security](#). For now, review the contents below for details about each section in the **application.yaml** configuration file.

Important

When editing the **application.yaml** file, the values for the configuration options that are strings must be enclosed in double quotation marks in certain cases. Specifically:

- For string options only, the values YES, NO, ON, OFF, TRUE, FALSE (in upper or lower case) must be quoted.
- If the option is a boolean (true/false) option, then any of the values in the previous bullet can be used without quotes.
- Values that look like numbers but are treated as strings (for example; PINs, phone numbers, encryption keys), that begin with leading zeroes must be quoted.
- Avoid placing leading zeroes on numeric options; doing so will cause your option to be interpreted as an octal value.

For example, specifying `crRegion: NO` (indicating Norway) will be interpreted as `crRegion: FALSE`. Instead, this must be specified using double quotation marks `crRegion: "NO"`.

Logging Settings

The purpose of this section is to tell Interaction Recording Web Services where to find the **logback.xml** file you created (or Interaction Recording Web Services created for you) as part of [Deploying the Web Application](#) and where to save logs.

The **application.yaml.sample** file includes the following default logging section:

```
logging:
  config: logback.xml
  file: cloud.log
  path: /var/log/jetty9
```

See [logging](#) for details about all supported configuration settings for this section.

Jetty Settings

Since Jetty is embedded in Interaction Recording Web Services, you have to use the jetty section of the **application.yaml** file to tell Interaction Recording Web Services how Jetty should behave.

The **application.yaml.sample** file includes the following default jetty section:

```
jetty:
  host: [RWS_HOST]
  port: 8080
  idleTimeout: 30000
  soLingerTime: -1
  sessionMaxInactiveInterval: 1800
  enableWorkerName: true
  enableRequestLog: true
  requestLog:
    filename: yyyy_mm_dd.request.log
    filenameDateFormat: yyyy_MM_dd
    logTimeZone: GMT
    retainDays: 90
    append: true
    extended: true
    logCookies: true
    logLatency: true
    preferProxiedForAddress: true
  enableSsl: false
  ssl:
    port: 443
    securePort: 8443
    keyStorePath: [KEYSTORE_PATH]
    keyStorePassword: [KEYSTORE_PASSWORD]
    keyManagerPassword: [KEY_MANAGER_PASSWORD]
    trustStorePath: [TRUSTSTORE_PATH]
    trustStorePassword: [TRUSTSTORE_PASSWORD]
  httpOnly: true
  secure: false
  sessionCookieName: GIRJSESSIONID
```

See [jetty](#) for details about all supported configuration settings for this section.

Cassandra Cluster Settings

The settings in the **cassandraCluster** section tell Interaction Recording Web Services how your Cassandra cluster should be managed and accessed.

The **application.yaml.sample** file includes the following default **cassandraCluster** section:

```
cassandraCluster:
  thrift_port: 9160
  jmx_port: 7199
  keyspace: sipfs
  nodes: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>]
  backup_nodes: [ToBeChangedOrRemoved: <CASSANDRA_BACKUP_DC_NODES>]
  replication_factor: [ToBeChanged: <REPLICATION_FACTOR>]
  write_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
```

```

    read_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
    max_conns_per_host: 16
    max_cons: 48
    max_pending_conns_per_host: 80
    max_blocked_threads_per_host: 160

    cassandraVersion: 1.2
    useSSL: [ToBeChanged: "false" | "true"]
    truststore: [ToBeChanged: path to client truststore]
    truststorePassword: [ToBeChanged: truststore password]
    userName: [ToBeChangedOrRemoved: <CASSANDRA_USER_NAME>]
    password: [ToBeChangedOrRemoved: <CASSANDRA_USER_PASSWORD>]

```

The **application.yaml.sample** file includes the following default **cassandraCluster** section for Cassandra 4:

```

cassandraCluster:
  native_transport_port: 9042
  jmx_port: 7199
  keypace: sipfs
  nodes: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>]
  dataCenterName: [ToBeChanged: name of primary datacenter]
  backup_nodes: [ToBeChangedOrRemoved: <CASSANDRA_BACKUP_DC_NODES>]
  backupDataCenterName: [ToBeChangedOrRemoved: name of backup datacenter]
  switchOverUnHealthyNodes: [ToBeChangedOrRemoved: number of nodes down to prompt failover]
  replication_factor: [ToBeChanged: <REPLICATION_FACTOR>]
  write_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
  read_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
  max_conns_per_host: 16
  max_cons: 48
  max_pending_conns_per_host: 80
  max_blocked_threads_per_host: 160

  cassandraVersion: 4.1.4
  useSSL: [ToBeChanged: "false" | "true"]
  truststore: [ToBeChanged: path to client truststore]
  truststorePassword: [ToBeChanged: truststore password]
  userName: [ToBeChangedOrRemoved: <CASSANDRA_USER_NAME>]
  password: [ToBeChangedOrRemoved: <CASSANDRA_USER_PASSWORD>]

```

Make sure you update all settings marked as [ToBeChanged]. See [cassandraCluster](#) for details about all supported configuration settings for this section.

Server Settings

The settings in the **serverSettings** section provide the core settings Interaction Recording Web Services needs to run your node.

The **application.yaml.sample** file includes the following default **serverSettings** section:

```
serverSettings:
  # URLs
  externalApiUrlV2: [ToBeChanged: public URL including protocol, address and port, <PUBLIC_SCHEMA_BASE_URL>]/api/v2
  internalApiUrlV2: [ToBeChanged: internal URL including protocol, address and port, <INTERNAL_SCHEMA_BASE_URL>]/internal-api
  undocumentedExternalApiUrl: [ToBeChanged: public URL including protocol, address and port, <PUBLIC_SCHEMA_BASE_URL>]/internal-api

  # Paths
  pathPrefix: [ToBeChangedOrRemoved: <PATH_PREFIX>]
  internalPathPrefix: [ToBeChangedOrRemoved: <INTERNAL_PATH_PREFIX>]

  # General
  temporaryAuthenticationTokenTTL: [ToBeChangedOrRemoved: <TEMPORARY_AUTHENTICATION_TOKEN_TTL>]
  enableCsrfProtection: false

  # Timeouts
  activationTimeout: 12000
  configServerActivationTimeout: 35000
  configServerConnectionTimeout: 15000
  connectionTimeout: 4000
  inactiveUserTimeout: 60
  reconnectAttempts: 1
  reconnectTimeout: 10000

  # OPS account
  opsUserName: [ToBeChanged: <OPS_USER_NAME>]
  opsUserPassword: [ToBeChanged: <OPS_USER_PASSWORD>]

  # CME credentials
  applicationName: [ToBeChanged: <CONFIG_SERVER_RWS_APPLICATION_NAME>]
  applicationType: CFGGenericClient
  cmeUserName: [ToBeChanged: <CONFIG_SERVER_USER_NAME>]
  cmePassword: [ToBeChanged: <CONFIG_SERVER_USER_PASSWORD>]
  syncNode: [ToBeChanged: "true"|"false"]

  # ConfigServer String Encoding
  configServerDefaultEncoding: windows-1252

  # Call Recording
  createCallRecordingCF: true
  crClusterName: [ToBeChanged: <NAME_OF_ES_CLUSTER>]
  crRegion: [ToBeChanged: <CR_REGION>]
  cryptoSecurityKey: [ToBeChanged: <CRYPTO_SECURITY_KEY>]
```

```
webDAVMaxConnection: 50
webDAVMaxTotalConnection: 500

# Multi regional supporting
nodePath: [ToBeChanged: node position in cluster, example: /<REGION>/HOST
nodeId: [ToBeChangedOrRemoved: unique value in cluster <NODE_ID>]

# SSL and CA
caCertificate: [ToBeChangedOrRemoved: <PATH_TO_CA_FILE>]
jksPassword: [ToBeChangedOrRemoved: <JKS_PASSWORD>]
webDAVTrustedCA: [ToBeChangedOrRemoved: "true" | "false" | <PATH_TO_CA_FILE>]
webDAVJksPassword: [ToBeChangedOrRemoved: <WEBDAV_JKS_PASSWORD>]
rcsTrustedCA: [ToBeChangedOrRemoved: "true" | "false" | <PATH_TO_CA_FILE>]
rcsJksPassword: [ToBeChangedOrRemoved: <RCS_JKS_PASSWORD>]
speechMinerTrustedCA: [ToBeChangedOrRemoved: "true" | "false" | <PATH_TO_CA_FILE>]
speechMinerJksPassword: [ToBeChangedOrRemoved: <SMIR_JKS_PASSWORD>]

# CORS
crossOriginSettings:
  allowedOrigins: [ToBeChangedOrRemoved: <CROSS_ALLOWED_ORIGINS>]
  allowedMethods: [ToBeChangedOrRemoved: <CROSS_ALLOWED_METHODS>]
  allowedHeaders: [ToBeChangedOrRemoved: <CROSS_ALLOWED_HEADERS>]
  allowCredentials: [ToBeChangedOrRemoved: <CROSS_ALLOW_CREDENTIALS>]
  corsFilterCacheTimeToLive: 120
  exposedHeaders: [ToBeChangedOrRemoved: <CROSS_EXPOSED_HEADERS>]

# Elasticsearch
elasticSearchSettings:
  retriesOnConflict: 3
  useTransportClient: true
  transportClient:
    nodes:
      - {host: [ToBeChanged: <ELASTIC_SEARCH_NODE1>], port: [ToBeChanged: <ELASTIC_SEARCH_PORT>]}
      - {host: [ToBeChanged: <ELASTIC_SEARCH_NODE2>], port: [ToBeChanged: <ELASTIC_SEARCH_PORT>]}
    useSniff: false
    ignoreClusterName: false
    pingTimeout: 5000
    nodesSamplerInterval: 5000
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCAN_READ_TIMEOUT_SECONDS>]
  countReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_COUNT_READ_TIMEOUT_SECONDS>]
  scrollTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCROLL_TIMEOUT_SECONDS>]
```

```
# Recording Settings
recordingSettings:
  auditLogDeletedFiles: [ToBeChangedOrRemoved: "true"|"false"]
  recordCryptoServerDecryptMaxConnection: 50
  recordCryptoServerDecryptMaxTotalConnection: 500
  recordCryptoServerDecryptSocketTimeout: 30000
  keySpaceNameSettingsCacheSecondsTTL: 300
  regionsSettingsCacheSecondsTTL: 300
  readOnlyRetryAfterSeconds: 1200

# Screen Recording
screenRecordingSettings:
  enableSameSiteCookieForScreenRecordingPlayback: [ToBeChangedOrRemoved: "true"|"false"]
  screenRecordingVoiceEnabled: [ToBeChangedOrRemoved: "true"|"false"]
  screenRecordingEServicesEnabled: [ToBeChangedOrRemoved: "true"|"false"]
  recordingInteractionEventsTTL: 172800
  clientSessionManagerCacheTTL: 60
  contactCenterInfoManagerCacheTTL: 90

# Caching Settings
cachingSettings:
  enableSystemWideCaching: [ToBeChangedOrRemoved: "true"|"false"]
  contactCenterFeaturesTTL: 30
  contactCenterSettingsTTL: 30

# Screen Recording Connections Reporting
screenRecordingConnectionReportingSettings:
  reportingEnabled: [ToBeChangedOrRemoved: "true"|"false"]
  createReportingCF: [ToBeChangedOrRemoved: "true"|"false"]
  connectionInfoHoursTTL: 168
  historyCountsMinutesTTL: 1440

# Multimedia Disaster Recovery
drMonitoringDelay: 1800

# DoS Filter Settings
enableDosFilter: [ToBeChanged: "true"|"false"]
dosFilterSettings:
  maxRequestsPerSec: 25
  delayMs: 100
  maxWaitMs: 50
  throttledRequests: 5
  throttleMs: 30000
```

```
maxRequestMs: 30000
maxIdleTrackerMs: 30000
insertHeaders: [ToBeChangedOrRemoved: <DOS_FILTER_INSERT_HEADERS>]
trackSessions: [ToBeChangedOrRemoved: <DOS_FILTER_TTACK_SESSIONS>]
remotePort: [ToBeChangedOrRemoved: <DOS_FILTER_REMOTE_PORT>]
ipWhitelist: [ToBeChangedOrRemoved: <DOS_FILTER_IP_WHITE_LIST>]

multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 67108864

# Media Life Cycle Management
backgroundScheduledMediaOperationsSettings:
  enableBackgroundScheduledMediaOperations: [ToBeChangedOrRemoved: "true"|"false"]
  schedulerThreads: 4
  schedulePollingInterval: 60
  speechMinerMaxConnection: 20
  speechMinerMaxTotalConnection: -1
  speechMinerSocketTimeout: 60000
  defaultBackupExportURI: [ToBeChangedOrRemoved: <DEFAULT_BACKUP_EXPORT_URI>]
  useFullPathInMediaFileBackup: false
  enableScanAndScroll: [ToBeChangedOrRemoved: "true"|"false"]
  scanIntervalsPerDay: [ToBeChangedOrRemoved: <SCHEDULE_MEDIA_OPERATION_SCAN_INTERVALS_PER_DAY>]

# CometD Settings
cometDSettings:
  cometdSessionExpirationTimeout: 60
  closeHttpSessionOnCometDExpiration: true
  maxSessionsPerBrowser: 1
  multiSessionInterval: 2000

# Log Header Settings
logHeaderSettings:
  enableLogHeader: [ToBeChangedOrRemoved: "true"|"false"]
  updateOnPremiseInfoInterval: 600

# Update on startup settings
updateOnStartup:
  opsCredentials: false
  features: false
```

Important

If you are using Elasticsearch 7.16.3 (or higher), refer to the below **elasticSearchSettings** section for setup.

```
# Elasticsearch
elasticSearchSettings:
  retriesOnConflict: 3
  useRestClient: true
  restClient:
    nodes: [ToBeChanged: <ELASTIC_SEARCH_REST_NODES>]
    authentication:
      type: [ToBeChangedOrRemoved: none|basic]
      userName: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_USER_NAME>]
      password: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_PASSWORD>]
    useSSL: [ToBeChangedOrRemoved: "true"|"false"]
    truststore: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PATH>]
    truststorePassword: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PASSWORD>]
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCAN_READ_TIMEOUT_SECONDS>]
  scrollTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCROLL_TIMEOUT_SECONDS>]
  countReadTimeoutSeconds: [ToBeChangedOrRemoved:
<ELASTIC_SEARCH_COUNT_READ_TIMEOUT_SECONDS>]
```

Make sure you update all settings marked as [ToBeChanged]. You should also be sure to do the following:

- Set the **applicationName** to the name of the application you created in [Creating the IRWS Node Application](#) — for example, IRWS_Node.
- In each Interaction Recording Web Services cluster or shared Interaction Recording Web Services and Web Services and Applications cluster, if both are deployed, one node in the cluster must be configured as the synchronization node: `syncNode: true`. All other nodes in the cluster must have `syncNode: false`.

Important

- To create the **ops** user and credentials in Cassandra and to enable the features in the **Interaction Recording Web Services** node, set the following parameters to true during the first Interaction Recording Web Services startup in the **application.yaml** file:

updateOnStartup

opsCredentials: true

features: true

After Interaction Recording Web Services is started, you must change both options to false for production:

updateOnStartup
opsCredentials: false
features: false

- A User object with user name set to default is a predefined object from Configuration Database and it is referred to as the Master Account. The Master Account is not alterable in any way, and you should not use it to perform regular contact center administrative tasks. For more information, see [Configuration Database](#).

The synchronization node is not responsible for importing the user name called default from Configuration Server into Cassandra, subscribing to change notifications with Configuration Server, or processing updates.

See [serverSettings](#) for details about all supported configuration settings for this section.

On Premise Settings

The settings in the **onPremiseSettings** section instruct Interaction Recording Web Services on how to communicate with the Configuration Server. The **application.yaml.sample** file includes the following default **onPremiseSettings** section:

```
# On Premise Settings (when syncNode is true)
onPremiseSettings:
  cmeHost: [ToBeChanged: <CONFIG_SERVER_HOST>]
  cmePort: [ToBeChanged: <CONFIG_SERVER_PORT>]
  backupCmeHost: [ToBeChanged: <BACKUP_CONFIG_SERVER_HOST>]
  backupCmePort: [ToBeChanged: <BACKUP_CONFIG_SERVER_PORT>]
  countryCode: [ToBeChanged: "US" | "CA" | etc]
  tlsEnabled: [ToBeChangedOrRemoved: "true"|"false"]
```

Make sure you update all settings marked as [ToBeChanged]. See [onPremiseSettings](#) for details about all supported configuration settings for this section.

Important

Note that settings under **onPremiseSettings** are used only once during the first initialization of RWS on the sync node. Further changes in the environment are retrieved from the Configuration Server directly. If a setting is configured incorrectly, please contact Genesys Customer Care for support.

Tuning the Interaction Recording Web Services Host Performance

Complete the following steps on each **Interaction Recording Web Services** node to tune the performance of the host environment.

Start

1. To optimize TCP/IP performance, add the following to the **/etc/sysctl.conf** file:

```
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 16384 16777216
net.core.somaxconn=4096
net.core.netdev_max_backlog=16384
net.ipv4.tcp_max_syn_backlog=8192
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_congestion_control=cubic
```

2. Increase the file descriptors by adding the following to the **/etc/security/limits.conf** file:

```
gir          hard nofile          100000
gir          soft nofile          100000
```

3. Run **sysctl -p** to reload the new values. These values will now always be loaded when rebooting.

End

Enabling features in the Feature Definitions file

The Feature Definitions file contains a list of features that are available for your contact center. The file is used to define features for the contact center by both Web Services (when installed) and Interaction Recording Web Services. For this reason, the procedure has a dependency on whether Web Services is being deployed along with Interaction Recording Web Services.

Perform the following operations on one of the **Interaction Recording Web Services** nodes.

Important

From Interaction Recording Web Services version 8.5.500.30 (or higher), Interaction Recording Web Services only Installations is supported .

For Web Services and Interaction Recording Web Services Installations

1. Locate the **gir-feature-definitions.json** file in the **installation_CD/config-templates** folder.

2. If you have already followed the [Enabling features in the Feature Definitions](#) file instructions from the *Web Services and Applications Deployment Guide (GWS)*, locate the **feature-definitions.json** file that was installed and edited into the **GWS_CONF** folder on the **Web Services** nodes.
If you did not already follow the [Enabling features in the Feature Definitions](#) file instructions, locate the **gws-feature-definitions.json** file in the **installation_CD/config-templates** folder.
3. Merge the contents of the two files together into a **feature-definitions.json** file in the **GWS_CONF** folder, as follows:
 - a. Ensure there is only one set of enclosing [...] (for example, first and last lines).
 - b. Ensure there is a comma after each { ... } excluding the last.
 - c. Ensure there are no duplicate items, for instance **api-provisioning-read** and **api-provisioning-write**.
4. Edit the file and for each feature that you want to enable for a new contact center, set the **autoAssignOnContactCenterCreate** flag to true. If you have already created your contact center or you are unsure of which Interaction Recording Web Services features to enable at this point, leave the **autoAssignOnContactCenterCreate** flags as they appear.

Important

The instructions that follow provide more detail about the Interaction Recording Web Services features and how to enable or disable them using **REST API** endpoints. For additional information, refer to [Configuring Features](#).

Merged Feature Definitions File - Example [+] [Show example.](#)

```
[
  {
    "id":"api-provisioning-read",
    "displayName":"API Provisioning Read",
    "description":"General provisioning read",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-provisioning-write",
    "displayName":"API Provisioning Write",
    "description":"General provisioning write",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-voice",
    "displayName":"Voice API",
    "description":"API for Voice",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-voice-predictive-calls",
    "displayName":"Voice API - Predictive calls",
    "description":"Enables predictive calls for a contact center",
    "autoAssignOnContactCenterCreate":true
  },
  {

```

```
    "id": "api-voice-outbound",
    "displayName": "Voice API Outbound",
    "description": "API for Outbound",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-supervisor-agent-control",
    "displayName": "API Supervisor Agent Control",
    "description": "API for Supervisors to Control Agent State",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-supervisor-monitoring",
    "displayName": "API Supervisor Monitoring",
    "description": "API for Supervisors to Monitor Agents",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-multimedia-chat",
    "displayName": "Multimedia Chat API",
    "description": "API for Multimedia Chat",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-email",
    "displayName": "Multimedia Email API",
    "description": "API for Multimedia Email",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-facebook",
    "displayName": "Multimedia Facebook API",
    "description": "API for Multimedia Facebook",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-twitter",
    "displayName": "Multimedia Twitter API",
    "description": "API for Multimedia Twitter",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-workitem",
    "displayName": "Multimedia Workitem API",
    "description": "API for Multimedia Workitem",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-user-account-management-email",
    "displayName": "User Account Management via Email",
    "description": "API for account management via email",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-devices-webrtc",
    "displayName": "WebRTC Support",
    "description": "API for WebRTC provisioning",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-ucs-voice",
    "displayName": "Support UCS for voice",
    "description": "For support contact center in voice",
```

```
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-voice-instant-messaging",
    "displayName":"API Voice Instant Messaging",
    "description":"API for Internal Agent-to-Agent Chat",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-platform-configuration-read",
    "displayName":"Platform Configuration API - read",
    "description":"Low-level configuration API",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-platform-configuration-write",
    "displayName":"Platform Configuration API - write",
    "description":"Low-level configuration API",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-voice-recording",
    "displayName":"Voice API Recording",
    "description":"API for Voice Recording",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-voice-screenrecording",
    "displayName":"Screen Recording API (Voice)",
    "description":"API for Agent Voice Screen Recording",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-supervisor-recording",
    "displayName":"API Supervisor Recording",
    "description":"API for Call Recording Supervisor",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-multimedia-screenrecording",
    "displayName":"Screen Recording API (Multimedia)",
    "description":"API for Agent Multimedia Screen Recording",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-recordings-decryption-proxying",
    "displayName":"API Recordings Decryption Proxying",
    "description":"API For HTCC proxied interaction recording decryption",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-screenrecording-connection-reporting",
    "displayName":"API Screen Recording Connections Reporting",
    "description":"APIs for reporting on screen recording client connections",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"schema-elasticsearch-v2-call-recording",
    "displayName":"Schema Elasticsearch Call Recording Index V2",
    "description":"Elasticsearch call recording index schema v2",
    "autoAssignOnContactCenterCreate":true
  },
  {

```

```

    "id": "schema-elasticsearch-migration-to-v2-call-recording",
    "displayName": "Schema Elasticsearch Migration To Call Recording Index V2",
    "description": "Elasticsearch call recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "schema-elasticsearch-v2-screen-recording",
    "displayName": "Schema Elasticsearch Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "schema-elasticsearch-migration-to-v2-screen-recording",
    "displayName": "Schema Elasticsearch Migration To Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "data-skip-attach-screenrecording-data-to-callrecording",
    "displayName": "Skip Attaching Screen Recording Data To Call Recording",
    "description": "Whether or not to skip attaching screen recording data to call recording metadata",
    "autoAssignOnContactCenterCreate": false
  }
]

```

5. Follow the steps in the [Ensuring the Feature Definitions file is Read at Start-Up](#) section.

For Interaction Recording Web Services Only Installations

1. Locate the **gir-feature-definitions.json** file in the **installation_CD/config-templates** folder.
2. Copy the file to **feature-definitions.json** file in the **GWS_CONF** folder, and open the file.
3. For each feature that you want to enable for a new contact center, set the **autoAssignOnContactCenterCreate** flag to **true**. If you are unsure of which Interaction Recording Web Services features to enable, leave them as they appear.

Important

The instructions that follow provide more detail about the Interaction Recording Web Services features and how to enable or disable them using **REST API** endpoints. For additional information, refer to [Configuring Features](#).

Feature Definitions File - Example [\[+\] Show example.](#)

```

{
  "id": "api-provisioning-read",
  "displayName": "API Provisioning Read",
  "description": "General provisioning read",

```

```
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-provisioning-write",
    "displayName": "API Provisioning Write",
    "description": "General provisioning write",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-voice-recording",
    "displayName": "Voice API Recording",
    "description": "API for Voice Recording",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-voice-screenrecording",
    "displayName": "Screen Recording API (Voice)",
    "description": "API for Agent Voice Screen Recording",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-supervisor-recording",
    "displayName": "API Supervisor Recording",
    "description": "API for Call Recording Supervisor",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-multimedia-screenrecording",
    "displayName": "Screen Recording API (Multimedia)",
    "description": "API for Agent Multimedia Screen Recording",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-recordings-decryption-proxying",
    "displayName": "API Recordings Decryption Proxying",
    "description": "API For HTCC proxied interaction recording decryption",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-screenrecording-connection-reporting",
    "displayName": "API Screen Recording Connections Reporting",
    "description": "APIs for reporting on screen recording client connections",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "schema-elasticsearch-v2-call-recording",
    "displayName": "Schema Elasticsearch Call Recording Index V2",
    "description": "Elasticsearch call recording index schema v2",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "schema-elasticsearch-migration-to-v2-call-recording",
    "displayName": "Schema Elasticsearch Migration To Call Recording Index V2",
    "description": "Elasticsearch call recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "schema-elasticsearch-v2-screen-recording",
    "displayName": "Schema Elasticsearch Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2",
    "autoAssignOnContactCenterCreate": true
  },
  {
  }
```

```
    "id": "schema-elasticsearch-migration-to-v2-screen-recording",
    "displayName": "Schema Elasticsearch Migration To Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "data-skip-attach-screenrecording-data-to-callrecording",
    "displayName": "Skip Attaching Screen Recording Data To Call Recording",
    "description": "Whether or not to skip attaching screen recording data to call recording metadata",
    "autoAssignOnContactCenterCreate": false
  }
]
```

4. Follow the steps in the [Ensuring the Feature Definitions file is Read at Start-Up](#) section.

Ensuring the Feature Definitions file is Read at Start-Up

The Feature Definitions file is by default not read at start-up.

To ensure that it is read at start-up:

1. Add the following setting to **application.yaml** under the **serverSettings** section, on one of the **Interaction Recording Web Services** nodes:

```
updateOnStartup:
  features: true
```

2. Restart the **Interaction Recording Web Services** node.
3. Ensure you remove the setting after Interaction Recording Web Services has been started.

Important

Instructions about starting can be found in the [Starting and Testing](#) page.

Next Step

- [Configure additional security \(optional\)](#).

Configuring Security

Web Services adheres to the standards described in the Open Web Application Security Project (OWASP) Top 10—see the [OWASP website](#) for details—and has adopted several methods of ensuring security, for example:

- Errors are logged locally to prevent information leakage through API requests.
- User sessions have a timeout option.
- Cross Site Request Forgery Protection

Read on for details about the additional security configurations that Interaction Recording Web Services includes.

Transport Layer Security (TLS)

Complete the procedures below to configure TLS for connections received by Interaction Recording Web Services, and for connections from Interaction Recording Web Services to the following:

- Configuration Server
- SIP Server
- Interaction Server
- WebDAV
- Recording Crypto Server
- SpeechMiner Interaction Receiver
- Cassandra

Configuring TLS on the Server Side for Interaction Recording Web Services

1. Enable SSL on Jetty by configuring the **SSL** section of the **application.yaml** file using the following parameters:

```
enableSsl: true
ssl:
  port: 443
  keyStorePath: keystore
  keyStorePassword: storepwd
```

For more information on the parameters, see [ssl](#).

Important

On Unix-based systems, port 443 is protected; typically, only the superuser root can open it. For security reasons, it is not recommended to run the server as root. Therefore, Genesys recommends that you bind it to a non-protected port. Typically, any port above 1024 can be used (for example, you could set it to 9443). If you want to continue using port 443, see [Setting Port 80 Access for a Non-Root User](#) in the Jetty documentation.

2. Acquire the certificate and private keys.
3. To load a certificate and private keys (jetty.crt), navigate to the GWS_HOME/etc directory and run the following commands:

```
keytool -keystore keystore -import -alias jetty -file jetty.crt -trustcacerts
```
4. When prompted for the keystore password, enter the default: storepwd
5. Restart Interaction Recording Web Services (Web Services).

To create a self-signed certificate for non-production purposes:

1. Run the following in GWS_HOME/etc:

```
keytool -genkey -keyalg RSA -keystore keystore -alias jetty -ext SAN=dns:<server_dns_name>,ip:<server_ip_address>
```
2. When prompted for the keystore password, enter the default: storepwd
For more information about configuring SSL, see [Configuring SSL/TLS](#).

To change the certificate:

1. Remove the existing certificate using the following command:

```
keytool -keystore keystore -delete -alias jetty
```
2. Acquire the certificate and private key in a X509 PEM file (for example, jetty.crt).
3. Load the certificate using the following command:

```
keytool -keystore keystore -import -alias jetty -file jetty.crt -trustcacerts
```
4. Restart Interaction Recording Web Services (Web Services).

To change the keystore password:

1. Execute the following command:

```
keytool -keystore keystore -storepasswd
```
2. Encode the new password using the following command:

```
java -cp lib/jetty-http-xxx.jar:lib/jetty-util-xxx.jar org.mortbay.jetty.security.Password <your password here>
```

Configuring TLS connections to Configuration Server, SIP Server, and Interaction

Server

Interaction Recording Web Services can use a secured Transport Layer Security (TLS) connection mechanism to connect to Configuration Server, Interaction Server, and SIP Server. When configured, Interaction Recording Web Services connects to secure ports on Configuration Server, Interaction Server, and SIP Server; verifies the server's certificate; and encrypts/decrypts network traffic. You can configure secured connections to Configuration Server, Interaction Server, and SIP Server in the following ways:

- [Minimal configuration for Configuration Server, SIP Server, and Interaction Server](#)
- [Validate the Certificate Against the CA](#)

Note that each connection is configured independently, but a similar mechanism is used to configure each connection.

Prerequisites

Before configuring Interaction Recording Web Services, make sure the secure port on the server is configured as described in [Introduction to Genesys Transport Layer Security](#) in the [Genesys Security Deployment Guide](#) and that certificates for the server and the Certificate Authority are configured and available.

Minimal configuration for Configuration Server, SIP Server, and Interaction Server

In this configuration, Interaction Recording Web Services does not check the certificate against the Certificate Authority, but all traffic is encrypted. To configure Interaction Recording Web Services with minimal configuration, all you need to do is configure a connection to a secured port on Configuration Server, SIP Server, and Interaction Server. You can do this using *either* of the following methods:

- For the initial connection to Configuration Server, set the `tlsEnabled` option to `true` in the **onPremiseSettings** section of the RWS **application.yaml** file on the RWS node that is configured to be the sync node. This creates a secured connection to Configuration Server the first time Interaction Recording Web Services starts.
- For an environment that is already configured with Configuration Server synchronization enabled, you can make changes with Configuration Server as described in the [Genesys Security Deployment Guide](#). These changes are synchronized back to the Cassandra database from Configuration Server.

Important

Configuration Server supports the auto-upgrade port connection for secure communication from other GIR components; however, a secure port (listening mode of type secured) must be used for connectivity from RWS to the Configuration Server.

Ensure that connections from the Cluster Application being used by RWS (either `IRWS_Cluster` or `WS_Cluster`; see [Installing Interaction Recording Web Services](#) for more information) specify the appropriate secure port on each of the servers.

Validate the Certificate Against the CA

The procedure to validate the certificate against the CA is common to Configuration Server, SIP Server and Interaction Server.

Ensure you have completed the procedure described in the [Minimal configuration for Configuration Server, SIP Server, and Interaction Server](#) section.

To support the client-side certificate check, Interaction Recording Web Services needs the public key for the Certificate Authority (CA). Interaction Recording Web Services supports the PEM and JKS key storage formats, but recommends using JKS because it's compatible with both Cassandra and HTTPS.

To validate the certificate against the CA, specify the path to a file containing the trusted CA in the **caCertificate** parameter in the **application.yaml** file. By specifying this parameter, this CA will be checked against the server CA for validation.

Important

Only a single CA can be used to validate the certificate from Configuration Server, SIP Server, and Interaction Server.

If the configured server certificate matches the hostname of the server for any of the following fields, then Interaction Recording Web Services will validate the certificate.

- Issuer CN
- Subject CN
- Subject Alternative Name DNS

To validate the certificate against the CA, complete the following steps.

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. If you plan to use a JKS file, you can generate it from a PEM file by importing the PEM certificate, as shown here:

```
keytool -importcert -file ca_cert.pem -keystore ca_cert.jks
```

2. Once you have the **ca_cert.jks** file, place it in a location accessible from your Interaction Recording Web Services host, such as:
 - A local folder on the Interaction Recording Web Services host

- A shared folder

3. Configure the following options in the **serverSettings** section of the **application.yaml** file:

- For a PEM file, set **caCertificate** to the location of the file. For example:

```
caCertificate: /opt/ca_cert.pem
```

- For a JKS file, set **caCertificate** to the location of the file and set **jksPassword** to the password for the key storage. For example:

```
caCertificate: /opt/ca_cert.jks
jksPassword: pa$$word
```

End

Configuring TLS for Connections to WebDAV

By default, Interaction Recording Web Services checks the WebDAV server's certificate against a Certificate Authority using the Java default trustStore **caCerts**. To configure Interaction Recording Web Services with a customized trustStore configuration or to disable certificate validation, set the following options in the **application.yaml** configuration file.

Name	Parent	Value	Default
webDAVTrustedCA	serverSettings	<p>Configures TLS certificate validation when Interaction Recording Web Services connects to a WebDAV server. Valid values are <code>true</code>, <code>false</code>, or a path to a file containing one or more CA certificates.</p> <ul style="list-style-type: none"> • If set to <code>true</code>, the certificate that WebDAV presents will be validated by caCerts in <code>\$JAVA_HOME/jre/lib/security</code>. • If set to <code>false</code>, the certificate that WebDAV presents will not be validated. • Any other value is considered as a path to a file containing a certificate for a Certificate Authority and RWS will use it to validate the WebDAV certificate. Both PEM and JKS 	true

Name	Parent	Value	Default
		key storage formats are supported. If the specified file does not exist, Interaction Recording Web Services will exit during initialization.	
webDAVJksPassword	serverSettings	The password for the key storage for WebDAV if the specified CA file is in JKS format. You can specify an encrypted password. For more information on encrypting a password, see Password Encryption .	Empty

Configuring TLS for Connections to Recording Crypto Server

By default, Interaction Recording Web Services checks the Recording Crypto Server's certificate against a Certificate Authority using the Java default trustStore **caCerts**. To configure Interaction Recording Web Services with a customized trustStore configuration or to disable certificate validation, set the following options in the **application.yaml** configuration file.

Name	Parent	Value	Default
rcsTrustedCA	serverSettings	<p>Configures TLS certificate validation when Interaction Recording Web Services connects to the Recording Crypto Server. This property can be set to <code>true</code>, <code>false</code>, or a path to a file containing one or more CA certificates.</p> <ul style="list-style-type: none"> If set to <code>true</code>, the certificate that RCS presents will be validated by caCerts in <code>\$JAVA_HOME/jre/lib/security</code>. If set to <code>false</code>, the certificate that RCS presents will not be validated. 	true

Name	Parent	Value	Default
		<ul style="list-style-type: none"> Any other value is considered as a path to a file containing one or more CA certificates and RWS will use it to validate the RCS certificate. The key storage format can be either PEM or JKS. If the specified file does not exist, Interaction Recording Web Services will exit during initialization. 	
rcsJksPassword	serverSettings	The password for the key storage for RCS if the specified CA file is in JKS format. You can specify an encrypted password. For more information on encrypting a password, see Password Encryption .	Empty

Configuring TLS for Connections to SpeechMiner Interaction Receiver

By default, Interaction Recording Web Services checks the SpeechMiner Interaction Receiver's certificate against a Certificate Authority using the Java default trustStore **caCerts**. To configure Interaction Recording Web Services with a customized trustStore configuration or to disable certificate validation, set the following options in the **application.yaml** configuration file.

Name	Parent	Value	Default
speechMinerTrustedCA	serverSettings	<p>Configures TLS certificate validation when Interaction Recording Web Services connects to SpeechMiner Interaction Receiver. Valid values are true, false, or a path to a file containing one or more CA certificates.</p> <ul style="list-style-type: none"> If set to true, the certificate that the SpeechMiner Interaction Receiver 	true

Name	Parent	Value	Default
		<p>presents will be validated by caCerts in \$JAVA_HOME/jre/lib/security.</p> <ul style="list-style-type: none"> If set to false, the certificate that the SpeechMiner Interaction Receiver presents will not be validated. Any other value will be considered as a path to a file containing one or more CA certificates and RWS will use it to validate the SpeechMiner Interaction Receiver certificate. The key storage format can be either PEM or JKS. If the specified file does not exist, Interaction Recording Web Services will exit during initialization. 	
speechMinerJksPassword	serverSettings	<p>The password for the key storage for SpeechMiner Interaction Receiver if the specified CA file is in JKS format. You can specify an encrypted password. For more information on encrypting a password, see Password Encryption.</p>	Empty

Configuring TLS for Connections with Cassandra

Genesys supports Transport Layer Security (TLS) for connections from Interaction Recording Web Services to Cassandra and between Cassandra nodes. You can configure secured connections for the following scenarios:

- [Secure Connections from Interaction Recording Web Services to Cassandra](#)
- [Secure Connections from Interaction Recording Web Services to Cassandra 4](#)

- [Secure Connections between Cassandra Nodes](#)

Secure Connections from Interaction Recording Web Services to Cassandra

Prerequisites

- You have installed [Bash](#), [Java keytool](#), and [OpenSSL](#).

Complete the following steps to configure TLS for connections from Interaction Recording Web Services to Cassandra.

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. Create the server-side keystore with a self-signed certificate and the client-side truststore — which contains the public part of server certificate — with the following commands:

```
#!/bin/bash
#generate keypair
keytool -genkeypair -alias cassandra -keyalg RSA -keysize 1024 -dname
"CN=<Cassandra node hostname>, OU=Test, O=Test Ltd, C=US" -keystore
server.jks
-storepass password -keypass password
#export certificate
keytool -exportcert -alias cassandra -file client.pem -keystore
server.jks -storepass password -rfc
#create client truststore and import certificate
keytool -importcert -alias cassandra -file client.pem -keystore
client.jks -storepass password -noprompt
```

2. Create a self-signed root authority, use it to sign the server certificate, store it to **server.jks** and create the client-side truststore, which trusts all certificates signed with root authority. Run the following commands:

```
#!/bin/sh

#generate self-signed root certificate
keytool -genkeypair -alias root -keyalg RSA -keysize 1024 -validity 3650 -dname
"CN=TestRoot, OU=Dev, O=Company, C=US" -keystore root.jks
-storepass password -keypass password

#export root certificate
keytool -exportcert -alias root -file root.crt -keystore root.jks -storepass password

#generate server-side certificate
keytool -genkeypair -alias server -keyalg RSA -keysize 1024 -validity 3650 -dname
"CN=TestServer, OU=Dev, O=Company, C=US"
-keystore server.jks -storepass password -keypass password
```

```
#create the sign request for server certificate
keytool -certreq -alias server -keystore server.jks -file server.csr -storepass password
-keypass password

#export private key of root auth: need later for signing the server certificate
keytool -v -importkeystore -srckeystore root.jks -srcalias root -destkeystore root.p12
-deststoretype PKCS12 -noprompt
-destkeypass password -srckeypass password -destalias root -srcstorepass password
-deststorepass password

openssl pkcs12 -in root.p12 -out private.pem -password pass:password -passin
pass:password -passout pass:password
rm root.p12

#sign the certificate
openssl x509 -req -CA private.pem -in server.csr -out server.crt -days 3650
-CAcreateserial -passin pass:password
rm private.pem
rm private.srl
rm server.csr

#import root certificate to client side trust store
keytool -importcert -alias root -file root.crt -keystore client.jks -storepass password
-noprompt

#import root certificate to server side key store
keytool -importcert -alias root -file root.crt -keystore server.jks -storepass password
-noprompt
rm root.crt

#import certificate sign reply into server-side keystore
keytool -import -trustcacerts -alias server -file server.crt -keystore server.jks
-storepass password -keypass password
rm server.crt
```

3. Configure Cassandra to use your generated certificates for the client connection by setting the `client_encryption_options` in the **cassandra.yaml** file. For example:

```
client_encryption_options:
  enabled: true
  keystore: <absolute path to server.jks file>
  keystore_password: password
  #the password specified in while creating storage
  # For the purpose of the demo the default settings were used.
  # More advanced defaults below:
  #protocol: TLS
  #algorithm: SunX509
  #store_type: JKS
  #cipher_suites: [TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA]
```

Important

To enable support for encryption, you must have the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction installed.

4. Confirm the Cassandra nodes can start successfully:
 - a. Edit the **conf/log4j-server.properties** file and uncomment the following line:

```
log4j.logger.org.apache.cassandra=DEBUG
```

- b. Start Cassandra and check the logs. If the configuration was successful, you shouldn't see any errors.
- c. Edit the **conf/log4j-server.properties** file and comment the following line to disable the functionality:

```
log4j.logger.org.apache.cassandra=DEBUG
```

- d. Check that SSL-to-client is working successfully using cqlsh.
 - Confirm that unsecured connections aren't possible by starting cqlsh locally—this forces it to connect to the Cassandra instance running on localhost. You should expect to see the exception in the cqlsh output.

```
cqlsh `hostname`
```

- Confirm that secured connections are possible by configuring cqlsh with SSL encryption. Create a PEM key which will be used in the **.cqlshrc** file:

```
// Create PEM for client
keytool -importkeystore -alias cassandra -srckeystore
server.jks -destkeystore server.p12 -deststoretype PKCS12
openssl pkcs12 -in server.p12 -out client.pem -nodes
```

- Create a **.cassandra/cqlshrc** file in your home or client program directory. The following settings must be added to the file as described below. When validate is enabled, the host in the certificate is compared to the host of the machine that it is connected to verify that the certificate is trusted.

```
[connection]
hostname = <hostname of Cassandra node>
port = 9042
factory = cqlshlib.ssl.ssl_transport_factory
```

```
[ssl]
certfile = /path/to/client.pem
# Optional, true by default
validate = true
```

- Verify that you can connect to Cassandra using cqlsh:

```
$ cqlsh --ssl
Connected to HTCC Cassandra Cluster at
ci-vm378.us.int.genesyslab.com:9042.
[cqlsh 5.0.1 | Cassandra 2.2.3 | CQL spec 3.3.1 | Native
protocol v4]
Use HELP for help.
```

- Configure Interaction Recording Web Services to use SSL with Cassandra. On each Interaction Recording Web Services node, edit the **application.yaml** file as follows:

```
cassandraCluster:
  useSSL: true
  trustStore: /path/to/client.jks
  truststorePassword: password
```

- Restart each Interaction Recording Web Services node:

```
systemctl restart gir
```

End

Secure Connections from Interaction Recording Web Services to Cassandra 4

Start

1. Create a self-signed root authority, use it to sign the server certificate, store it to server-side keystore and create the client-side truststore, which trusts all certificates signed with root authority. Run the following commands:

```
#!/bin/bash
#generate self-signed root certificate
keytool -genkeypair -alias root -keyalg RSA -keysize 2048 -validity 3650 -dname
"CN=TestRoot, OU=Dev, O=Company, C=US" -keystore root.jks -storepass password
-keypass password -ext KeyUsage=digitalSignature,keyCertSign -ext
BasicConstraints=ca:true

#export root certificate
keytool -exportcert -alias root -file root.crt -keystore root.jks -storepass password

#export private key of root auth: need later for signing the server certificate
keytool -v -importkeystore -srckeystore root.jks -srcalias root -destkeystore
root.p12 -deststoretype PKCS12 -noprompt -destkeypass password -srckeypass password
-destalias root -srcstorepass password -deststorepass password
openssl pkcs12 -in root.p12 -out private.pem -password pass:password -passin
pass:password -passout pass:password -legacy
rm root.p12

#import root certificate to client side trust store
keytool -importcert -alias root -file root.crt -keystore client.jks -storepass
password -noprompt

#import root certificate to server side key store
keytool -importcert -alias root -file root.crt -keystore server.jks -storepass
password -noprompt
rm root.crt
```

2. Create the server-side keystore with a self-signed certificate and the client-side truststore — which contains the public part of server certificate — with the following commands:

Important

In case of Multi Node Cassandra, the following steps should be executed for each Cassandra node.

```
#!/bin/bash
#generate server-side certificate
keytool -genkeypair -alias <node> -keyalg RSA -keysize 2048 -dname "CN=<Cassandra
node>, OU=Test, O=Test Ltd, C=US" -keystore server.jks -storepass password -keypass
password

#export certificate
keytool -exportcert -alias <node> -file <node>_client.pem -keystore server.jks
-storepass password -rfc

#create and import certificate to client truststore
keytool -importcert -alias <node> -file <node>_client.pem -keystore client.jks
-storepass password -noprompt

#create the sign request for server certificate
```

```
keytool -certreq -alias <node> -keystore server.jks -file <node>.csr -storepass
password -keypass password

#sign the certificate
openssl x509 -req -CA private.pem -in <node>.csr -out <node>.cert -days 3650
-CAcreateserial -passin pass:password

rm private.pem
rm private.srl
rm <node>.csr

#import certificate sign reply into server-side keystore
keytool -import -trustcacerts -alias <node_cert> -file <node>.cert -keystore server.jks
-storepass password -keypass password
rm <node>.cert
```

3. Continue from Step.3 in [Secure Connections from Interaction Recording Web Services to Cassandra](#).

End

Secure Connections between Cassandra Nodes

When you enable SSL for connections between Cassandra nodes, you ensure that communication between nodes in the Cassandra cluster is encrypted, and that only other authorized Cassandra nodes can join the cluster.

The steps below show you how to create a single certificate to be used by all Cassandra nodes in the cluster. This simplifies cluster management because you don't need to generate a new certificate each time you add a new node to the cluster, which means you don't need to restart all nodes to load the new certificate.

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. Generate a keystore and truststore. See Step 1 of [Secure Connections from Interaction Recording Web Services to Cassandra](#) for details.
2. On each Cassandra node in the cluster, set **server_encryption_options** in the **cassandra.yaml** file. For example:

```
server_encryption_options:
  internode_encryption: all
  keystore: <absolute path to keystore >
  keystore_password: <keystore password - somePassword in our sample>
  truststore: <absolute path to trustore>
  truststore_password: <truststore password - somePassword in our sample>
```

3. Check the Cassandra logs. If the configuration was successful, you shouldn't see any errors.

End

Configuring TLS for Connections with Elasticsearch and Elasticsearch basic authentication

Genesys supports Transport Layer Security (TLS) and authentication for connections from Interaction Recording Web Services (RWS) to Elasticsearch 8 and also for communication between Elasticsearch nodes. You can configure these connections manually or use Elasticsearch's built-in autoconfiguration tools.

- [Secure Connections from Interaction Recording Web Services to Elasticsearch 8](#)
- [Secure Connections between Elasticsearch Nodes](#)

Secure Connections from Interaction Recording Web Services to Elasticsearch 8

Configuring RWS with Elasticsearch Default certificates:

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

1. Elasticsearch 8 comes with security features enabled by default, even without any additional configuration. On the first startup, it automatically:
 1. Enables basic authentication.
 2. Generates a random password for the built-in elastic superuser.
 3. Enables HTTPS on localhost with a self-signed certificate.
 4. Generates default certificates and keystores for HTTP and transport layers.

Once Elasticsearch 8 has successfully started, the following files are generated in the Elasticsearch configuration directory (/etc/elasticsearch or where the **elasticsearch.yml** resides).

File	Purpose
http_ca.crt	The Certificate Authority (CA) certificate that signed the HTTP layer certificate of this cluster. Clients connecting over HTTPS should trust this CA.
http.p12	A PKCS#12 keystore containing the key and certificate for the HTTP layer of this node.
transport.p12	A PKCS#12 keystore containing the key and certificate for the transport layer, used for secure communication between all nodes in the cluster.

2. RWS configuration sample
 1. We can use http_ca.crt to create a truststore for RWS connecting to Elasticsearch over HTTPS.
 2. We can find <elastic user password in Elasticsearch logs.

3. Configure Interaction Recording Web Services to use SSL with Elasticsearch. On each Interaction Recording Web Services node, edit the **application.yaml** file as follows:

```
elasticSearchSettings:
  ...
  restClient:
    nodes:
      - {host: <elastic-search-node>, port: 9200}
    authentication:
      type: basic
      userName: elastic
      password: <ELASTIC_USER_PASSWORD>
    useSSL: true
    truststore: rws-client.jks # contains http_ca.crt
    truststorePassword: password
```

3. Restart each Interaction Recording Web Services node:

```
systemctl restart gir
```

Configuring RWS with Elasticsearch Manually generated certificates:

1. Generate Elasticsearch Node Keystore & Certificate

```
keytool -genkeypair -alias <elasticsearch-node> -keyalg RSA -keysize 2048 -dname
"CN=<Elasticsearch node hostname>, OU=Test, O=Test Ltd, C=US" -keystore es-server.jks
-storepass password -keypass password
keytool -exportcert -alias <elasticsearch-node> -file es-node.pem -keystore es-
server.jks -storepass password -rfc
```

2. Create Client-side Truststore for RWS

```
keytool -importcert -alias elasticsearch-node -file es-node.pem -keystore rws-
client.jks -storepass password -noprompt
```

3. To Create Additional Users in elasticsearch

```
<Elasticsearch_installation_directory>/bin/elasticsearch-users useradd <rws-user> -p
<password> -r <superuser>
```

4. Configure SSL options for Elasticsearch by editing the `elasticsearch.yml` in all elasticsearch nodes. For example:

```
xpack.security.enabled: true
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.keystore.path: /path/to/es-server.jks
xpack.security.http.ssl.keystore.password: password
```

5. Restart Elasticsearch services and check the logs. If the configuration was successful, you shouldn't see any errors.

1. Verify cluster health:

```
curl -u <rws-user>:<password> --cacert /path/to/ca.crt
https://<host>:9200/_cluster/health
```

6. Configure Interaction Recording Web Services to use SSL with Elasticsearch. On each Interaction Recording Web Services node, edit the **application.yaml** file as follows:

```
elasticSearchSettings:
  ...
  restClient:
    nodes:
```

```
- {host: <elastic-search-node1>, port: 9200}
authentication:
  type: basic
  userName: <ELASTIC_SEARCH_USER_NAME>
  password: <ELASTIC_SEARCH_PASSWORD>
  useSSL: true
  truststore: rws-client.jks
  truststorePassword: password
```

7. Restart each Interaction Recording Web Services node:

```
systemctl restart gir
```

Configuring RWS with Elasticsearch Existing certificates:

1. Make sure you have the Elasticsearch CA certificate file.
2. Create the RWS truststore in .p12 format using keytool:

```
keytool -importcert -trustcacerts -alias elastic-ca -file elastic-stack-ca.crt
-keystore rws-truststore.p12 -storetype PKCS12
```

Elasticsearch Utilities:

1. Below are some useful references from the official Elastic documentation and community resources:
 1. How to enable security, passwords, HTTPS, etc. → [Set up minimal security for Elasticsearch](#)
 2. How to enable HTTPS for the REST API. → [Configure TLS/SSL on HTTP layer](#)
 3. Details about the elasticsearch-certutil helper script. → [Elasticsearch certutil documentation](#)
 4. Which users and roles are already available. → [Built-in users and roles](#)

Secure Connections between Elasticsearch Nodes

1. How to secure inter-node communication. → [Configure TLS/SSL on Transport layer](#)
2. Edit the elasticsearch.yml according to the configuration. For manual configuration example:

```
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.keystore.path: /path/to/es-server.jks
xpack.security.transport.ssl.keystore.password: password
xpack.security.transport.ssl.truststore.path: /path/to/es-server.jks
xpack.security.transport.ssl.truststore.password: password
xpack.security.transport.ssl.verification_mode: certificate
```

Cassandra Authentication

Interaction Recording Web Services supports Cassandra authentication, which validates incoming user connections to the Cassandra database. Implementing Cassandra authentication requires you to perform configuration in both Cassandra and Interaction Recording Web Services.

Configure Cassandra Authentication

The user account and password required for authentication are managed inside the **cassandra.yaml** file. Configure Cassandra authentication according to the [Cassandra 2.2 documentation](#).

Interaction Recording Web Services Configuration

To support Cassandra authentication, configure the appropriate credentials in the **cassandraCluster** section of the **application.yaml** file:

```
cassandraCluster:
  thrift_port: 9160
  jmx_port: 7199
  ...
  userName: <superuser name>
  password: <superuser password>
  ...
```

Important

- If the **userName** and **password** are not configured, RWS will connect to Cassandra anonymously.
- To encrypt the password for added security, see [Password Encryption](#).

Password Encryption

For added security, consider encrypting your passwords in the **application.yaml** file by using the following procedure:

1. Run the RWS application with the **--encrypt** parameter followed by the password you need to encrypt. For example, if the password is "ops":

```
$ java -jar gir.jar --encrypt ops
CRYPT:an03xPrxLAu9p==
```

RWS will encrypt and print the password. The server will not actually start.

2. Copy the printed encrypted password and paste into the **application.yaml** file. For example:

```
webDAVJksPassword: CRYPT:an03xPrxLAu9p==
```

The server only decrypts passwords that start with the **CRYPT:** prefix. Passwords without the **CRYPT:** prefix are considered plain text and remain unmodified.

CSRF Protection

Interaction Recording Web Services provides protection against Cross Site Request Forgery (CSRF) attacks. For general information and background on CSRF, see the [OWASP CSRF Prevention Cheat Sheet](#).

Important

If CSRF protection is enabled, then the label/tagging and deletion prevention functionality cannot be used in SpeechMiner, as SpeechMiner does not support CSRF.

To set up Cross Site Request Forgery protection, set the following options in the **serverSettings** section of the **application.yaml** file on each of your Interaction Recording Web Services nodes:

- **enableCsrfProtection**—determines whether CSRF protection is enabled on the Web Services node.
- **crossOriginSettings**—specifies the configuration for cross-origin resource sharing in Interaction Recording Web Services. Make sure this option has the **exposedHeaders** setting with a value that includes X-CSRF-HEADER,X-CSRF-TOKEN.

For example, your configuration might look like this:

```
enableCsrfProtection: true
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-Type,Accept,
Origin,Cookie,authorization,ssid,surl,ContactCenterId"
  allowCredentials: true
  exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

For more information about CSRF protection in the Interaction Recording Web Services API, see [Cross Site Request Forgery Protection](#).

CORS Filter

Interaction Recording Web Services supports Cross-Origin Resource Sharing (CORS) filter, which allows applications to request resources from another domain. For general information and background on CORS, see [Cross-Origin Resource Sharing](#).

To set up Cross-Origin Resource Sharing, make sure you set the **crossOriginSettings** option in the **serverSettings** section of the **application.yaml** file on each of your Interaction Recording Web Services nodes . It specifies the configuration for cross-origin resource sharing in Interaction Recording Web Services. Make sure this option has the **exposedHeaders** setting with a value that includes X-CSRF-HEADER,X-CSRF-TOKEN.

For example, your configuration might look like this:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,X-CSRF-TOKEN"
  allowCredentials: true
  exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

For more information about CORS in the Interaction Recording Web Services API, see [Cross-Origin Resource Sharing](#).

Interaction Recording Web Services Authentication Flow

Interaction Recording Web Services provides authentication in the following sequence:

1. Configuration Server Authentication

- If a request contains a basic authentication header and Configuration Server authentication is enabled for this contact center, Configuration Server authentication is applied.
 - If successful, user is authenticated and execution flow proceeds to the authorization stage.
 - If authentication headers are not present, Configuration Server authentication is disabled, or authentication fails, execution flow proceeds to the next step.

2. Interaction Recording Web Services Authentication

- If a request contains a basic authentication header and Configuration Server authentication is not enabled for this contact center, Interaction Recording Web Services authentication is applied.
 - If successful, user is authenticated and execution flow proceeds to the authorization stage.
 - If authentication headers are not present or authentication fails, execution flow proceeds to the next step.

Next Step

- [Starting and Testing](#)

Starting and Testing

Once you've installed and configured Interaction Recording Web Services, you're ready to start the individual nodes and confirm the service is working.

Starting the Interaction Recording Web Services Nodes

Complete the following steps for each Interaction Recording Web Services node, starting with the **syncNode**.

Important

Verify that the SR Service is active before an agent attempts to log into Workspace Web Edition (WWE).

To create the **ops** user and credentials in Cassandra and to enable the features in the Interaction Recording Web Services node, set the following parameters to true during the first Interaction Recording Web Services startup in the **application.yaml** file:

```
updateOnStartup  
opsCredentials: true  
features: true
```

RHEL 8/9

Start the RWS Service by entering the following command:
`sudo systemctl start gir`

Important

After Interaction Recording Web Services is started, you must change both options to false for production:

```
updateOnStartup  
opsCredentials: false  
features: false
```

Testing Interaction Recording Web Services

Complete the steps below to verify each Interaction Recording Web Services node is up and running.

Start

1. Type the following URL into a web browser:

`http://ws_host:ws_port/api/v2/diagnostics/version`

- *ws_host*—The host name or IP address for the Interaction Recording Web Services node.
- *ws_port*—The port for the Interaction Recording Web Services node.

For example, the URL might be `http://192.0.2.20:8080/api/v2/diagnostics/version`

If the request is successful, the version is printed in the browser:

```
{"statusCode":0,"version":"8.5.200.96"}
```

End

Next Step

- [Configure your required features](#)

Configuring Features

Review the sections below for more information about how to configure Interaction Recording Web Services to use the specified features.

Configuration for Voice Recordings

Interaction Recording Web Services requires a specific configuration for GIR **call** recordings to work correctly. The following sections describe how to configure Interaction Recording Web Services for call recordings.

Configuring the Interaction Recording Web Services Parameters

1. To support call recordings, it's important that you update the following settings in the `serverSettings` section of the **application.yaml** file:
 - `undocumentedExternalApiUrl`
 - `createCallRecordingCF`
 - `crClusterName`
 - `crRegion`
 - `cryptoSecurityKey`
 - `webDAVMaxConnection`
 - `webDAVMaxTotalConnection`
 - `nodePath`
 - `recordingSettings`, in particular **`recordCryptoServerDecryptMaxConnection`**, **`recordCryptoServerDecryptMaxTotalConnection`** and **`recordCryptoServerDecryptSocketTimeout`**
 - `multiPartResolverMaxUploadSize`
 - `multiPartResolverMaxInMemorySize`
 - `backgroundScheduledMediaOperationsSettings`, in particular **`enableBackgroundScheduledMediaOperations`** and **`defaultBackupExportURI`**
2. Determine the contact center ID for Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
  curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
  Recording Web Services port>/api/v2/ops/contact-centers; echo
}
```

Interaction Recording Web Services returns the following output:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
```

```
Recording Web Services port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>"}}}
```

- Using a text editor, create a new file called `add_voice_features` with the following content:

```
{
  "uris":[
    "/api/api-voice-recording",
    "/api/api-supervisor-recording",
    "schema-elasticsearch-v2-call-recording"
  ]
}
```

- Execute the following command:

```
{
curl -u ops:ops -X POST -d @add_voice_features
http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/features
--header "Content-Type: application/json"; echo
}
```

Configuring the Storage Credentials for Interaction Recording Web Services

Enable Voice Recording

Start

- Determine the contact center ID on Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services Port>/api/v2/ops/contact-centers; echo
}
```

The following output is returned:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services Port>/api/v2/ops/contact-centers/<contact center ID (in hex
format)>"}}
```

Important

Use the <contact center ID (in hex format)> in all subsequent commands.

- Using a text editor, create a new file called `create_table` with the following content:

```
{
  "operationName":"createCRCF"
}
```

Important

You do not need to create the table manually when the **createCallRecording** option is set to **true** in the **application.yaml** file. The table will be automatically created by Interaction Recording Web Services (RWS).

3. Execute the following command:

```
{
  curl -u ops:ops -X POST -d @create_table http://<Interaction Recording Web Services
  Server>:<Interaction Recording Web Services Port>/api/v2/ops/
  contact-centers/<contact center ID (in hex format)>/recordings
  --header "Content-Type: application/json"; echo
}
```

End

Enable Storage

Start

1. Using a text editor, create a new file called `recording_settings` with the following content:

```
{
  "store": [
    {
      "webDAV": {
        "userName": "user1",
        "password": "password1",
        "uri": "http://apache1/recordings"
      }
    },
    {
      "webDAV": {
        "userName": "user2",
        "password": "password2",
        "uri": "http://apache2/recordings"
      }
    }
  ]
}
```

Important

The URI in `recording_settings` is case sensitive and must match the URI in the IVR Profile. For example:

```
"uri": "http://GENESYSREC1/recordings"
is not the same as
"uri": "http://genesysrec1/recordings"
```

2. Execute the following command:

```
{
```



```
curl -u ops:ops -X POST -d @recording_settings
  http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/settings/call-
recordings
--header "Content-Type: application/json"; echo
}
```

End

Configuring the Call Recording Audit Log

Interaction Recording Web Services provides an audit log for the following recording operations:

- Playback of the recording media file
- Deletion of the recording file

Complete the steps below to configure the audit log:

Start

1. Stop Interaction Recording Web Services using the following command:

```
sudo service gir stop
```
2. Edit the **GWS_HOME/etc/logback.xml** file and update the configuration to include INFO level messaging. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Example LOGBACK Configuration File
  http://logback.qos.ch/manual/configuration.html
-->
<configuration scan="true">
  <appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <filter class="ch.qos.logback.classic.filter.LevelFilter">
      <level>INFO</level>
      <onMatch>ACCEPT</onMatch>
      <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY for
printing only INFO-->
    </filter>
    <file>${jetty.logs}/recording.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <maxHistory>720</maxHistory><!-- 1 Month -->
    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}] [%X{req.userAgent}]
[%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
    </encoder>
  </appender>
  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${jetty.logs}/cloud.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <!-- hourly rollover -->
      <fileNamePattern>${jetty.logs}/cloud-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <!-- keep 5 days' worth of history -->
      <maxHistory>120</maxHistory>
    </rollingPolicy>
  </appender>
</configuration>
```

```

    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} %-5level [%X{principal.name}]
[%X{session}] [%X{contactCenter}] [%thread] %X{req.requestURI} %X{req.queryString}
%logger{36} %msg%n</pattern>
    </encoder>
  </appender>
  <logger name="com.<domain>.cloud.v2.api.controllers.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>.cloud.v2.api.tasks.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>" level="WARN" />
  <logger name="com.<domain>.cloud" level="DEBUG" />
  <logger name="com.<domain>.cloud.rtreporting" level="WARN" />
  <logger name="com.<domain>.salesforce.security" level="INFO" />

  <root level="WARN">
    <appender-ref ref="FILE" />
  </root>
</configuration>

```

3. For MLM, create a **RECORDING** appender if it does not exist. For example:

```

<appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <filter class="ch.qos.logback.classic.filter.LevelFilter">
    <level>INFO</level>
    <onMatch>ACCEPT</onMatch>
    <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY for
printing only INFO-->
  </filter>
  <file>${jetty.logs}/recording.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd}.gz</fileNamePattern>
    <maxHistory>720</maxHistory><!-- 1 Month -->
  </rollingPolicy>
  <encoder>
    <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}] [%X{req.userAgent}]
[%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
  </encoder>
</appender>

```

4. Add the following loggers for the **RECORDING** appender:

```

<logger name="com.genesyslab.cloud.v2.api.controllers.callrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.controllers.screenrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.callrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.interactionrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.screenrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.settings">

```

```

    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.genesyslab.cloud.v2.media.scheduler">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.genesyslab.cloud.v2.media.task">
    <appender-ref ref="RECORDING" />
  </logger>

```

For more information about Logback, see [Logback configuration](#).

5. Start GIR using the following command:

```
sudo service gir start
```
6. Review the audit log. Open the <LOG_PATH>/recording.log file, where <LOG_PATH> is the path parameter for the logging section in your application.yaml. By default, this is /var/log/jetty9. The following example shows that two recordings are requested for playback and deletion:

```

10/28/2013 15:46:03.203 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/
537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

```

```

10/28/2013 15:46:03.341 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] failed

```

```

10/28/2013 15:46:10.946 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

```

```

10/28/2013 15:46:11.033 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] succeed

```

```

10/28/2013 15:46:52.179 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid0 Delete metadata and media-files of call-recording is requested. contact-center
[46284f2f-d615-4329-957a-f5341edfd5d7], call-recording [recid0]

```

```

10/28/2013 15:46:52.216 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/

```

```

recid0 Delete metadata and media-files of call-recording failed. contact-center
[46284f2f-d615-4329-957a-f5341edfd5d7], call-recording [recid0]

10/28/2013 15:46:56.253 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete metadata of call-recording is requested. contact-center [46284f2f-
d615-4329-957a-f5341edfd5d7], call-recording [recid1]

10/28/2013 15:46:56.420 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete metadata of call-recording succeeded. contact-center [46284f2f-
d615-4329-957a-f5341edfd5d7], call-recording [recid1]

```

End

Configuring the API Thread Pool

Interaction Recording Web Services provides properties for the Call Recording API thread pool by configuring the **hystrix.properties** file.

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name]. execution.isolation.thread. timeoutInMilliseconds	N/A	The hystrix timeout. The default value is set to 6000.
hystrix.threadpool.[API Pool Name] .coreSize	N/A	The thread pool size. The default value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateCallRecordingApiTaskV2	ApiCreatePool	Create call recording.
DeleteCallRecordingApiTaskV2	ApiDeletePool	Delete call recording.
GetCallRecordingApiTaskV2	ApiGetPool	Get call recording metadata.
GetCallRecordingCFInfoApiTaskV2	ApiGetPool	Get call recording CF Information.
GetCallRecordingMediaApiTaskV2	ApiGetPool	Streaming call recording media.
QueryCallRecordingApiTaskV2	ApiQueryPool	Query call recording metadata.

For more information about the Call Recording API, see the [Genesys Interaction Recording API Reference](#).

Configuration for Screen Recordings

As with call recordings, Interaction Recording Web Services requires a specific configuration for GIR **screen** recordings to work correctly. The following sections describe how to configure Interaction Recording Web Services for screen recordings.

Configuring the Interaction Recording Web Services Parameters

Complete the steps below to support screen recordings:

Start

1. For **Cassandra 2.2**, Update the following settings in the serverSettings section of the **application.yaml** file. Your configuration should look something like this:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: <Interaction Recording Web Services Servers>,<SpeechMiner Web
Servers>
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,Range"
  allowCredentials: true
screenRecordingSettings:
  screenRecordingEServicesEnabled: true
  screenRecordingVoiceEnabled: true
screenRecordingConnectionReportingSettings:
  reportingEnabled: true
  createReportingCF: true
multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 67108864
```

For **Cassandra 4.1**, Update the following settings in the serverSettings section of the **application.yaml** file. Your configuration should look something like this:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: <Interaction Recording Web Services Servers>,<SpeechMiner Web
Servers>
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,Range"
  allowCredentials: true
screenRecordingSettings:
  screenRecordingEServicesEnabled: true
  screenRecordingVoiceEnabled: true
screenRecordingConnectionReportingSettings:
  reportingEnabled: false
  createReportingCF: false
multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 67108864
```

Make the following changes to the example above:

- Change <Interaction Recording Web Services Servers> and <SpeechMiner Web Servers> to the HTTP/HTTPS addresses of the Interaction Recording Web Services instances and SpeechMiner Web Servers.

- **multiPartResolverMaxUploadSize** controls the maximum allowed size (in bytes) for a screen recording video file that can be uploaded to Interaction Recording Web Services. This parameter should be aligned with **maxDurationMinutes**, so if you change its value, ensure that you also consider the **maxDurationMinutes** value specified within the *Advanced Configuration for the Screen Recording Service* section in the [Deploying the Screen Recording Service - Advanced Configuration](#) page. The maximum size of a file that can be uploaded by the Screen Recording Service must be less than or equal to the **multiPartResolverMaxUploadSize**.

2. Determine the contact center ID on Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
  curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
  Recording Web Services port>/api/v2/ops/contact-centers; echo
}
```

Interaction Recording Web Services returns the following output:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>"]}
```

3. For **Cassandra 2.2**, Using a text editor, create a new file called `add_screen_features` with the following content:

```
{
  "uris":[
    "/api/api-voice-screenrecording",
    "/api/api-multimedia-screenrecording",
    "/api/api-screenrecording-connection-reporting",
    "schema-elasticsearch-v2-screen-recording"
  ]
}
```

For **Cassandra 4.1**, Using a text editor, create a new file called `add_screen_features` with the following content:

```
{
  "uris":[
    "/api/api-voice-screenrecording",
    "/api/api-multimedia-screenrecording",
    "schema-elasticsearch-v2-screen-recording"
  ]
}
```

4. Execute the following command:

```
{
  curl -u ops:ops -X POST -d @add_screen_features
  http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
  Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/features
  --header "Content-Type: application/json"; echo
}
```

- Use the **api-voice-screenrecording** parameter for voice interactions, and use the **api-multimedia-screenrecording** parameter for non-voice interactions.
- Use the **api-screenrecording-connection-reporting** parameter to enable the collection of information about Screen Recording Services client connections for the contact center. **This feature is not applicable for Cassandra 4.1.**
- If you wish to direct the SpeechMiner UI to Interaction Recording Web Services instead of Recording Crypto Server for decryption of screen recordings, add the **api-recordings-decryption-proxying** parameter to the list of features enabled for the contact center above. Note that this requires

additional configuration.

- Using a text editor, create a new file called `create_stats_table`, with the following content:**This step is not applicable for Cassandra 4.1.**

```
{
  "operationName": "CreateReportingCFs"
}
```

- Execute the following command:**This step is not applicable for Cassandra 4.1.**

```
{
  curl -u ops:ops -X POST -d @create_stats_table http://<Interaction Recording Web
  Services Server>:<Interaction Recording Web Services Port>/api/v2/ops/contact-
  centers/<contact center ID (in hex format)>/screen-recording-connections --header
  "Content-Type: application/json"; echo
}
```

End

Configuring the Storage Credentials for Interaction Recording Web Services

Complete the steps below to configure storage credentials for Interaction Recording Web Services.

Start

- Determine the contact center ID on Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
  curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
  Recording Web Services port>/api/v2/ops/contact-centers; echo
}
```

Interaction Recording Web Services returns the following output:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>"]}
```

Important

Use the `<contact center ID (in hex format)>` construction in all subsequent commands.

- Using a text editor, create a new file called `create_table`, with the following content:

```
{
  "operationName": "createCRCF"
}
```

- Execute the following command:

```
{
```

```
curl -u ops:ops -X POST -d @create_table http:// <Interaction Recording Web Services
Server>:<Interaction Recording Web Services Port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>/screen-recordings
--header "Content-Type: application/json"; echo
}
```

4. Enable storage for a single or multiple locations:

Important

Within the storage settings, the same location can be specified multiple times if you have inactive ("active": false) settings specified as well as "active": true. However, you must ensure that for a specific location, only one value has "active": true set. For additional information about storage settings, refer to [Interaction Recording Web Services \(Web Services\) Group Settings](#). See the **Property Descriptions** section for details about the supported property values.

- For a **single** location:

a. Using a text editor, create the create_single_location file:

```
{
  "name": "storage",
  "location": "/",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "<webdav user>",
        "password": "<webdav password>",
        "storagePath": "<webdav uri>"
      }
    }
  ]
}
```

Important

Replace <webdav user>, <webdav password>, <webdav uri> with the appropriate values.

b. Execute the following command:

```
{
curl -u ops:ops -X POST -d @create_single_location http:// <Interaction Recording Web
Services Server>:<Interaction Recording Web Services Port>/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
}
```

- For **multiple** locations:

a. Using a text editor, create the create_first_location file:

```
{
  "name": "storage",
```



```

    "location": "<node_location>",
    "value": [
      {
        "storageType": "webDAV",
        "active": true,
        "credential": {
          "userName": "<webdav user>",
          "password": "<webdav password>",
          "storagePath": "<webdav uri>"
        }
      }
    ]
  }
}

```

- b. Execute the following command:

```

{
curl -u ops:ops -X POST -d @create_first_location http://<Interaction Recording
Web Services Server>:<Interaction Recording Web Services Port>/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
}

```

Important

Replace <node_location>, <webdav user>, <webdav password>, <webdav uri> with the appropriate values. The values for the <node_location> are similar to the **nodePath** settings in the **application.yaml** file, but allow a hierarchical representation. For example, an Interaction Recording Web Services node uses a storage setting with a location of "/US" in the nodePath set to "/US/AK" or "/US/HI".

For more information on hierarchical location setting, go [\[\[1\]\]](#)

- c. Repeat steps a and b for each location required.

End

For more information on the properties of this settings group, see [Interaction Recording Web Services Settings Groups](#).

Configuring the API Thread Pool

Interaction Recording Web Services provides properties for the Call Recording API thread pool by configuring the **hystrix.properties** file.

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name]. execution.isolation.thread. timeoutInMilliseconds	N/A	The hystrix timeout. The default value is set to 6000.
hystrix.threadpool.[API Pool]	N/A	The thread pool size. The default

Property/API Name	Thread Pool Name	Description
Name] .coreSize		value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateScreenRecordingApiTaskV2	ApiUploadPool	Create screen recording
DeleteScreenRecordingMediaApiTaskV2	ApiDeletePool	Delete screen recording
GetScreenRecordingApiTaskV2	ApiGetPool	Get screen recording metadata
GetScreenRecordingMediaApiTaskV2	ApiStreamPool	Stream screen recording media
QueryScreenRecordingApiTaskV2	ApiQueryPool	Query screen recording metadata

For more information about the Call Recording API, see the [Genesys Interaction Recording API Reference](#).

Configuration Options

You can set the configuration options below in the corresponding sections of the **application.yaml** file on your Interaction Recording Web Services nodes. For details, see [Configuring Interaction Recording Web Services](#).

Important

When editing the **application.yaml** file, the values for the configuration options that are strings must be enclosed in double quotation marks in certain cases. Specifically:

- For string options only, the values YES, NO, ON, OFF, TRUE, FALSE (in upper or lower case) must be quoted.
- If the option is a boolean (true/false) option, then any of the values in the previous bullet can be used without quotes.
- Values that look like numbers but are treated as strings (for example; PINs, phone numbers, encryption keys), that begin with leading zeroes must be quoted.
- Avoid placing leading zeroes on numeric options; doing so will cause your option to be interpreted as an octal value.

For example, specifying `crRegion: NO` (indicating Norway) will be interpreted as `crRegion: FALSE`. Instead, this must be specified using double quotation marks `crRegion: "NO"`.

logging

Settings in this section are listed under **logging**.

config

Default Value: `logback.xml`

Valid Values: A valid path

Mandatory: No

Specifies the path to the **logback.xml** file. You created this file (or Interaction Recording Web Services created it for you) as part of [Deploying the Web Application](#).

file

Default Value: `cloud.log`

Valid Values: A valid file name

Mandatory: No

Specifies the name of the log file. This value is stored in `${LOG_FILE}` which may be used in **logback.xml**.

path

Default Value: /var/log/jetty9

Valid Values: A valid path

Mandatory: No

Specifies the path to the log file. This value is stored in `${LOG_PATH}` which may be used in **logback.xml**.

jetty

Settings in this section are listed under **jetty**.

host

Default Value: 0.0.0.0

Valid Values: A host name or IP address

Mandatory: No

Specifies the host name or IP address of the Jetty host. This value should be the same as GWS_HOST you defined as part of [Deploying the Web Application](#).

port

Default Value: 8080

Valid Values: A valid port

Mandatory: No

Specifies the port of the Jetty host. This value should be the same as GWS_PORT you defined as part of [Deploying the Web Application](#).

idleTimeout

Default Value: 30000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum idle time, in milliseconds, for a connection.

soLingerTime

Default Value: -1

Valid Values: An integer greater than 0, or -1 to disable

Mandatory: No

Specifies the socket linger time.

sessionMaxInactiveInterval

Default Value: 1800

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the period, in seconds, after which a session is deemed idle and saved to session memory.

enableWorkerName

Default Value: true

Valid Values: true, false

Mandatory: No

Specifies whether to add the WorkerName parameter into the sessionCookieName cookie.

enableRequestLog

Default Value: true

Valid Values: true, false

Mandatory: No

Enables request logging. If you set the value to true, you must also set values for the [requestLog](#) option.

requestLog

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
filename	No	yyyy_mm_dd.cloud-request.log	Specifies the log file name format.
filenameDateFormat	No	yyyy_MM_dd	Specifies the log file name date format.
logTimeZone	No	GMT	Specifies the timestamp time zone used in the log.
retainDays	No	90	Specifies the time interval, in days, for which Jetty should retain logs.
append	No	true	Specifies whether Jetty appends to the request log file or starts a new file.
extended	No	true	Specifies whether Jetty logs extended data.
logCookies	No	true	Specifies whether Jetty logs request cookies.
logLatency	No	true	Specifies whether Jetty logs the request latency.
preferProxiedForAddress	No	true	Specifies whether Jetty logs IP address or the IP address from the X-Forwarded-For request header.

Mandatory: No

Specifies how Jetty should handle request logging. For example:

```
enableRequestLog: true
requestLog:
  filename: yyyy_mm_dd.cloud-request.log
  filenameDateFormat: yyyy_MM_dd
  logTimeZone: GMT
  retainDays: 90
  append: true
  extended: true
  logCookies: false
  logLatency: true
  preferProxiedForAddress: true
```

These options only take effect if **enableRequestLog** is set to true.

enableSsl

Default Value: false

Valid Values: true, false

Mandatory: No

Enables Secure Sockets Layer support. If you set the value to true, you must also set values for the **ssl** option.

ssl

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
port	No	443	The SSL port. This option is the equivalent of the Jetty "https.port" variable.
securePort	No	8443	The port to which integral or confidential security constraints are redirected. This option is the equivalent of the Jetty "jetty.secure.port" variable.
idleTimeout	No	30000	The maximum idle time, in milliseconds, for a connection.
soLingerTime	No	-1	The socket linger time. A value of -1 disables this option.
keyStorePath	No	None	The keystore path.
keyStorePassword	No	None	The keystore password.
keyManagerPassword	No	None	The key manager password.

Name	Mandatory	Default Value	Description
keyStoreProvider	No	None	The keystore provider.
keyStoreType	No	JKS	The keystore type.
trustStorePath	No	None	The truststore path.
trustStorePassword	No	None	The truststore password.
trustStoreProvider	No	None	The truststore provider.
trustStoreType	No	JKS	The truststore type.
needClientAuth	No	None	Set this option to true if SSL needs client authentication.
wantClientAuth	No	None	Set this option to true if SSL wants client authentication.
certAlias	No	None	The alias of the SSL certificate for the connector.
validateCerts	No	None	Set this option to true if the SSL certificate has to be validated.
validatePeerCerts	No	None	Set this option to true if SSL certificates of the peer have to be validated.
trustAll	No	None	Set this option to true if all certificates should be trusted if there is no keystore or truststore.
renegotiationAllowed	No	None	Set this option to true if TLS renegotiation is allowed.
excludeCipherSuites	No	None	Specifies the array of cipher suite names to exclude from enabled cipher suites.
includeCipherSuites	No	None	Specifies the array of cipher suite names to include in enabled cipher suites.
endpointIdentificationAlgorithm	No	None	Specifies the endpoint identification algorithm. Set this option to "HTTPS" to enable hostname verification.
includeProtocols	No	None	The array of protocol names (protocol versions) to include for use on this engine.

Name	Mandatory	Default Value	Description
excludeProtocols	No	None	The array of protocol names (protocol versions) to exclude from use on this engine.

Mandatory: No

Specifies how Jetty should handle support for Secure Sockets Layer. For example:

```
enableSsl: true
ssl:
  port: 443
  securePort: 8443
  idleTimeout: 30000
  soLingerTime: -1
```

These options only take effect if **enableSsl** is set to true.

httpOnly

Default Value: true

Valid Values: true, false

Mandatory: No

If true, it sets an HTTP-only flag for session cookies.

secure

Default Value: false

Valid Values: true, false

Mandatory: No

If true, it sets a secure cookie flag for session cookies.

sessionCookieName

Default Value: GIRJSESSIONID

Valid Values: Any string which can be used as a cookie name as per [RFC 6265](#)

Mandatory: No

Defines the name of the session cookie used by Interaction Recording Web Services.

sessionCookieName can only contain the following characters:

- Letters: a-z or A-Z
- Digits: 0-9
- Hyphen (-)
- Underscore (_)

cassandraCluster

Settings in this section are listed under **cassandraCluster** for Cassandra 4.1.

native_transport_port

Default Value: 9042

Valid Values: A valid port

Mandatory: No

Specifies the port used by CQL. This is required if not using the default port.

dataCenterName

Default Value: datacenter1

Valid Values: Name of the primary datacenter used by the new CQL driver

Mandatory: No

Name of the primary datacenter used by the new CQL driver. The required value can be found by running `nodetool status` on the Cassandra instance. This is required if not using the default datacenter name.

backupDataCenterName

Default Value: datacenter1

Valid Values: Name of the secondary datacenter used by the new CQL driver if there is a secondary datacenter

Mandatory: No

Name of the secondary datacenter used by the new CQL driver if there is a secondary datacenter. The required value can be found by running `nodetool status` on the Cassandra instance. This is required if using a Multi datacenter Cassandra. and not using the default secondary datacenter name.

switchOverUnHealthyNodes

Default Value: -1

Valid Values: The number of nodes in the primary Cassandra datacenter that need to fail before RWS will attempt to switch to the backup datacenter.

Mandatory: No

The number of nodes in the primary Cassandra datacenter that need to fail before RWS will attempt to switch to the backup datacenter. The default value is -1, meaning that no attempt to switchover will be made. This is required only if using a Multi datacenter Cassandra.

jmx_port

Default Value: 7199

Valid Values: A valid port

Mandatory: No

Specifies the port Cassandra uses for Java Manage Extension (JMX).

keyspace

Default Value: sipfs

Valid Values: A valid keyspace name

Mandatory: Yes

Specifies the name of the Cassandra keyspace. This name should be the same as the keyspace name you set while [Creating the Cassandra keyspace](#). If you used the keyspace creation scripts that come with Interaction Recording Web Services, then you can leave this value as sipfs.

nodes

Default Value: None

Valid Values: A comma-separated list of IP addresses or host names

Mandatory: Yes

Specifies the Cassandra node IP addresses or host names.

backup_nodes

Default Value: None

Valid Values: A comma-separated list of IP addresses or host names

Mandatory: No

Specifies the backup Cassandra node IP addresses or host names. This option is intended for deployments that have two separate Cassandra data centers — Interaction Recording Web Services switches from primary to backup if the primary nodes are unavailable. If your deployment is small with only one data center, you can ignore this option.

replication_factor

Default Value: None

Valid Values: An integer less than or equal to the number of nodes in the cluster

Mandatory: Yes

Specifies a replication factor appropriate for your Cassandra topology. This value must be the same as the replication factor you set in Step 2 of the [Creating the Cassandra keyspace](#) procedure.

read_consistency_level

Default Value: None

Valid Values: CL_ONE, CL_QUORUM, CL_LOCAL_QUORUM

Mandatory: Yes

Specifies the read consistency level appropriate for your Cassandra topology:

Development (1 Cassandra node)	Single Datacenter (1 datacenter with a minimum of three Cassandra nodes)	Two Datacenters (datacenters with a minimum of three Cassandra nodes in each datacenter)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

write_consistency_level

Default Value: None

Valid Values: CL_ONE, CL_QUORUM, CL_LOCAL_QUORUM

Mandatory: Yes

Specifies the write consistency level appropriate for your Cassandra topology:

Development (1 Cassandra node)	Single Datacenter (1 datacenter with a minimum of three Cassandra nodes)	Two Datacenters (datacenters with a minimum of three Cassandra nodes in each datacenter)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

max_conns_per_host

Default Value: 16

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of connections to allocate for a single host's pool.

max_cons

Default Value: 48

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of connections in the pool.

max_pending_conns_per_host

Default Value: 80

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of pending connection attempts per host.

max_blocked_threads_per_host

Default Value: 160

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of blocked clients for a host.

cassandraVersion

Default Value: 4.1.4

Valid Values: Cassandra version

Mandatory: No

Specifies the Cassandra version for your Interaction Recording Web Services deployment.

useSSL

Default Value: None

Valid Values: true, false

Mandatory: No

Specifies whether Cassandra should use Secure Sockets Layer (SSL). This option is only valid for Cassandra versions 1.2.x and higher.

truststore

Default Value: None

Valid Values: A valid path.

Mandatory: No

Specifies the path to the truststore.

truststorePassword

Default Value: None

Valid Values: A valid password.

Mandatory: No

Specifies the password for the truststore.

userName

Default Value: None

Valid Values: A valid Cassandra username.

Mandatory: No

Specifies the username if Cassandra is configured to use authentication.

password

Default Value: None

Valid Values: A valid Cassandra password.

Mandatory: No

Specifies the password if Cassandra is configured to use authentication.

serverSettings

Settings in this section are listed under **serverSettings**.

URLs

externalApiUrlV2

Default Value: None

Valid Values: A public schema-based URL ending with /api/v2.

Mandatory: Yes

Specifies the prefix used for resources in the public API. In a development environment, the host and port should be set to the host name or IP address of the Interaction Recording Web Services node. In a production environment, the host and port should be set to the host name or IP address of the load balancer in a production environment. For example, `https://192.0.2.20/api/v2`.

`internalApiUrlV2`

Default Value: None

Valid Values: A public schema-based URL ending with `/internal-api`.

Mandatory: Yes

Specifies the prefix used for internal resources. In a development environment, the host and port should be set to the host name or IP address of the Interaction Recording Web Services node. In a production environment, the host and port should be set to the host name or IP address of the load balancer in a production environment. For example, `http://192.0.2.20/internal-api`.

`undocumentedExternalApiUrl`

Default Value: None

Valid Values: A public schema-based URL ending with `/internal-api`.

Mandatory: Yes

Specifies the reachable Interaction Recording Web Services server address for the SpeechMiner UI and the Screen Recording Service. For example, `http://192.0.2.20:8090/internal-api`

Paths

`pathPrefix`

Default Value:

Valid Values: A valid prefix

Mandatory: No

Specifies a prefix that Interaction Recording Web Services adds to the relative URIs it includes in responses. For example, if you set **pathPrefix** to `/api/v2` and make the following request:

```
GET http://localhost:8080/api/v2/devices
```

Interaction Recording Web Services returns the following response:

```
{
  "statusCode":0,
  "paths":[
    "/api/v2/devices/971ed91d-82bf-490b-94d2-02d240165764",
    "/api/v2/devices/a3f9e854-54d8-4260-bea3-d6e450ee7df0"
  ],
  "uris":[
    "http://localhost:8080/api/v2/devices/7c7ab1f7-e596-41bc-9ff4-4a12c489865f",
    "http://localhost:8080/api/v2/devices/a3f9e854-54d8-4260-bea3-d6e450ee7df0"
  ]
}
```

Notice that paths includes relative URIs with the `/api/v2` prefix.

`internalPathPrefix`

Default Value: Empty

Valid Values: A valid prefix

Mandatory: No

Specifies a prefix that Interaction Recording Web Services adds to the relative URIs that it includes in responses to internal APIs. See `pathPrefix` for details.

General

`temporaryAuthenticationTokenTTL`

Default Value: 300

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the time to live, in seconds, for the temporary authentication token.

`enableCsrfProtection`

Default Value: false

Valid Values: true, false

Mandatory: No

Enables cross site request forgery protection. If you set the value to `true`, make sure you use the default values for **exposedHeaders** in the `crossOriginSettings` option. If you have already updated the **exposedHeaders**, just make sure the values include the defaults.

Important

If CSRF protection is enabled, then the label/tagging and deletion prevention functionality cannot be used in SpeechMiner, as SpeechMiner does not support CSRF.

Timeouts

`activationTimeout`

Default Value: 12000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for connecting to any Genesys server (except Configuration Server). This may include several individual attempts if the initial attempt to connect is unsuccessful.

Important

The activation timeout for Configuration Server is specified with the **configServerActivationTimeout** option.

configServerActivationTimeout

Default Value: 35000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for connecting to Configuration Server. This may include several individual attempts if the initial attempt to connect is unsuccessful.

configServerConnectionTimeout

Default Value: 15000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for an individual connection attempt to Configuration Server.

connectionTimeout

Default Value: 4000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for an individual connection attempt to any Genesys server (except Configuration Server).

Important

The connection timeout for Configuration Server is specified with the **configServerConnectionTimeout** option.

inactiveUserTimeout

Default Value: 60

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the interval, in seconds, at which the inactive user cleanup process is run by the server. This process is run to invalidate HTTP sessions for users who have been deleted or whose user roles have changed.

reconnectAttempts

Default Value: 1

Valid Values: An integer greater than 0.

Mandatory: Yes

Specifies the number of attempts Interaction Recording Web Services makes to connect to any Genesys server before attempting to connect to the backup.

reconnectTimeout

Default Value: 10000

Valid Values: An integer greater than 0.

Mandatory: Yes

Specifies the timeout, in milliseconds, between the reconnect attempts.

OPS account

opsUserName

Default Value: None

Valid Values: Any alphanumeric value that can include special characters

Mandatory: Yes

Specifies the name of the Interaction Recording Web Services super user. Interaction Recording Web Services creates or updates this user at startup if **opsCredentials** is set to `true` in the **updateOnStartup** section of the **application.yaml** file.

opsUserPassword

Default Value: None

Valid Values: Any alphanumeric value, including special characters

Mandatory: Yes

Specifies the password for the Interaction Recording Web Services super user. Interaction Recording Web Services creates or updates the password for the **ops** user at startup if **opsCredentials** is set to `true` in the **updateOnStartup** section of the **application.yaml** file.

CME credentials

applicationName

Default Value: None

Valid Values: A valid application name

Mandatory: Yes

The name of the Interaction Recording Web Services node application object in Configuration Server. For example, `IRWS_Node`.

applicationType

Default Value: None

Valid Values: A valid application type

Mandatory: Yes

The type of the Interaction Recording Web Services node application object in Configuration Server. This value should be `CFGGenericClient`.

cmeUserName

Default Value: None

Valid Values: A valid Configuration Server user

Mandatory: Yes

The username that the Interaction Recording Web Services server uses to connect to Configuration Server.

Important

Genesys recommends that you use the provided "default" account in Configuration Server. It is possible to use a different account, but you must take care in configuring the user's account permissions. Outside of a lab setting, this is best done in consultation with Genesys.

cmePassword

Default Value: None

Valid Values: A valid password

Mandatory: Yes

The password for the Configuration Server user Interaction Recording Web Services uses to connect to Configuration Server.

syncNode

Default Value: false

Valid Values: true, false

Mandatory: Yes (for sync node), No (all other nodes)

Specifies whether the node is the synchronization node. This node is responsible for importing objects from Configuration Server into Cassandra, subscribing to change notifications with Configuration Server, and processing updates.

Important

In each Interaction Recording Web Services cluster or shared Interaction Recording Web Services and Web Services and Applications cluster, if both are deployed, one node in the cluster must be configured as the synchronization node: `syncNode: true`. All other nodes in the cluster must have `syncNode: false`.

ConfigServer String Encoding

configServerDefaultEncoding

Default Value: windows-1252

Valid Values: A valid java string encoding

Mandatory: No

The configuration server can be installed in one of two modes. One mode uses UTF-8 for encoding strings; the other uses the default character encoding of the machine that the configuration server is installed on. If you are using UTF-8, RWS will communicate using UTF-8 and this parameter is not used. If you are not using UTF-8, this value should be set to the value of the default string encoding of the machine that the configuration server is installed on.

Call Recording

createCallRecordingCF

Default Value: false

Valid Values: true, false

Mandatory: No

Specifies if the column families required for Call Recording will be created when a contact center is created.

crClusterName

Default Value: None

Valid Values: A valid cluster name

Mandatory: Yes

Specifies the name of the cluster to enable search functionality in Elasticsearch. The value must be the same for all Interaction Recording Web Services nodes in the cluster and must match the **cluster.name** parameter configured in **elasticsearch.yml** for each Elasticsearch node. For example, if there are five nodes in the Interaction Recording Web Services cluster, all five nodes must have the same value as in the **crClusterName** option.

crRegion

Default Value: None

Valid Values: String

Mandatory: Yes

Specifies the name of the region where the Interaction Recording Web Services node is located. Ensure that this value is the same on all RWS nodes.

cryptoSecurityKey

Default Value: None

Valid Values: A valid security key

Mandatory: Yes

Specifies the security key used for encryption for call recording settings stored in the database. The value must be the same for all Interaction Recording Web Services nodes in the cluster. For example, if there are five nodes in the Interaction Recording Web Services cluster, all five nodes must have the same value as in the **cryptoSecurityKey** option.

webDAVMaxConnection

Default Value: 50

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of WebDAV client TCP connections to each route. When the number of WebDAV client requests to the same WebDAV server are less than this value, a new TCP connection is established for better performance. Otherwise, the new request is queued until any ongoing request finishes.

webDAVMaxTotalConnection

Default value: 10 * value of [webDAVMaxConnection](#)

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of TCP connections from the Interaction Recording Web Services node to all WebDAV storage.

Multi regional supporting

nodePath

Default Value: None

Valid Values: A location and node ID, separated by a "/" — for example, /US/node1

Mandatory: Yes

Specifies the location and ID of the Interaction Recording Web Services node within the deployment topology. This value must be unique across the deployment. For example, a value of /US/node1 means that the node is located in the US region and has an ID of "node1". The node ID can be the hostname, the IP address, or any other unique identifier.

nodeId

Default Value: None

Valid Values: Any unique identifier, such as the node host name or IP address

Mandatory: No

Specifies the unique identifier for the Interaction Recording Web Services node. Each node in a cluster must have a unique nodeId.

SSL and CA

caCertificate

Default Value: None

Valid Values: Path to a signed certificate or empty

Mandatory: No

Specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if .jks, you can also set [jksPassword](#)). The certificate will be used if the IRWS_Cluster application uses [Transport Layer Security \(TLS\)](#) to connect to the Configuration Server, SIP Server, and Interaction Server. If left empty, or if the parameter is not specified, the certificates returned from the servers will not be validated.

jksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in [caCertificate](#), when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see [Password Encryption](#).

webDAVTrustedCA

Default Value: true

Valid Values: true, false, or a path to a file containing a certificate for a Certificate Authority

Mandatory: No

When using a secure connection to the WebDAV Server, controls whether the certificate is validated, and how. If set to false, the certificate presented by the WebDAV Server will not be validated. If set to true, the certificate presented by the WebDAV Server will be validated by **caCerts** in \$JAVA_HOME/jre/lib/security. Otherwise, specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if it is in .jks format, you can also set [webDAVJksPassword](#)).

webDAVJksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in [webDAVTrustedCA](#) when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see [Password Encryption](#).

rcsTrustedCA

Default Value: true

Valid Values: true, false, or a path to a file containing a certificate for a Certificate Authority

Mandatory: No

When using a secure connection to the Recording Crypto Server, controls whether the certificate is validated, and how. If set to false, the certificate presented by RCS will not be validated. If set to true, the certificate presented by RCS will be validated by **caCerts** in \$JAVA_HOME/jre/lib/security. Otherwise, specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if it is in .jks format, you can also set [rcsJksPassword](#)).

rcsJksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in [rcsTrustedCA](#) when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see [Password Encryption](#).

speechMinerTrustedCA

Default Value: true

Valid Values: true, false, or a path to a file containing a certificate for a Certificate Authority

Mandatory: No

When using a secure connection to the SpeechMiner Interaction Receiver, controls whether the certificate is validated, and how. If set to false, the certificate presented by the SpeechMiner Interaction Receiver will not be validated. If set to true, the certificate presented by the SpeechMiner Interaction Receiver will be validated by **caCerts** in \$JAVA_HOME/jre/lib/security. Otherwise, specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if it is in .jks format, you can also set [speechMinerJksPassword](#)).

speechMinerJksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in [speechMinerTrustedCA](#) when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see [Password Encryption](#).

CORS

crossOriginSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
allowedOrigins	No	None	Specifies a comma-separated list of allowed origins supported by this Interaction Recording Web Services node. For example, http://*.genesys.com , http://*.genesyslab.com
allowedMethods	No	GET,POST,PUT,DELETE,OPTIONS	Specifies a comma-separated list of HTTP methods supported by the server.
allowedHeaders	No	X-Requested-With,Content-Type,Accept,Origin,Cookie,X-CSRF-TOKEN,authorization,ssid,surl,ContactCenterId	Specifies whether to include the Access-Control-Allow-Headers header as part of the response to a pre-flight request. This specifies which header field names can be used during the actual request.
allowCredentials	No	true	Specifies the value of the Access-Control-Allow-Credentials header. This should typically be left at the default value.
corsFilterCacheTimeToLive	No	120	Specifies for how long (in seconds) the cross origin settings are cached before being reloaded.
exposedHeaders	No	X-CSRF-HEADER,X-CSRF-TOKEN	Specifies which custom headers are allowed in

Name	Mandatory	Default Value	Description
			cross-origin HTTP responses. This should typically be left at the default value. If you do modify the value and you enable the enableCsrfProtection option, make sure the value for exposedHeaders includes X-CSRF-HEADER, X-CSRF-TOKEN.

Mandatory: No

Specifies the configuration for cross-origin resource sharing in Interaction Recording Web Services. For example:

```
...
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId"
  allowCredentials: true
  exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

Elasticsearch

elasticSearchSettings

Default Value: None**Valid Values:**

Name	Mandatory	Default Value	Description
retriesOnConflict	No	3	Controls how many times to retry if there is a version conflict when updating a document.
waitToIndexTimeout	No	5000	Specifies the length of time (in milliseconds) that the Interaction Recording Web Services will wait while Elasticsearch is indexing data.
scanReadTimeoutSeconds	No	60	Specifies the length of time that the Interaction Recording Web Services waits for Elasticsearch to return results from setting up a scan and

Name	Mandatory	Default Value	Description
			scroll search request.
countReadTimeoutSecondsNo		60	Specifies the length of time that the Interaction Recording Web Services waits for Elasticsearch to return results from a count search request.
scrollTimeoutSeconds	No	240	Specifies how long Elasticsearch should keep the Search Context alive when handling scan and scroll requests from the Muxer and MLM components. This value must be long enough to process each batch of results. However, it does not need to be long enough to process all data. You can change this value based on the performance results in your environment.
useTransportClient	No	true	Specifies whether Interaction Recording Web Services should use a transport client for Elasticsearch. This is only applicable for Elasticsearch 1.x.
transportClient	Yes, if useTransportClient is true.	Values specified in TransportClientSettings	Specifies the configuration Interaction Recording Web Services should use for the transport client. For details see TransportClientSettings in the next table. This is only applicable for Elasticsearch 1.x.
useRestClient	no	false	Specifies whether Interaction Recording Web Services should use a REST client for Elasticsearch. This is only applicable for Elasticsearch 7.16.3 or higher.
restClient	Yes, if useRestClient is true.	Values specified in RestClientSettings .	Specifies the configuration Interaction Recording Web Services should use for the REST

Name	Mandatory	Default Value	Description
			client. For details, see RestClientSettings in the following tables. This is only applicable for Elasticsearch 7.16.3 or higher.

TransportClientSettings

Name	Mandatory	Default Value	Description
nodes	Yes, if useTransportClient is true.	null	Specifies the list of Elasticsearch nodes the transport client should connect to.
useSniff	no	false	Specifies if the transport client should use sniffing functionality and perform auto-discovery of Elasticsearch nodes in the cluster.
ignoreClusterName	no	false	Specifies if Interaction Recording Web Services should ignore the name of the cluster when connecting to the cluster.
pingTimeout	no	5000	Specifies, in milliseconds, the ping timeout for Elasticsearch nodes.
nodesSamplerInterval	no	5000	Specifies, in milliseconds, how often Interaction Recording Web Services should sample/ping the Elasticsearch nodes listed and connected.

Mandatory: No

Specifies the configuration for Elasticsearch in Interaction Recording Web Services. For example:

```
...
elasticSearchSettings:
  retriesOnConflict: 2
  waitToIndexTimeout: 5000
  useTransportClient: true
  transportClient:
    nodes:
      - {host: 127.0.0.1, port: 9300}
    useSniff: true
    ignoreClusterName: true
    pingTimeout: 10000
```



```

nodesSamplerInterval: 10000
scanReadTimeoutSeconds: 60
countReadTimeoutSeconds: 60
scrollTimeoutSeconds: 240

```

RestClientSettings

Name	Mandatory	Default Value	Valid Values	Description
nodes	Yes, if useRestClient is true.	null	A comma-separated list of IP addresses or host names as given in sample.	Specifies the list of Elasticsearch nodes the REST client should connect to.
authentication.type	No	none	none, basic	Authentication method.
authentication.userName	Yes, if Elasticsearch authentication type is basic .	None	A valid Elasticsearch username.	Specifies the username if Elasticsearch is configured to use authentication.
authentication.password	Yes, if Elasticsearch authentication type is basic .	None	A valid Elasticsearch password.	Specifies the password if Elasticsearch is configured to use authentication.
useSSL	No	None	true,false	Specifies whether Elasticsearch should use Secure Sockets Layer (SSL). This option is only valid for Elasticsearch versions 8.18.2 and higher.
truststore	No	None	A valid path.	Specifies the path to the truststore.
truststorePassword	No	None	A valid password.	Specifies the password for the truststore.

Mandatory: No

Specifies the configuration for Elasticsearch in Interaction Recording Web Services. For example:

```

...
elasticSearchSettings:
  retriesOnConflict: 3
  useRestClient: true
  restClient:
    nodes: [ToBeChanged: <ELASTIC_SEARCH_REST_NODES>]
    authentication:
      type: [ToBeChangedOrRemoved: none|basic]
      userName: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_USER_NAME>]
      password: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_PASSWORD>]
    useSSL: [ToBeChangedOrRemoved: "true"|"false"]

```

```

truststore: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PATH>]
truststorePassword: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PASSWORD>]
waitToIndexTimeout: 5000
scanReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCAN_READ_TIMEOUT_SECONDS>]
scrollTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCROLL_TIMEOUT_SECONDS>]
countReadTimeoutSeconds: [ToBeChangedOrRemoved:
<ELASTIC_SEARCH_COUNT_READ_TIMEOUT_SECONDS>]

```

Recording

recordingSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
auditLogDeletedFiles	No	None	If set to true, Interaction Recording Web Services generates an audit log for each individual recording file that is deleted.
recordCryptoServerDecryptMaxConnection	No	50	Specifies the maximum TCP connections to each Recording Crypto Server instance defined in local-decrypt-uri-prefix settings. See Screen Recording Local Decrypt URI Prefix for details.
recordCryptoServerDecryptMaxTotalConnection	No	10 * recordCryptoServerDecryptMaxConnection	Specifies the maximum TCP connections to all Recording Crypto Server instances defined in local-decrypt-uri-prefix settings. See Screen Recording Local Decrypt URI Prefix for details.
recordCryptoServerDecryptSocketTimeout	No	30000	Specifies the socket timeout, in milliseconds, for TCP connections to Recording Crypto Server instances defined in local-decrypt-uri-prefix settings. See Screen Recording Local Decrypt URI Prefix for details.
keyspaceNameSettingsCacheSecondsTTL	No	300	Specifies the time to live in seconds of the keyspace name for a given contact center and location from

Name	Mandatory	Default Value	Description
			Cassandra in a cache.
regionsSettingsCacheSecondsTTL	No	300	Specifies the time to live in seconds of a regions setting for a location stored in Cassandra in a cache.

Mandatory: No

Specifies the configuration for recording in Interaction Recording Web Services. For example:

```
recordingSettings:
  auditLogDeletedFiles: true
  recordCryptoServerDecryptMaxConnection: 50
  recordCryptoServerDecryptMaxTotalConnection: 500
  recordCryptoServerDecryptSocketTimeout: 30000
  regionsSettingsCacheSecondsTTL: 300
```

Screen Recording

screenRecordingSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
enableSameSiteCookieForScreenRecordingPlayback	No	false	Specifies whether Interaction Recording Web Services will return the SameSite=None and Secure cookie attributes on the cookie used when playing back screen recordings from the SpeechMiner browser application. Important: Before enabling this option, ensure that the connection between the SpeechMiner browser application and RWS is configured to use HTTPS. If you set this option to true and are using HTTP, the cookie will not be returned by the browser.
screenRecordingVoiceEnabled	No	false	Specifies whether the current Interaction Recording Web Services node supports screen recording for voice interactions. If set to false, the node rejects CometD requests from

Name	Mandatory	Default Value	Description
			the Screen Recording Service for agents with the voice channel.
screenRecordingEServicesEnabled	Enabled	false	Specifies whether the current Interaction Recording Web Services node supports screen recording for non-voice interactions. If set to false, the node rejects CometD requests from the Screen Recording Service for agents with the eServices channel.
recordingInteractionEventsTTL	TTL	172800	Specifies the time to live (TTL) for Cassandra to cache a screen recording interaction event.
clientSessionManagerCacheTTL	TTL	60	Specifies the TTL for the Interaction Recording Web Services node to cache agent information (such as the agent's name) so that the node doesn't have to read the information from Interaction Recording Web Services on each request.
contactCenterInfoManagerCacheTTL	Cache TTL	90	Specifies the TTL for the Interaction Recording Web Services node to cache contact center information so that the node doesn't have to read the information from Interaction Recording Web Services on each request.

Mandatory: No

Specifies the screen recording configuration parameters. For example:

```
...
screenRecordingSettings:
  enableSameSiteCookieForScreenRecordingPlayback: false
  screenRecordingVoiceEnabled: false
  screenRecordingEServicesEnabled: false
  recordingInteractionEventsTTL: 172800
  clientSessionManagerCacheTTL: 60
  contactCenterInfoManagerCacheTTL: 90
```

Screen Recording Connections Reporting

reportingEnabled

Default Value: false

Valid Values: true, false

Mandatory: No

Enables the Screen Recording Connection Reporting feature.

createReportingCF

Default Value: false

Valid Values: true, false

Mandatory: No

Specifies if the column families required for Screen Recording Connection reporting will be created when a contact center is created.

connectionInfoHoursTTL

Default Value: 7 * 24

Valid Values: Integer

Mandatory: No

Specifies the Time To Live (in hours) to use when writing columns to the src_rep_node_<id> column family. Screen Recording Service (SR Service) connections older than the Time to Live will not be listed in the SR Service connection information queries.

historyCountsMinutesTTL

Default Value: 24 * 60

Valid Values: Integer

Mandatory: No

Specifies the Time To Live (in minutes) to use when writing columns to the src_rep_hist_<id> column family. This number determines the maximum number of values that can be reported for a statistic in historic count queries.

Multimedia Disaster Recovery

drMonitoringDelay

Default Value: 1800

Valid Values: Integer

Mandatory: No

Specifies the interval (in seconds) that will be used for monitoring Disaster Recovery synchronization.

Caching

cachingSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
contactCenterFeaturesTTL	No	30	The TTL, in seconds, for contact-center feature IDs in cache.
contactCenterSettingsTTL	No	30	The TTL, in seconds, for contact-center custom settings in cache.

Mandatory: No

Specifies how Interaction Recording Web Services should handle various caching scenarios. For example:

```
...
cachingSettings:
  contactCenterFeaturesTTL: 30
  contactCenterSettingsTTL: 30
```

DoS Filter

enableDosFilter

Default Value: false

Valid Values: true, false

Mandatory: No

Enables the denial of service filter. If you set the value to true, you must also set values for the [dosFilterSettings](#) option.

dosFilterSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
maxRequestsPerSec	No	25	Specifies the maximum number of requests from a connection per second. Requests that exceed this are first delayed, then throttled.
delayMs	No	100	Specifies the delay, in milliseconds, imposed on all requests over the rate limit, before they are considered at all. Valid values: <ul style="list-style-type: none"> • -1 = reject request • 0 = no delay • Any other number = delay in milliseconds

Name	Mandatory	Default Value	Description
maxWaitMs	No	50	Specifies the length of time, in milliseconds, to blocking wait for the throttle semaphore.
throttledRequests	No	5	Specifies the number of requests over the rate limit that are able to be considered at once.
throttleMs	No	30000	Specifies the length of time, in milliseconds, to asynchronously wait for semaphore.
maxRequestMs	No	30000	Specifies the length of time, in milliseconds, to allow the request to run.
maxIdleTrackerMs	No	30000	Specifies the length of time, in milliseconds, to keep track of request rates for a connection, before deciding that the user has gone away, and discarding the connection.
insertHeaders	No	true	If set to true, DoSFilter headers are inserted into the response.
trackSessions	No	true	If set to true, the usage rate is tracked by session if a session exists.
remotePort	No	false	If set to true and session tracking is not used, then the rate is tracked by IP address + port (effectively connection).
ipWhitelist	No	""	A comma-separated list of IP addresses that is not rate limited.

Mandatory: No

Specifies how Interaction Recording Web Services should handle denial of service. For example:

```
...
enableDosFilter: true
dosFilterSettings:
  maxRequestsPerSec: 30
  ipWhitelist: 192.168.0.1,192.168.0.2
```

These options only take effect if **enableDosFilter** is set to true.

multiPartResolverMaxUploadSize**Default Value:** 536870912**Valid Values:** Integer**Mandatory:** Yes

This parameter should be aligned with `maxDurationMinutes`, so if you change its value, ensure that you also consider the `maxDurationMinutes` value specified within the Advanced Configuration for the Screen Recording Service section in the [Deploying the Screen Recording Service - Advanced Configuration](#) page. The maximum size of a file that can be uploaded by the Screen Recording Service must be less than or equal to the `multiPartResolverMaxUploadSize`.

multiPartResolverMaxInMemorySize**Default Value:** 67108864**Valid Values:** Integer**Mandatory:** Yes

Specifies the maximum allowed size (in bytes) before uploads are written to disk.

Media Life Cycle management**backgroundScheduledMediaOperationsSettings****Default Value:** None**Valid Values:**

Name	Mandatory	Default Value	Description
<code>enableBackgroundScheduledMediaOperations</code>	No	false	Specifies whether this Interaction Recording Web Services node can perform MLM operations.
<code>schedulerThreads</code>	No	4	Specifies the number of scheduler worker threads.
<code>schedulePollingInterval</code>	No	60	Specifies how often, in seconds, Interaction Recording Web Services polls for gir-scheduler settings and synchronizes the rule schedule.
<code>speechMinerMaxConnection</code>	No	20	Specifies the maximum number of concurrent TCP connections for the same route when Interaction Recording Web Services issues API requests to SpeechMiner.
<code>speechMinerMaxTotalConnection</code>	No	-1	Specifies the size of the connection pool when

Name	Mandatory	Default Value	Description
			Interaction Recording Web Services issues API requests to SpeechMiner. If the value of this option is less than 1, Interaction Recording Web Services sets the size of the pool to the value <code>speechMinerMaxConnection * 10</code> .
<code>speechMinerSocketTimeout</code>	No	60000	Specifies how long Interaction Recording Web Services should wait, in milliseconds, for the SpeechMiner API response before timing out.
<code>defaultBackupExportURI</code>	No	None	Specifies the location to store backed up recordings. For example, <code>file:///tmp/archLocDefault</code> .
<code>useFullPathInMediaFileBackup</code>	No	false	Specifies whether to include the full path or file name only during an MLM backup operation
<code>enableScanAndScroll</code>	No	false	Specifies whether to turn on the feature where MLM uses Elasticsearch scan and scroll queries to determine the recording IDs on which to act.
<code>scanIntervalsPerDay</code>	No	24	When MLM is configured to use Elasticsearch scan and scroll queries to determine the recording IDs on which to act, this parameter determines the number of scan intervals used in a day of recordings. Reduce this value to reduce the number of Elasticsearch scan queries performed by an MLM Task for its work, assuming that all other things remain equal. Reducing this value also increases the lifetime of the search context

Name	Mandatory	Default Value	Description
			<p>created by each Elasticsearch scan query, which in turn increases the number of open file descriptors in use by Elasticsearch.</p> <p>Note: When configuring, ensure that the number of seconds in a day (i.e. 24 * 60 * 60) is exactly divisible by the configured value.</p>

Mandatory: No

Specifies the configuration for Interaction Recording Web Services to schedule purge and backup events. For example:

```
backgroundScheduledMediaOperationsSettings:
  enableBackgroundScheduledMediaOperations: true
  schedulerThreads: 4
  schedulePollingInterval: 60
  speechMinerMaxConnection: 20
  speechMinerMaxTotalConnection: -1
  speechMinerSocketTimeout: 60000
  defaultBackupExportURI:
  useFullPathInMediaFileBackup: false
  enableScanAndScroll: true
  scanIntervalsPerDay: 24
```

CometD

cometDSettings

Default Value: None**Valid Values:**

Name	Mandatory	Default Value	Description
cometdSessionExpirationTimeout		60	Specifies the timeout for the CometD session to expire on disconnect. It might take an additional minute for the session to be closed after it expires. If you set this option to -1, the session never expires. An agent can log in again before the end of this timeout to disable session expiration.
closeHttpSessionOnCometDExpiration		true	Enables or disables HTTP session invalidation when CometD times out.

Name	Mandatory	Default Value	Description
maxSessionsPerBrowser	No	1	Specifies the maximum number of sessions (tabs/frames) allowed to long poll from the same browser; a negative value allows unlimited sessions.
multiSessionInterval	No	2000	Specifies the period of time, in milliseconds, for the client normal polling period, in case the server detects more sessions (tabs/frames) connected from the same browser than allowed by the maxSessionsPerBrowser parameter. A non-positive value means that additional sessions will be disconnected.

Mandatory: No

Specifies the configuration for the CometD-specific transport server embedded into the Interaction Recording Web Services application. For example:

```
cometDSettings:
  cometdSessionExpirationTimeout: 60
  closeHttpSessionOnCometDExpiration: true
  maxSessionsPerBrowser: 2
  multiSessionInterval: 4000
```

Log header

enableLogHeader

Default Value: false

Valid Values: true, false

Mandatory: No

Specifies whether Interaction Recording Web Services includes a header in its main log file. This header contains key information about the Interaction Recording Web Services installation, including the version, start time, libraries, and any applicable settings from the **application.yaml** file.

updateOnPremiseInfoInterval

Default Value: 600

Valid Values: Integer

Mandatory: No

Specifies a period (in seconds) during which the premise environment log header information is updated.

updateOnStartup

opsCredentials

Default Value: false

Valid Values: true, false

Mandatory: No

Changes take effect: When the Interaction Recording Web Services server is started or restarted. Specifies whether to update the stored ops credentials to the values specified in the **opsUserName** and **opsUserPassword** parameters.

features

Default Value: false

Valid Values: true, false

Mandatory: No

Changes take effect: When the Interaction Recording Web Services server is started or restarted. Specifies whether to update the supported features to the list specified in the **feature-definitions.json** file. See [Enabling features in the Feature Definitions file](#) for details.

onPremiseSettings

Settings in this section are listed under **onPremiseSettings**.

Important

- The following settings should be specified for the sync node (syncNode: true). They are not required on the other nodes in the cluster.
- Note that settings under **onPremiseSettings** are used only once during the first initialization of RWS on the sync node. Further changes in the environment are retrieved from the Configuration Server directly. If a setting is configured incorrectly, please contact Genesys Customer Care for support.

cmeHost

Default Value: None

Valid Values: A valid IP address or host name

Mandatory: Yes (for sync node), No (all other nodes)

Specifies the Configuration Server host name (FQDN) or IP address.

cmePort

Default Value: None

Valid Values: A valid port

Mandatory: Yes (for sync node), No (all other nodes)
Specifies the Configuration Server port.

backupCmeHost

Default Value: None

Valid Values: A valid IP address or host name

Mandatory: Yes (for sync node), No (all other nodes)

Specifies the backup Configuration Server host name (FQDN) or IP address. You should only configure this option if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

backupCmePort

Default Value: None

Valid Values: A valid port

Mandatory: Yes (for sync node), No (all other nodes)

Specifies the backup Configuration Server port. You should only configure this option if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

countryCode

Default Value: None

Valid Values: A two-letter country code

Mandatory: Yes (for sync node), No (all other nodes)

The premise contact center's country code. For example, "US".

tlsEnabled

Default Value: None

Valid Values: true, false

Mandatory: Yes (for sync node), No (all other nodes)

Specifies whether Interaction Recording Web Services should use a secure connection to the Configuration Server.

Deploying Web Services and Applications for GIR

Warning

- The content on this page only applies to version 8.5.210.02 or earlier of Genesys Interaction Recording. If you're using a later version, you'll need to install the Interaction Recording Web Services component instead. See [Deploying Interaction Recording Web Services](#) for details.
- If you upgrade to Interaction Recording Web Services, it does not provide API support for non-GIR related Web Services, such as Workspace Web Edition.

Genesys Interaction Recording (GIR) needs Web Services to store and manage the recording files.

Web Services uses the following major components:

- [WebDAV Server](#)—The file management device that stores and manages the GIR recording files
- [Cassandra Database](#)—The java-based cluster-schemed database for Web Services to store interaction metadata.
- [Web Services Server](#)—A REST API server that pushes and pulls the interaction metadata to and from the Cassandra database.
- [Workspace Web Edition](#)—The web-based Agent Desktop.

Important

- Screen Recording for voice interactions requires that Agent Info specify the default agent place in the Default Place field. The default place must be assigned at least one DN. If a DN is not assigned to the default place, the agent will not be associated with a device and Screen Recording will not produce interactions for the agent.
- The 8.5.201.29 release of Genesys Web Services and Applications is not supported with the Genesys Interaction Recording Solution.
- The URI in recording_settings is case sensitive and must match the URI in the IVR Profile. For example:

```
"uri": "http://GENESYSREC1/recordings"
```

is not the same as:

```
"uri": "http://genesysrec1/recordings"
```

The following steps describe how to deploy the Web Services components for GIR.

Deploy the WebDAV Storage Server

1. Install WebDAV, by running the following command:

```
yum install httpd
```

2. Edit the `/etc/httpd/conf/httpd.conf` file, and append the following to the end of the file:

```
Alias /recordings /mnt/recordings
<Directory /mnt/recordings>
    Options Indexes MultiViews FollowSymLinks
    EnableSendfile off
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
<Location "/recordings">
    DAV On
    AuthType Basic
    AuthName "user"
    AuthUserFile /var/www/htpasswd
    Require valid-user
</Location>
```

3. Open the firewall. Because WebDAV is an HTTP server, the incoming default HTTP and/or HTTPS ports (80 and/or 443) must be open to the server.

Important

It is possible to use custom ports by changing the permitted incoming ports in the firewall, the virtual host configuration file, and the URL used to reach the WebDAV server.

4. Create the directory to keep the recording files, and set the permission to Apache, using the following command:

```
mkdir /mnt/recordings
chown apache:apache /mnt/recordings
```

Important

Due to performance concerns, Genesys does not recommend using a remote directory for WebDAV.

5. Create a WebDAV user for httpd, and configure the password. The following example creates a user called "user":
`htpasswd -c /var/www/htpasswd user`

Warning

If the Recording Muxer is deployed for screen recording, make sure all WebDAV storages of the same contact center region are using the same username and password.

6. Configure the httpd to start on boot up (and start it now) using the following command:

```
chkconfig --levels 235 httpd on  
service httpd start
```

7. Test the WebDAV installation.

- a. Upload a `hello.world` file to the WebDAV server using the following command:

```
curl -T hello.world -u user:password http://myserver/recordings/hello.world
```

- b. Using a browser, open the `http://myserver/recordings/hello.world` URL. The browser will request for user credentials.

8. The WebDAV server is installed.

Deploy the Cassandra Database

Web Services stores the information about call recordings in a Cassandra database. For each contact center, the distinct column families with unique names exist for storing call recording information. These column families are created when the contact center is created, and deleted when contact center is deleted.

To deploy the Cassandra database for GIR, see the [Installing and Configuring Cassandra](#) section of the *Web Services and Applications Guide*.

Important

Web Services deletes column families only if they do not contain any call recordings; otherwise they should be deleted manually from Cassandra using the `cassandra-cli` tool.

Deploy Web Services and Applications

To install and configure Web Services and Applications, see the [Web Services and Applications Guide](#).

For Voice Recordings

Web Services requires a specific configuration in addition to the configuration that is described in the *Web Services and Applications Deployment Guide* for GIR **call** recordings to work correctly. The

following sections describe how to configure Web Services for call recordings.

Configuring the Web Services Parameters

To configure Web Services for Genesys Interaction Recording, add parameters to the **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the **server-settings.yaml** instead).

[+] Show the Parameters

Parameter Name	Mandatory	Description	Type	Default Value
enableBackgroundScheduledMediaOperations	N	Specifies whether to allow Web Services to schedule purge and backup events.	Boolean	True
createCallRecordingCn	N	Specifies whether to create a call recording column family (CRCF) for a new contact center.	Boolean	False
crClusterName	Y	Specifies the name of the elasticsearch cluster name.	Non-empty String	None Note: This is a mandatory parameter, and the value must be the same for all Web Services nodes in the cluster. For example, if there are five nodes in the Web Services cluster, all five nodes must have the same crClusterName value.
crRegion	N	Specifies the name of the region where the Web Services node resides.	Non-empty String	None
cryptoSecurityKey	Y	Specifies the security key used for Web Services encryption of the recording settings in the database.	Non-empty String	None Note: This is a mandatory parameter, and the value must be the same for all Web Services nodes in the cluster. For example, if there are five nodes in the Web Services cluster, all five nodes must have the same cryptoSecurityKey value.
defaultBackupExportURL	N	Specifies the location to store	Non-empty String	None

Parameter Name	Mandatory	Description	Type	Default Value
		backed up recordings. For example, file:///tmp/archLocDefault '.		
multiPartResolverMaxUploadSize		Specifies the maximum size, in KB, of the recording file.	Integer	536870912
multiPartResolverMaxInMemorySize		Specifies the maximum length of time allowed to upload a recording file.	Integer	536870912
nodePath	Y	Specifies the location and ID of the Workspace Web Edition & Web Services node within the deployment topology. This value must be unique across the deployment. For example, a value of /US/node1 means that the node is located in the US region and has an ID of "node1". The node ID can be the hostname, the IP address, or any other unique identifier.	Non-empty String	None
recordCryptoServerDecryptMaxConnection	N	Specifies the maximum TCP connections to each Recording Crypto Server instance defined in local-decrypt-uri-prefix settings. Note: This option applies to the Web Services version 8.5.200.85 and later only.	Integer	50
recordCryptoServerDecryptMaxTotalConnection	N	Specifies the maximum TCP	Integer	10 * recordCryptoServerDecryptMax

Parameter Name	Mandatory	Description	Type	Default Value
		connections to all Recording Crypto Server instances defined in local-decrypt-uri-prefix settings. Note: This option applies to the Web Services version 8.5.200.85 and later only.		
recordCryptoServerDecryptSocketTimeout		Specifies the socket timeout, in milliseconds, for TCP connections to Recording Crypto Server instances defined in ' <i>local-decrypt-uri-prefix</i> ' settings. Note: This option applies to the Web Services version 8.5.200.85 and later only.	Integer	30000
webDAVMaxConnections		Specifies the maximum TCP connections for each WebDAV Storage.	Integer	50
webDAVMaxTotalConnections		Specifies the maximum TCP connections the Web Services node allows to all WebDAV Storages.	Integer	10 * webDAVMaxConnections
undocumentedExternalApiUrl		Specifies the reachable Web Services Server address for the SpeechMiner UI, and the Screen Recording Client. Note: This option applies to the Web Services version 8.5.200.40 and later only.	String	http://<IP Address>:8090/internal-api

Configuring the Elasticsearch Engine

The Web Services Call Recording API uses the elastic search as the query engine. A configuration file

is required if call recording is enabled (for example, **JETTY_HOME/resources/elasticsearch.yml**).

[+] Show the Steps to Configure Elasticsearch

Configure the **JETTY_HOME/resources/elasticsearch.yml** file as follows:

```
index.analysis.analyzer.whitespace_lowercase.tokenizer: whitespace
index.analysis.analyzer.whitespace_lowercase.filter: lowercase

transport.tcp.port: 9200
http.port: 9300

discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: <comma separated list of HTCC nodes which host the ES>
discovery.zen.minimum_master_nodes: 2

gateway.recover_after_nodes: 2
gateway.recover_after_time: 1m
gateway.expected_nodes: 3

threadpool.index.queue_size: -1
threadpool.bulk.queue_size: -1

path.conf: <Path to genconfig folder>/elasticsearch
path.data: <Path to the folder where ES stores its data>
```

For more configuration information, see <http://www.elasticsearch.org/guide/>.

The Elasticsearch engine also requires a large PermGen space.

To increase the PermGen space:

- Add the following to your **JAVA_OPTIONS**:

```
JAVA_OPTIONS="-XX:MaxPermSize=512m -Djsse.enableSNIExtension=false"
```

- If you are using **/etc/default/jetty**, add:

```
JAVA_OPTIONS="-Xmx2048m -XX:MaxPermSize=512m -Xms2048m -Djsse.enableSNIExtension=false"
```

Important

The Elasticsearch index is saved in the **Jetty-Home/data** directory—for example, **/opt/jetty/data**.

Rebuilding the Elasticsearch Index

If you must upgrade your Jetty 8 version to Jetty 9 version, you might need to add the elasticsearch data file to the new Web Services cluster.

To move the elasticsearch data:

- Rebuild the elasticsearch index using the following command:

```
curl -XPOST "http://<FE VM host>/api/v2/ops/contact-centers/<ID contact center>/recordings"
-d '{ "operationName":"forceIndex", "from":<Time of previous 'green' state or backup
snapshot>}'
```

The command above executes the forceIndex operation and is used to rebuild the elasticsearch index when needed. The following information provides additional details for this API.

HTTP Request

POST
.../api/v2/ops/contact-centers/{id}/recordings

Request Body

```
{
  "operationName":"forceIndex",
  "from":1369272257713,
  "to":1369275857713,
  "purgeOld":true
}
```

The following table describes the request body attributes:

Attributes	Type	Mandatory	Description
operationName	String	Y	The name of the operation. In this case it is forceIndex.
from	Long Integer	Y	The Java time stamp (in milliseconds) which equals the UNIX time * 1000. This is the time stamp from which the records are re-indexed.
to	Long Integer	N	The Java time stamp (in milliseconds) which equals the UNIX time * 1000. This is the time stamp to which the records are re-indexed. If not specified, the

Attributes	Type	Mandatory	Description
			current time of the request processing is used.
purgeOld	Boolean	N	Specifies whether the old index should be deleted prior to re-indexing. This attribute is necessary if the Web Services updated version uses indexes with a different structure. The default value is false.

Restarting Web Services

For more information on starting and stopping Web Services, see the [Web Services Deployment Guide](#).

Configuring the Storage Credentials for Web Services

To enable voice recording:

1. Determine the contact center ID on Web Services using the following command with the ops username and password (ops:ops):

```
curl -u ops:ops http://<Web Services Server>:8080/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://<Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>"]}
```

Important

Use the <contact center ID (in hex format)> in all subsequent commands.

2. In a text editor, create the create_table file using the following command:

```
{
  "operationName": "createCRCF"
}
curl -u ops:ops -X POST -d @create_table http://htcc:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>/recordings
--header "Content-Type: application/json"; echo
```

To enable storage:

1. Using a text editor, create a new file called recording_settings with the following content:

```
{
```

```
"store": [
  {
    "webDAV": {
      "userName": "user1",
      "password": "password1",
      "uri": "http://apache1/recordings"
    }
  },
  {
    "webDAV": {
      "userName": "user2",
      "password": "password2",
      "uri": "http://apache2/recordings"
    }
  }
]
```

Important

The URI in `recording_settings` is case sensitive and must match the URI in the IVR Profile. For example:

```
"uri": "http://GENESYSREC1/recordings"
```

is not the same as:

```
"uri": "http://genesysrec1/recordings"
```

2. Execute the following command:

```
{
  curl -u ops:ops -X PUT -d @recording_settings
    http://<Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex
format)/settings/recordings
    --header "Content-Type: application/json"; echo
}
```

Configuring the Call Recording Audit Log

Web Services provides an audit log for the following call recording operations:

- Playback of the recording media file
- Deletion of the call recording file

To configure the audit log:

1. Stop the Web Service Jetty using the following command:
`sudo service jetty stop`
2. Update the Jetty LogBack Configuration:
 - Edit the `/opt/jetty/resources/logback.xml` file to include INFO level messaging **[+] Show example**
:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Example LOGBACK Configuration File
  http://logback.qos.ch/manual/configuration.html
-->
<configuration scan="true">
  <appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <filter class="ch.qos.logback.classic.filter.LevelFilter">
      <level>INFO</level>
      <onMatch>ACCEPT</onMatch>
      <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY for
printing only INFO-->
    </filter>
    <file>${jetty.logs}/recording.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <maxHistory>720</maxHistory><!-- 1 Month -->
    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}]
[%X{req.userAgent}] [%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
    </encoder>
  </appender>
  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${jetty.logs}/cloud.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <!-- hourly rollover -->
      <fileNamePattern>${jetty.logs}/cloud-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <!-- keep 5 days' worth of history -->
      <maxHistory>120</maxHistory>
    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} %-5level [%X{principal.name}]
[%X{session}] [%X{contactCenter}]
[%thread] %X{req.requestURI} %X{req.queryString} %logger{36} %msg%n</pattern>
    </encoder>
  </appender>
  <logger name="com.<domain>.cloud.v2.api.controllers.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>.cloud.v2.api.tasks.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>" level="WARN" />
  <logger name="com.<domain>.cloud" level="DEBUG" />
  <logger name="com.<domain>.cloud.rtreporting" level="WARN" />
  <logger name="com.<domain>.salesforce.security" level="INFO" />

  <root level="WARN">
    <appender-ref ref="FILE" />
  </root>
</configuration>
```

- For MLM:

- Create a **RECORDING** appender if it does not exist. **[+]** Show example

```
<appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <filter class="ch.qos.logback.classic.filter.LevelFilter">
    <level>INFO</level>
    <onMatch>ACCEPT</onMatch>
```



```

        <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY
for printing only INFO-->
    </filter>
    <file>${jetty.logs}/recording.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
        <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd}.gz</fileNamePattern>
        <maxHistory>720</maxHistory><!-- 1 Month -->
    </rollingPolicy>
    <encoder>
        <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}]
[%X{req.userAgent}] [%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
    </encoder>
</appender>

```

- Add the following loggers:

```

<logger name="com.genesyslab.cloud.v2.api.controllers.callrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.controllers.screenrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.callrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.interactionrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.screenrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.settings">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.media.scheduler">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.media.task">
    <appender-ref ref="RECORDING" />
</logger>

```

For more information about Jetty Logback, see [Logback configuration](#).

3. Start Jetty using the following command:

```
sudo service jetty start
```

4. Review the audit log. **[+] Show example**

- Open the **/var/log/jetty/recording.log** file. The following example shows that two recordings are requested for playback and deletion:

```

10/28/2013 15:46:03.203 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/
537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

```

```

10/28/2013 15:46:03.341 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-

```

```

d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] failed

10/28/2013 15:46:10.946 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

10/28/2013 15:46:11.033 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] succeed

10/28/2013 15:46:52.179 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid0 Delete recording [reci
d0] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] requested

10/28/2013 15:46:52.216 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid0 Delete recording
[recid0] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] failed

10/28/2013 15:46:56.253 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete recording [reci
d1] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] requested

10/28/2013 15:46:56.420 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete recording
[recid1] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] succeeded

```

Setting the Advanced Options for Web Services

API Thread Pool

Web Services provides properties for the Call Recording API thread pool via `archaius.[+]` **Show the properties**

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
<code>hystrix.command.[API Name].</code>	N/A	The hystrix timeout. The default

Property/API Name	Thread Pool Name	Description
execution.isolation.thread.timeoutInMilliseconds		value is set to 6000.
hystrix.threadpool.[API Pool Name] .coreSize	N/A	The thread pool size. The default value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateCallRecordingApiTaskV2	ApiCreatePool	Create call recording
DeleteCallRecordingApiTaskV2	ApiDeletePool	Delete call recording
GetCallRecordingApiTaskV2	ApiGetPool	Get call recording meta data
GetCallRecordingCFInfoApiTaskV2	ApiGetPool	Get call recording CF Information
GetCallRecordingMediaApiTaskV2	ApiGetPool	Streaming call recording media
QueryCallRecordingApiTaskV2	ApiQueryPool	Query call recording Meta data

For more information about the Web Services Call Recording API, see the [Web Services API Reference](#).

For more information about how to use Workspace Web Edition for Voice Recording, see the [Workspace Web Edition Help](#).

For Screen Recordings

As with call recordings, Web Services requires a specific configuration for GIR **screen** recordings to work correctly. The following sections describe how to configure Web Services for screen recordings.

Configuring the Parameters

1. On all Web Services instances, modify the **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the **server-settings.yaml** instead), and add the following parameters:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: <Web Services Servers>,<SpeechMiner Web Servers>
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,Range"
  allowCredentials: true
screenRecordingSettings:
  screenRecordingEServicesEnabled: true
  screenRecordingVoiceEnabled: true
screenRecordingConnectionReportingSettings:
  reportingEnabled: true
  createReportingCF: true
multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 536870912
```

Important

- Change <Web Services Servers> and <SpeechMiner Web Servers> to the HTTP/HTTPS addresses of the Web Services instances and SpeechMiner Web Servers.
- **multipartResolverMaxUploadSize** controls the maximum allowed size for the Screen Recording video file that can be uploaded to Web Services and Applications (in bytes). Setting the value too high (10MB+) for this parameter may cause performance and/or security issues for Web Services and Applications.

2. Add screen recording features to the Contact Center:

```
POST http://<htcc-host-prefix>/api/v2/ops/contact-centers/
bea09df2-82c5-441a-9072-5f2fc15fadc4/features
{
  "uris":[
    "/api/api-voice-screenrecording",
    "/api/api-multimedia-screenrecording",
    "/api/api-screenrecording-connection-reporting"
  ]
}
```

Important

- Use the **api-voice-screenrecording** parameter for voice interactions, and use the **api-multimedia-screenrecording** parameter for non-voice interactions.
- If you wish to direct the SpeechMiner UI to Web Services instead of Recording Crypto Server for decryption of screen recordings, add the **api-recordings-decryption-proxying** parameter to the list of features enabled for the contact center above. Note that this requires additional configuration and applies to the Web Services version 8.5.200.85 and later only.

Restarting Web Services

For more information on starting and stopping Web Services, see the [Web Services Deployment Guide](#).

Configuring the Storage Credentials for Web Services

1. Determine the contact center ID on Web Services using the following command with the ops username and password (ops:ops):

```
curl -u ops:ops http://<Web Services Server>:<Web Services port>/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://<Web Services Server>:<Web Services port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>"]}
```

Important

Use the <contact center ID (in hex format)> in all subsequent commands.

2. In a text editor, create a new file called `create_table`, with the following content:

```
{
  "operationName": "createCRCF"
}
```

And then execute the following command:

```
curl -u ops:ops -X POST -d @create_table http:// <Web Services Server>:<Web Services Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/screen-recordings --header "Content-Type: application/json"; echo
```

3. Enable storage for a single or multiple locations:

- For a **single** location:

- a. In a text editor, create the `create_single_location` file. **[+] Show how**

```
{
  "name": "storage",
  "location": "/",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "<webdav user>",
        "password": "<webdav password>",
        "storagePath": "<webdav uri>"
      }
    }
  ]
}
```

Important

Replace <webdav user>, <webdav password>, <webdav uri> with the appropriate values.

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_single_location http:// <Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>/settings/screen-recording --header "Content-Type: application/json"; echo
```

- For **multiple** locations:

- a. In a text editor, create the `create_first_location` file. **[+] Show how**

```
{
  "name": "storage",
  "location": "<node_location>",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "<webdav user>",
        "password": "<webdav password>",
        "storagePath": "<webdav uri>"
      }
    }
  ]
}
```

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_first_location http://<Web Services
Server>:8080/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
```

Important

Replace <node_location>, <webdav user>, <webdav password>, <webdav uri> with the appropriate values. The values for the <node_location> are similar to the nodePath settings in the Web Services **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the server-settings.yaml instead), but allow a hierarchical representation. For example, a Web Services node uses a storage setting with a location of "/US" in the nodePath set to "/US/AK" or "/US/HI".

For more information on hierarchical location setting, see https://docs.genesys.com/Documentation/CR/8.5.2/Solution/GWSSettings#Hierarchical_Location_Matching.

- c. Repeat steps a and b for each location required.

For more information on the properties of this settings group, see [Interaction Recording Web Services Settings Groups](#).

Setting the Advanced Options for Web Services

API Thread Pool

Web Services provides properties for the Screen Recording API thread pool via archaius.[+] **Show the properties**

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name]. execution.isolation.thread. timeoutInMilliseconds	N/A	The hystrix timeout. The default value is set to 6000.

Property/API Name	Thread Pool Name	Description
hystrix.threadpool.[API Pool Name] .coreSize	N/A	The thread pool size. The default value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateScreenRecordingApiTaskV2	ApiUploadPool	Create screen recording
DeleteScreenRecordingMediaApiTaskV2	ApiDeletePool	Delete screen recording
GetScreenRecordingApiTaskV2	ApiGetPool	Get screen recording meta data
GetScreenRecordingMediaApiTaskV2	ApiStreamPool	Stream screen recording media
QueryScreenRecordingApiTaskV2	ApiQueryPool	Query screen recording meta data

For more information about the Web Services Call Recording API, see the [Genesys Interaction Recording API Reference](#).

Deploying SIP Server for GIR

Genesys Interaction Recording (GIR) needs SIP Server for routing, call control and to initiate the recordings. The following steps describe how to deploy and configure SIP Server for GIR, and how to configure the DNSs for GIR.

You can also use these configuration settings with SIP Cluster, but certain limitations might apply. Any limitations for SIP Cluster are noted in each section, where applicable.

For more information about the SIP Server configuration settings described on this page, see the [SIP Server Deployment Guide](#).

SIP Server

1. Install and configure SIP Server as described in the [SIP Server Deployment Guide](#).
2. In addition to the configuration described in the deployment guide, set the following SIP Server options:

Section Name	Parameter Name	Description
TServer	msml-support	Set to <code>true</code> to enable support of the call recording solution.
	resource-management-by-rm	Set to <code>true</code> to enable support of the call recording solution. Resource monitoring and notification will be done by the Resource Manager. SIP Server will contact Media Server through Resource Manager.
	msml-record-support	Set to <code>true</code> to enable SIP Server to engage GVP as a Media Server through the msml protocol for call recording.
	msml-record-metadata-support	Set to <code>true</code> to send additional metadata in the INFO message of Genesys Media Server when starting call recording.
	record-consult-calls	Specifies whether to record consult calls: <ul style="list-style-type: none"> • <code>true</code>—record consult calls. • <code>false</code>—do not record consult calls.
	recording-filename	Must be set to <code>\$UUID\$_\$DATE\$_\$TIME\$</code>
	wrap-up-time	(Optional) Duration of time (in seconds) to record the agent's

Section Name	Parameter Name	Description
		screen while they are in the After Call Work (ACW) state. For more information, see Agent Login .

VoIP Service DN

1. Create a new MSML DN object and add the following parameters to the **General** tab:
 - **Number** = The name of the MSML Server
 - **Type**= Voice over IP Service
2. Add the following parameters to the **Annex** tab of the new DN:

Section Name	Parameter Name	Description
TServer	contact	Set this to the Resource Manager IP address and port. Use the following format: sip: <Resource Manager_IP_address:Resource Manager_SIP_port> Specifies the contact URI that SIP Server uses for communication with the treatment server.
	service-type	Set to msml
	prefix	Set to msml=
	subscription-id	Set to the name of the tenant to which this SIP Server belongs, using the following syntax <TenantName>
	refer-enabled	Set to false
	make-call-rfc3725-flow	Set to 1
	ring-tone-on-make-call	Set to false
	sip-hold-rfc3264	Set to true
	oos-check	Set to 5
	oos-force	Set to 4

Agent DN

On the Agent's DN, in the **[TServer]** section, set the following parameters:

- If you want to start recording based on static DN-level settings, set the **record** parameter to true.

Important

This parameter can be set in either the **Agent DN** or **Agent Login** object, but not both. If setting it in **Agent DN**, make sure that the **record** parameter is not set to true in **Agent Login**.

- If you are using WDE or WWE, set the **enable-agentlogin-presence** parameter to false as the required information is provided by WDE or WWE.
- If you are not using WDE or WWE, set **enable-agentlogin-presence** to true. This option is required to provide agent hierarchy and name to SpeechMiner to ensure correct access limitations.

Agent Login

On the **Annex** of the Agent Login object, in the **[TServer]** section, set the following parameters:

- To start recording based on static DN-level settings, set the **record** parameter to true.

Important

This parameter can be set in either the **Agent Login** or **Agent DN** object, but not both. If setting it in **Agent Login**, make sure that the **record** parameter is not set to true in **Agent DN**.

- If you want to record the agent's screen while they are in the After Call Work (ACW) state, set the **wrap-up-time** in seconds; for example, set **wrap-up-time=10**. For more information, see the **isACWEnabled** parameter on the **Deploying the Screen Recording Service - Advanced Configuration** page.

Important

Agent Login objects are not supported if you are using SIP Cluster.

Deploying Interaction Concentrator for GIR

Important

The ICON deployment procedure is not required when using the Voice Processor instead of the Recording Processor Script (RPS).

Genesys Interaction Recording needs Interaction Concentrator (ICON) to store detailed reporting data from various sources in a contact center empowered with Genesys software.

Installing ICON

Install and configure ICON as described in the [ICON Deployment Guide](#). You can read more about ICON [here](#).

Important

Genesys Interaction Recording requires that the ICON database be case insensitive. Genesys also recommends that you use a separate ICON database for GIR (for example, do not use the same ICON database for GIR that is being used by Genesys Info Mart reporting).

If you want to deploy a single instance of the ICON database across multiple sites, see the [Supported Deployment Scenarios](#) in the ICON Deployment Guide.

Configuring ICON

In addition to the configuration described in the deployment guide, configure your ICON application as follows:

1. To collect all metadata, in the **[callconcentrator]** section, set the following parameters:
 - **adata-reasons-history** = none
 - **adata-extensions-history** = none
 - **adata-userdata-history** = all
 - **role** = all
2. To collect attached data, in the **[custom-states]** section, set the following parameters:

- **EventData** = <type1>,<key1>,<type2>,<key2>... where <typeN> is the data type (for example, char or int) and <keyN> is the attached data key name.
- **store-event-data** = conf

To improve ICON performance for Genesys Interaction Recording, Genesys recommends updating the ICON database schema with the following new indexes:

- Index G_PARTY:
 - NONCLUSTERED/NONUNIQUE INDEX G_PARTY.CALLID
- Index G_USERDATA_HISTORY:
 - NONCLUSTERED/NONUNIQUE INDEX G_USERDATA_HISTORY.CALLID
- Index G_IS_LINK:
 - NONCLUSTERED/NONUNIQUE INDEX G_IS_LINK.CALLID
- Index G_CUSTOM_DATA_S:
 - NONCLUSTERED/NONUNIQUE INDEX G_CUSTOM_DATA_S.CALLID

For optimal performance, it is recommended that the ICON's gsysPurge81 stored procedure (or similar) be used regularly to purge call data from the ICON database that is older than two days. See the [ICON User's Guide](#) for more information.

Important

Genesys Interaction Recording requires data from the following ICON tables:

- G_IS_LINK
- G_CALL
- G_PARTY
- G_PARTY_HISTORY
- G_AGENT_STATE_HISTORY
- G_CUSTOM_DATA_S
- G_USERDATA_HISTORY
- G_SECURE_USERDATA_HISTORY
- GC_AGENT

Make sure that you are populating these tables. For more information, see the [ICON Deployment Guide](#).

Deploying Recording Crypto Server

Genesys Interaction Recording (GIR) needs the Recording Crypto Server (RCS) to manage the certificates and the encryption/decryption process when retrieving and playing back the stored recording files.

Important

- RCS does not support on-the-fly configuration changes. Restart RCS to apply changes to the Genesys Advanced Disconnect Detection Protocol (ADDP) configuration.
- RCS will not start if Configuration Server is using UCS-2 encoding. In this scenario, use UTF-8 or set the Configuration Server option **[confserv] allow-mixed-encoding** to true.

Installing Recording Crypto Server

Preparing the Host

You must install the correct JRE version on the host machine where the Recording Crypto Server will be installed. For Recording Crypto Server 8.5.095.22 (or higher), JRE 17 is required. For Recording Crypto Server 8.5.095.17 (or lower), JRE 8 is required.

Important

For more detailed information about the supported versions for each operating system, see the [Genesys Supported Operating Environment Reference Guide](#).

To install JRE:

1. Perform one of the following:

- For Recording Crypto Server 8.5.095.22 (or higher), download and install Java Runtime Environment (JRE) 17 from your preferred provider. For example, you can download this from use an OpenJDK version of the software.
 - For Recording Crypto Server 8.5.095.17 (or lower), download and install Java Runtime Environment (JRE) 8 from your preferred provider. For example, you can download this from Oracle or use an OpenJDK version of the software.
- Set the following environment variables for your host, as follows:

- (Linux) Insert the following lines into the **/etc/profile** file:
`export JAVA_HOME=/usr/lib/java/jre-<version of Java downloaded>/jre`
Log out and log in again to activate the new environment variables in the current session.
- (Windows) Create a new System Variable named **JAVA_HOME** and use the path that was used during installation as the value. To do this, right-click your Computer icon. Select **Properties > Advanced System Settings > Environment Variables**, and then create the **JAVA_HOME** variable.

Installing Recording Crypto Server Using the Deployment Wizard

For instructions about installing Recording Crypto Server using the Genesys Administrator Extension, see the [Solution Deployment](#) section of the Genesys Administrator Extension User Guide.

When Recording Crypto Server (RCS) is started for the first time, and then terminated (either by using the Solution Control Interface or by killing the process) soon after, the RCS directory structure might be left in a partially initialized state. This can cause RCS to fail on subsequent attempts to start. To work around this, do not terminate RCS for at least 60 seconds starting it for the first time. If the directory structure is still invalid, delete all sub-directories in the RCS root directory, except for the **conf** and **legal** directories. When RCS is re-started, the required directories will be created.

Installing the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files

Important

If you are using Java 17, this step is not required.

In older versions of Java 8, the default installation limits key sizes to 128 bits. Larger key sizes can be enabled by installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

Important

- If you are using an OpenJDK version of Java, no additional cryptography configuration is required.
- If the Java 8 version you are using is older than Java 8u151, then follow the installation steps for JCE described below.
- If you are using Java 8u151 or 8u152, you do not need to download and install JCE. However, you must make a change to the unlimited policy option in the **JRE_HOME/lib/security/java.security** file. Find the **#crypto.policy=unlimited** line and remove the hash (#) character to uncomment it.
- If you are using Java 8u161 or newer, no additional cryptography configuration is required.

To install:

1. If you are using the Oracle version of Java 8, download the **Java 8** specific package from the **Oracle** website and follow the instructions provided with the package.
2. Copy the **Local_policy.jar** and **Us_export_policy.jar** files to the **JRE_HOME/lib/security** directory. If there are already copies of these files in that directory, make backup copies of these existing files in case you want to revert the installation.

Important

Make sure that the policy files are installed before starting the RCS for the first time. RCS will not start without these files.

Upgrading Recording Crypto Server

1. Make a backup copy of the **rcs.properties** file.
2. Make a backup copy of the **keystore** file.
3. Uninstall the Recording Crypto Server component.
4. Install the new Recording Crypto Server component.
5. Copy the settings from the backup copy of the **rcs.properties** file to the new **rcs.properties** file.
6. Copy the backup **keystore** file to the desired **keystore** file location and update the **rcs.properties** configuration file's **keystorepath** parameter to point to this file.

Configuring Recording Crypto Server

This section describes how to configure the Recording Crypto Server in your environment using Genesys Administrator Extension.

For more information about using Genesys Administrator Extension, see the [Genesys Administrator Extension Help](#).

Configuring the KeyStore and Certificate Authority

For information on how Genesys supports TLS for secure data exchange, refer to [Securing Connections Using TLS](#) in the [Genesys Security Deployment Guide](#).

The Recording Crypto Server stores certificate and key data files based keystores. Certificates uploaded to the server can be optionally validated against a Certificate Authority (CA).

Important

The CA configuration is used for recording certificates and not for TLS network

connections. This section describes the keystore and CA related configuration parameters.

To limit access, all recording encryption key related parameters are stored in a local **<Recording Crypto Server Install Directory>/conf/rcs.properties** configuration file.

The following table lists the parameters used in the **rcs.properties** configuration file.

Parameter Name	Default Value	Description
keystorepath	keystore.bin	Specifies the path to the keystore file. If HA is enabled, the keystore file should be accessed through a network share (see Configure HA).
keystorepassword	genesys	Specifies the password that accesses the keystore file. Note: The keystorepassword parameter can be overridden by the RCS_KEYSTORE_PASSWORD environment variable. In this case the same password is used for both keystorepassword and keypassword.
keypassword	genesys	Specifies the password used for each private key that is added to the keystore. Note: <ul style="list-style-type: none">The same password is used for each private key.The keypassword parameter can be overridden by the RCS_KEYSTORE_PASSWORD environment variable. In this case, the same password is used for both keystorepassword and keypassword.
cacertstorepath	Java-R00T	Specifies the CA certificate keystore. Possible values are: <ul style="list-style-type: none">Java-R00T—The path to the default Java JRE CA certificate file.Windows-R00T—The path to the Windows system keystore. This is not valid for Linux systems.

Parameter Name	Default Value	Description
		<ul style="list-style-type: none">File Path—The path to use the CA keystore. This file must be a Java JKS keystore file.None—Disables validation of certificates.
cacertstorepassword	changeit	Specifies the password for the CA certificate keystore.

The following shows an example **rcs.properties** configuration file:

```
keystorepath=keystore.bin
keystorepassword=keystorepassword
keypassword=keypassword
cacertstorepath=Java-R00T
cacertstorepassword=capassword
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

The Recording Crypto Server uses API calls to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for recording playback and archival operations. To configure the Interaction Recording Web Services (Web Services) connection, set the following parameters in the **[htcc]** section of the Recording Crypto Server application:

Parameter Name	Default Value	Description
baseurl	https://htcchost:8080	Specifies the base URL for the Interaction Recording Web Services (Web Services) connection. This parameter is dependent on the Interaction Recording Web Services (Web Services) server protocol (http or https), port, and URL suffix.
internalUrlPrefix	/api/v2	Controls the prefix added to requests sent to Interaction Recording Web Services to retrieve recording files. By default, or if a value other than disable is specified, RCS will concatenate the baseurl , internalUrlPrefix , and the mediaPath returned by RWS as the request URL. If the internalUrlPrefix value is set to disable , RCS will use the mediaUri from the metadata instead when fetching the recordings from RWS.
domain	Empty string	Specifies the domain of the Interaction Recording Web Services (Web Services) contact

Parameter Name	Default Value	Description
		center. This is the domain ID set for the contact center within Interaction Recording Web Services (Web Services).
user	ops	Specifies the name of the operations user for the Interaction Recording Web Services (Web Services) connection.
password	opspassword	Specifies the password of the operations user for the Interaction Recording Web Services (Web Services) connection.
max-sr-playback-connections	50	Specifies the maximum number of HTTP connections between Recording Crypto Server and Interaction Recording Web Services (Web Services) for screen recording playback.
contactcenterid	Empty string	Specifies the contact center ID value in the RCS requests sent to Interaction Recording Web Services (RWS). If this value is not specified, the contact center ID information is derived from the /api/v2/ops/contact-centers request sent to RWS. Important: If you are a Recording Crypto Server API user and you specify an empty Contact Center ID (CCID) when using the /rcs/contact-centers/<ccid>/recordings/... path, you will receive a misleading HTTP 403 Access is denied message.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. RCS will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web

Parameter Name	Default Value	Description
		Services (Web Services) in the Configuring Transport Layer Security (TLS) Connections (Optional) section.

Configuring Cross Origin Resource Sharing (CORS)

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has Configuring Cross-Site Request Forgery (CSRF) protection enabled, CORS must be configured.

To configure CORS, set the following options in the **[cors]** section of the Recording Crypto Server application:

Parameter Name	Default Value	Description
allowed-origins	empty	Specifies the allowed origins list that is attached in the HTTP response Access-Control-Allow-Origins header, sent to a cross-origin request. In this case, it must be a list of all base URLs used by the users to connect to SpeechMiner Web.
allowed-headers	X-Requested-With, Content-Type, Accept, Origin, Cookie, authorization, Access-Control-Allow-Headers, X-Response-Access-Control-CenterId, X-Response-Access-Control-CenterId	Specifies the allowed headers list that is attached in the HTTP response Access-Control-Allow-Headers header, sent to a cross-origin request.
allowed-methods	GET, POST, PUT, DELETE, OPTIONS	Specifies the allowed methods list that is attached in the HTTP response Access-Control-Allow-Methods header, sent to a cross-origin request.
allow-credentials	true	Specifies the value sent in Access-Control-Allow-Credentials header of the HTTP response to cross-origin request.
enable-cors-filter	true	Enables handling cross-origin requests originating from other domains like SpeechMiner Web. The value should always be true for CORS security to be active.

Configuring SameSite Cookie for Screen Recording Playback (Optional)

The Recording Crypto Server provides the ability to enable the **SameSite=None** and **Secure** cookie attributes for the cookie used for screen recording playback in the SpeechMiner browser application. These attributes are not set by default.

To configure the **SameSite** and **Secure** cookie attributes, set the following option within the **[general]** section of the Recording Crypto Server application:

Important

Before enabling this option, ensure that the connection between the SpeechMiner browser application and Recording Crypto Server is configured to use HTTPS. If you set the value of this option to `true` and are using HTTP, the cookie will not be returned by the browser.

Parameter Name	Default Value	Description
<code>samesite.enable</code>	<code>false</code>	Specifies whether the SameSite=None and Secure cookie attributes are set during screen recording playback from the SpeechMiner browser application.

Configure Passwords

Important

- In a Linux or Windows environment, RCS supports reading the RCS keystore password from an environment variable instead of from the configuration file. When both are available, the environment variable takes precedence.
- **RCS_KEYSTORE_PASSWORD** - maps to the existing configuration parameters `keystorepassword` and `keypassword` in the RCS properties file. When specified the same password is used for both parameters.

In a Windows environment only, the Recording Crypto Server (RCS) can store the password in the Windows Vault instead of in the `rcs.properties` file.

For example, run the following commands for the Recording Crypto Server located at `<Recording Crypto Server Directory>\scripts\powershell`:

Command to store: `encryptPassword.bat [-store <path to credentials store>] -password <password>`

Command to start RCS: `startRCS.bat [-store <path to credentials store>] -rcs <command to start RCS>`

For example:

```
startRCS.bat -store C:\GCTI\RecordingCryptoServer\rcs.secret -rcs java %JAVA_OPTS%  
-jar rcs.war -host host1.example.com -port 8888 -app RCS_Application
```

where:

- **host1.example.com** is the host for the Configuration Server.

- **8888** is the port for the Configuration Server.
- **RCS_Application** is the RCS application object.

Important

If the command `<path to credentials store>` contains a space, the path must be enclosed with quotation marks ("").

Configuring Archiving

The Recording Crypto Server provides support for automatic archiving of recordings that are older than a predefined time.

Important

Genesys recommends that the Media Lifecycle Management (MLM) functionality, which provides more flexible backup and purging rules, be used instead (see [Media Lifecycle Management](#)). New features, such as protecting recordings from deletion, are not supported with the Recording Crypto Server archiving mechanism.

To configure archiving, set the following options:

1. In the **[general]** section, set the **archive.block-size** option to the number of recordings RCS will fetch for archiving. The valid value ranges from 100 to 10000 and the default value is 5000. This option is used to verify that RCS does not run out of memory when it fetches all of the recordings at one time for archiving.

Important

Genesys recommends setting the RCS maximum Java heap size to no less than 1024 MB when **archive.block-size** is 5000. This setting enables you to avoid RCS running out of memory. Increase the maximum Java heap size accordingly when you increase the **archive.block-size**. To set the maximum Java heap size for RCS, add the **JVM** option (`-xmx1024m`), to the RCS start script.

2. On the Annex tab of each Tenant (including the Environment Tenant), in the **[recording.archive]** section, set the following parameters:

Parameter Name	Default Value	Description
interval	1	Specifies how often, in days, the archiving process runs.
retentiontime	60	Specifies how long, in days, to keep the recordings before archiving them.

Parameter Name	Default Value	Description
speechminerurl	https://host/interactionreceiver	Specifies the SpeechMiner URL where the recording metadata is stored.
user	archiveuser	Specifies the SpeechMiner username used to authenticate the SpeechMiner database.
password	changeit	Specifies the SpeechMiner password that is used to authenticate the SpeechMiner database.
outputfolder	archive	Specifies the destination folder where the archived recordings are stored.
speechminer-trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to SpeechMiner Interaction Receiver. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. RCS will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to SpeechMiner Interaction Receiver in the Configuring Transport Layer Security (TLS) Connections (Optional) section.

Important

Genesys does not recommend using a Network driver for Recording Crypto Server archive output. Therefore, set output to be a physical hard drive on the same machine.

Configuring High Availability

The Recording Crypto Server provides High Availability (HA) support to multiple Recording Crypto Server instances accessed through a load balancer. In this mode, all Recording Crypto Server instances use the same keystore file accessed through a network share, and are accessed through a single URL that utilizes the load balancer. To configure HA:

1. Set the Redundancy Type to Hot Standby on each Recording Crypto Server application instance. This setting enables logic for coordinated access to a shared keystore file.

2. Create a network share for the keystore file and set the **keystorepath** parameter in the Recording Crypto Server local configuration file to point to this file. Ensure that each Recording Crypto Server instance has read and write access to the keystore file.
3. Set the Recording Crypto Server URL parameter of the SpeechMiner application to the load balancer URL of Recording Crypto Server. If Genesys Administrator Extension is to be configured with a tenant specific URL for Recording Crypto Server, set this to the URL of the load balancer.
4. Create a Recording Crypto Server Cluster application using the `recording_crypto_850` application template, and set the following parameters:
 - On the General tab:
 - Application Name—The name of the cluster (for example, `RCS_Cluster`).
 - Working Directory—A period ".".
 - Command Line—A period ".".
 - Command Line Arguments—A period ".".
 - Host—The name of the host that the load balancer is installed on. This host must be in the configuration database.
 - On the Ports tab:
 - Configure the port by following the instructions provided in the [Configure HTTP / HTTPS Port](#) section.
5. Add a connection in the Genesys Administrator Extension application to the Recording Crypto Cluster application.

Important

For RCS HA configuration, each RCS instance operates in primary mode. The Backup Server setting on the Server Info tab of each RCS application should be set to None.

Example Load Balancer Configuration

The following is example configuration for the Apache load balancer. The details of setting up the required Apache modules are not shown. The load balancer setup must include "session sticky" so that a session that starts on a particular balancer member continues to be directed to the same member. This is achieved in the example below using the **route** and **stickysession** parameters. The **route** value must be set to the application name of the Recording Crypto Server instance, where " " characters in the name are replaced with the `_` character. For example, if the application name is **RCS 1**, set the **route** value to `RCS_1`.

```
<Proxy balancer://rcscluster>
BalancerMember https://rcshost1:port/rcs disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember https://rcshost2:port/rcs disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcscluster
```

If High Availability mode is not to be used, set the Recording Crypto Server's application Redundancy

Type to Not Specified. For this mode, the keystore file can be located on the local file system, a network share is optional.

Configuring an HTTP / HTTPS Port

To configure a port, follow these steps:

1. Log onto Genesys Administrator Extension (GAX).
2. In the GAX **Configuration** tab, choose **Environment**. Then, click **Applications** and select Recording Crypto Server application.
3. Go to the **Ports** tab in the Recording Crypto Server application.
4. Add a port or edit the existing one by entering values in the fields, **Port ID** and **Communication Port**. Note that there must be only one port.

You can configure either an unsecured port or a secured port based on your requirement.

Configuring an Unsecured (HTTP) Port

1. Enter the value `http` in the **Connection Protocol** field.
2. Enter the value `unsecured` in the **Listening Mode** field.
3. Leave the other fields empty and click **Save**.

Configuring a Secured (HTTPS) Port

1. Leave the **Connection Protocol** field empty.
2. Enter the value `secured` in the **Listening Mode** field. This sets the value `tls=1` in the **Transport Parameters** field automatically.
3. If you are setting up Mutual TLS, add `tls-mutual=1` to the **Transport Parameters** field.
4. Configure the secure port parameters at the appropriate level, as follows:

Important

If the protocol is set to `https` or left blank, a TLS server certificate and private key must be configured. This is done using the common method for Genesys applications as documented in the [Genesys Security Deployment Guide](#). The certificate and private key can be configured in the host object, the application object, and the application port entry for HTTPS.

- Host Level
 - a. In the GAX **Configuration** tab, choose **Environment** and click **Hosts**.
 - b. Click on the host object on which the server is running and enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
 - c. Restart the Recording Crypto Server.

- Application Level
 - a. In the GAX **Configuration** tab, choose **Environment** and click **Applications**.
 - b. In the **General** tab of the Recording Crypto Server application object, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
 - c. Restart the Recording Crypto Server.
- Port Level
 - a. In the GAX **Configuration** tab, choose **Environment** and click **Applications**.
 - b. In the **Ports** tab of the Recording Crypto Server application object, click on the port that you created earlier and enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
 - c. Restart the Recording Crypto Server.

Configuration of certificates at the port level has precedence over the application level, which has precedence over the host level. The private key PEM file must be in PKCS8 format. This can be achieved using the following openssl command:
openssl pkcs8 -topk8 -nocrypt -in private_keyfile.pem -inform PEM -out private_keyfile_pkcs8.pem

For more information on securing connections, refer to the [Genesys Security Deployment Guide](#).

Configuring the Connection to the Primary Configuration Server

To work with Configuration Server High Availability, the Recording Crypto Server (RCS) requires a connection to the primary Configuration Server application. For information on how to set this connection, see the [Framework 8.5.1 Management Layer User's Guide](#).

RCS supports an Advanced Disconnect Detection Protocol (ADDP) connection to the Configuration Server. To enable ADDP, perform the following:

- Add the Configuration Server to the RCS Connections tab.
- Specify the connection protocol as ADDP.
- Configure remote and local timeouts, valid values are 0-3600, where 0 means no timeout.
- Specify the required trace mode, either Local, Remote, or both.

For additional details, see the Advanced Disconnect Detection Protocol page in the [Framework 8.5.1 Deployment Guide](#).

Important

- You will see log messages about ADDP activity in the RCS logs despite switching ADDP Trace Mode to **Trace Is Turned Off** or **Trace On Server Side**. This is due to the underlying libraries handling ADDP protocol functionality.
- ADDP debug logging can be suppressed by modifying the **suppress-debug-loggers** value in the **[log]** section of the RCS configuration to contain:

```
com.genesyslab.platform.commons.connection.interceptor.AddpInterceptor  
, com.genesyslab.platform.commons.timer.impl.SchedulerImpl
```

- Genesys Advanced Disconnect Detection Protocol (ADDP) will appear in the **[log]** section of the Configuration Server log files when **verbose=all**.

Configuring Log Output

The Recording Crypto Server supports the Genesys Management Framework log configuration. For information on how to set up log output appropriate for your Recording Crypto Server application, see the Common Log Options section of the [Framework 8.5.1 Configuration Options Reference Manual](#).

Configuring the Connection to Message Server

The Recording Crypto Server must have a connection to the Message Server application to enable central auditing and alarming. For information on how to set this connection, see the [Framework 8.5.1 Management Layer User's Guide](#).

Configuring Transport Layer Security (TLS) Connections (Optional)

Configuring TLS connection to Interaction Recording Web Services (Web Services)

1. Set up TLS on Interaction Recording Web Services. For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[htcc]** section of the Recording Crypto Server configuration file, set the `baseurl` parameter to use https.
3. In the **[htcc]** section of the Recording Crypto Server configuration file, configure the **trusted-ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted-ca** to true.
 - If the TLS certificate being used by RWS is a self-signed certificate, set **trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted-ca** to false. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime

Environment 8 versions disallow MD5 signatures for certificates.

Important

The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to SpeechMiner Interaction Receiver

1. Set up TLS on SpeechMiner Interaction Receiver. For more information, see [SpeechMiner Server-Side Configuration](#).
2. On the **Annex** tab of each Tenant (including the Environment Tenant), in the **[recording.archive]** section, set the `speechminerurl` parameter to use `https`.
3. In the **[recording.archive]** section, configure the `speechminer-trusted-ca` parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **speechminer-trusted-ca** to `true`.
 - If the TLS certificate is a self-signed certificate, set **speechminer-trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and TLS is used only for encrypted transmission, set **speechminer-trusted-ca** to `false`. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 8 versions disallow MD5 signatures for certificates.

Important

The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to Message Server

1. Set up TLS on Message Server. For more information, see [Securing Core Framework Connections](#) section in the *Genesys Security Deployment Guide*. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. To connect to the secure TLS port, see [Configuring a Secure Client Connection to Other Genesys Servers](#)

section in the *Genesys Security Deployment Guide*.

3. In the properties of the **Connection** table, configure the **trusted-ca** parameter as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted-ca** to true.
- If the TLS certificate is a self-signed certificate, set **trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and TLS is used only for encrypted transmission, remove the **trusted-ca** parameter from the configuration. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 8 versions disallow MD5 signatures for certificates.

Important

The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to Configuration Server

1. Set up TLS on the Configuration Server. For more information, see [Configuring TLS on Configuration Server](#) in the *Genesys Security Deployment Guide*. Refer to [Genesys Security Deployment Guide](#) to acquire TLS certificates and private keys.
2. In the command line arguments of start information in the RCS application properties, change the port to use the Configuration Server Auto-Detect port.

For more information about the Recording Crypto Server options, see the [Genesys Interaction Recording Options Reference](#).

Deploying the Recording Plug-in for GAX

Installing Recording Plug-in for GAX

Prerequisites

- For Recording Plug-in for GAX version 8.5.500.05 (or higher):
 - Required software: Genesys Administrator Extension 9.0.107.x or later.
 - Required software: **Interaction Recording Web Services** 8.5.205.69 (or higher) instance where the Recording Lifecycle Scheduler settings and Screen Recording Certificates are updated.
 - Required software: **Recording Crypto Server** 8.5.095.22 (or higher) instance to store the Recording Certificates.
- For Windows, keep the full path of your Recording Plug-in installation directory ready.
- For Linux, keep the full path of your Genesys Administrator Extension installation directory ready.
Keep the full path of your Recording Plug-in installation directory ready.

Important

The Recording Plug-in template XML file includes the role privilege data that must be imported into Genesys Administrator Extension.

The full path of the Recording Plug-in installation directory and the full path of the GAX installation directory should be different.

Installing the Plug-in

To install the Plug-in:

1. Install the Recording Plug-in IP.
 - For Windows, the file is located at <IP plugin directory>/setup.exe.
 - For Linux, the file is located at <IP plugin directory>/install.sh.
2. Perform one of the following as required:
 - For Recording Plug-in for GAX version 8.5.500.05 (or higher), copy the gax-rcs-8.5.5xx.xx.jar file from the installed Recording Plug-in folder to GAX_ROOT/webapp/WEB-INF/lib directory.
For example, copy gax-rcs-8.5.5xx.xx.jar from C:\Program Files\GCTI\RecPluginGAX64 to C:\Program Files\GCTI\Genesys Administrator Extension\webapp\WEB-INF\lib.
 - For Recording Plug-in for GAX version 8.5.097.61 (or lower), copy the gax-rcs-8.5.0xx.xx.jar file from the installed Recording Plug-in folder to GAX_ROOT/plugin-ins directory.

For example, copy `gax-rcs-8.5.0xx.xx.jar` from `C:\Program Files\GCTI\RecPluginGAX64` to `C:\Program Files\GCTI\Genesys Administrator Extension\plug-ins`.

3. Restart Genesys Administrator Extension.
4. Import the metadata to Genesys Administrator Extension:
 - Log in to GAX.
 - From the top menu, choose **Configuration > Application Templates**.
 - Click **New**.
 - Click **Import Metadata**.
 - Click **Choose File** and choose the `recording_plugin_855.xml` file located in the **Templates** folder of Recording Plugin IP.
 - Click **OK**.
 - Provide a unique name for the template.
 - Enter the Recording Plug-in version that is defined in `recording_plugin_855.xml`.
 - Choose **Type** as **Genesys Administrator Server**.
 - Click **Save**.
5. Upload and deploy the new **SPD**.

Installing the Plug-in via GAX Installation Package UI (Discontinued after GIR 8.5.224.00)

Important

The Recording Plug-in template XML file includes the role privilege data that must be imported into Genesys Administrator Extension. This data is imported when the IP and template files are imported into Genesys Administrator Extension.

Installing the Plug-in from within Genesys Administrator Extension

Prerequisites

- Be prepared to enter the directory path to an installation directory, or to a zipped file.
- Required software: Genesys Administrator Extension 8.1.4 or later.

To install the Plug-in:

1. Select **Installation Packages** from the **Administration** menu.
2. Click the "plus" icon (+) at the upper right of the **Installation Packages** window. The **Software Installation Wizard** dialog appears to the right of the current window, offering these **Import Type**

Selection choices as radio buttons:

- Installation Package Upload (includes templates)
- Installation Package Upload (template uploaded separately)
- UNC Path to Mounted CD or Directory
- UNC Path to an Existing Administrator Repository
- UNC Path to Zipped IPs from Support

3. Select the radio button that matches your installation source and click the Next button.

4. The next dialog will request input according to your choice in the previous step:

- Installation Package Upload (includes templates) requires you to choose a zipped IP file.
- Installation Package Upload (template uploaded separately) requires you to select a zipped IP file, an XML template and an APD template (all three).
- Each of the three choices that begin with UNC Path requires a directory path that you may type or paste into the entry field. You may see a request to correct an error; type or paste your correction. When GAX is ready to install, the Finish button will be enabled.

5. Click the Finish button and wait for the upload to complete. When you see the message, Import has started. You may now close this wizard, close the Software Installation Wizard dialog by clicking the Close button at the bottom right or the X icon at the top right. The Plug-in is ready to install.

6. Select the item that you imported from the Installation Packages window. A dialog with that title appears to the right.

7. The Genesys Interaction Recording Plug-in for GAX dialog offers these actions:

- Download—Downloads the installation package to your computer.
- Delete—Erases the IP.
- Copy to Tenants—Copies the IP to the tenant(s) that you specify. Select a tenant and click Finish.
- Deploy Profile: install—Displays the IP Deployment Wizard start dialog. All following steps in this procedure are the result of this choice.

8. Click Next to display a list of host computers for possible installation. Select one or more hosts for installation using the check box to the left of each host name, then click Next.

9. At the Application Parameters dialog, complete these fields:

- Application name for host
- Tenant Name
- App port
- Primary Configuration Server
- Backup Configuration Server
- Skip IP Re-install

Notes:

- Click the Information (i) icon to the right of each field title, for tool tip help.
- A red * indicates a mandatory entry.

- Click Next when you have completed all mandatory fields.
10. Perform this step depending on the OS:
 - For Windows, at the Installation Parameters (`silent.ini`) dialog, complete the IPCommon: InstallPath field. This is the path where the IP binaries will be extracted to, and the directory must exist on the machine. The default answer offered is `C:\genesys\GCTI\`.
 - For Linux, at the Installation Parameters (`silent.ini`) dialog, complete the following steps:
 - a. The IPCommon: InstallPath field. This is the path where the IP binaries will be extracted to, and the directory must exist on the machine. The default answer offered is `/home/genesys/GCTI`.
 - b. The RecPluginGAX: GAX_Directory field, which corresponds to the installation root folder for the GAX installation. The Recording Plugin for GAX jar file should be placed in `<GAX_ROOT>/webapp/WEB-INF/lib` directory.
 11. At the Deployment dialog, verify that the answers you gave are all correct. If they are correct, click Finish and wait for the installation to complete.
 12. Restart Genesys Administrator Extension.

Upgrading the Plug-in

Using GAX 8.1.4

If you are using Genesys Administrator Extension version 8.1.4, perform the following steps:

Prerequisites

- The previous version of the Plug-in must be uninstalled.
- Be prepared for these information requests and choices:
 - You will need the full path to your Genesys Administrator Extension installation.
 - You will either confirm the default installation directory, or enter a new one.
 - If the target installation directory is populated, you will choose an action:
 - Back up all files in the directory.
 - Overwrite only the files contained in this package.
 - Wipe the directory clean.
- Perform Step 4 of the **Installing the Plug-in** section. This is required to import the Plug-in metadata to Genesys Administrator Extension.

1. Stop Genesys Administrator Extension.

2. Run the installation executable:

- For Windows, this file is `<IP plugin directory>/setup.exe`.
- For Linux, this file is `<IP plugin directory>/install.sh`.

- Copy gax-rcs-8.5.xxx.xx.jar from the installed Recording Plug-in folder to GAX_ROOT/plugin-ins directory.
For example, copy gax-rcs-8.5.097.57.jar from C:\Program Files\GCTI\RecPluginGAX64 to C:\Program Files\GCTI\Genesys Administrator Extension\plugin-ins.
- Start Genesys Administrator Extension.
- Upload and deploy the new **SPD** file.

Using GAX 9.0.107.x (or higher)

If you are using Genesys Administrator Extension version 9.0.107.x or later, perform the following steps:

Prerequisites

- The previous version of the Plug-in must be uninstalled:
 - For Windows, use the **Control Panel** to uninstall the Plug-in. You must manually remove the Plug-in gax-rcs-8.5.5xx.xx.jar file from the GAX_ROOT\webapp\WEB-INF\lib directory.
 - For Linux, manually remove the Plug-in gax-rcs-8.5.5xx.xx.jar file from the GAX_ROOT/webapp/WEB-INF/lib/ directory.
- Be prepared for these information requests and choices:
 - You will need the full path to your Genesys Administrator Extension installation.
 - You will either confirm the default installation directory, or enter a new one.
 - If the target installation directory is populated, you will choose an action:
 - Back up all files in the directory.
 - Overwrite only the files contained in this package.
 - Wipe the directory clean.
- Perform Step 4 of the **Installing the Plug-in** section. This is required to import the Plug-in metadata to Genesys Administrator Extension.

1. Stop Genesys Administrator Extension.
2. Delete the previous version of the plug-in .jar file (for example, gax-rcs-8.5.5xx.xx.jar) from the GAX_ROOT/plugin-ins/ directory.
3. Copy the gax-rcs-8.5.5xx.xx.jar file from the installed Recording Plug-in folder to GAX_ROOT/webapp/WEB-INF/lib directory.
For example, copy gax-rcs-8.5.5xx.xx.jar from C:\Program Files\GCTI\RecPluginGAX64 to C:\Program Files\GCTI\Genesys Administrator Extension\webapp\WEB-INF\lib.
4. Start Genesys Administrator Extension.
5. Upload and deploy the new **SPD** file.

Configuring the Plug-in

For the Recording Plug-in connection to the Recording Crypto Server in the Genesys Administrator Extension application connections, add a connection to the Recording Crypto Server application. If the Recording Crypto Server is setup in HA mode where there are multiple Recording Crypto Server instances behind a load balancer, set this connection to the **Recording Crypto Server cluster application**.

Configure for Screen Recording

Single-Tenant Environment

1. Using Genesys Administrator Extension, navigate to **Configuration > Applications > <Genesys Administrator Extension Application Object> > Application Options**.
2. Under the **rccs** section, set the **htcc_base_url** parameter to the Interaction Recording Web Services server URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; `http://<Interaction Recording Web Services IP Address>:8081`

Multi-Tenant Environment

1. Using Genesys Administrator Extension, navigate to **Configuration > Tenant> <Tenant Object> > Options**.
2. Under the recording section, set the **htcc_base_url** parameter to the Tenant-specific Interaction Recording Web Services load balancer URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; `http://<Interaction Recording Web Services IP Address>:8081`

Important

If this parameter does not exist or has an empty value, the Recording Plug-in retrieves the Interaction Recording Web Services (Web Services) base URL from the **htcc** section of the configured Recording Crypto Server parameters.

Configuring Transport Layer Security

1. Review the documentation for TLS support for GAX at **Transport Layer Security (TLS)**.
2. Configure TLS for Interaction Recording Web Services (see **Configuring TLS on the Server-Side for Interaction Recording Web Services**) and Recording Crypto Server (see **Configure HTTP**).
3. Add a CA certificate to the trust store by executing the following command line:

```
keytool -importcert -file <root_ca certificate of rcs/rws> -keystore keystore
```
4. Configure the trust store location for GAX by updating the Java environment. For example, on Linux or Windows you can configure this by adding the following lines to the **setenv.sh** or **setenv.bat** script, respectively:

```
set JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore="<path to keystore file which is generated using above command>"
set JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword="<password>"
```

5. Enable Transport Layer Security (TLS) certificate validation by using the following parameters:

- **trusted_ca_rcs** - Configures whether or not the certificate will be validated when making a secure outbound connection to Recording Crypto Server (RCS). This parameter can be configured in the **[rcs]** section of the GAX application object or in the **[recording]** section of a specific tenant object. If this parameter is configured in both sections, then the tenant level configuration takes higher priority than the application level. Valid values are `true` or `false`. This parameter is optional, and defaults to `true`.
- **trusted_ca_rws** - Configures whether or not the certificate will be validated when making a secure outbound connection to Interaction Recording Web Services (RWS). This parameter can be configured in the **[rcs]** section of the GAX application object or in the **[recording]** section of a specific tenant object. If this parameter is configured in both sections, then the tenant level configuration takes higher priority than the application level. Valid values are `true` or `false`. This parameter is optional, and defaults to `true`.

When validation is disabled, all the certificates are not validated, thereby ignoring the following:

- Expired certificates
- Certificates with missing chain
- No trusted CA installed in case of self signed certificate

Configure Roles and Privileges

To configure the roles and privileges required for Genesys Interaction Recording, see [Configuring Access Control for GIR Users](#).

For more information about the Plug-in options, see the [Genesys Interaction Recording Options Reference](#).

Deploying Recording Processor Script

Recording Processor Script (Python 3)

Important

Voice Processor, a multi-threaded microservice based on the Node.js platform, is an alternative to the Recording Processor Script (RPS).

- For information on deploying Voice Processor, see [Deploying Voice Processor](#).
- To migrate from an existing RPS deployment to Voice Processor, see [Migrating from RPS to Voice Processor](#).
- RPS is not supported for deployments integrated with SIP Cluster. If your deployment uses SIP Cluster, you must use Voice Processor.

For new deployments, Genesys recommends using Voice Processor instead of RPS.

Prerequisites

Before installing and configuring the RPS, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.
- For Recording Processor Script 8.5.500.13 (or higher), Recording Muxer Script must be upgraded to 8.5.500.10 (or higher).

Installing Recording Processor Script

Installing on Windows

1. Install 64 bit Python 3.11.5 from the [Python](#) website. To make Python 3 to work with OpenSSL 3.0.13, follow the below steps:
 - Download `libcrypto-3.dll` and `libssl-3.dll` from the [Python Binary repository](#).

- In [python-source-folder]\DLLs, replace with the above downloaded DLL files.
2. Install the **RPS IP** with the installer.

Important

Offline installation is supported starting from RPS 8.5.500.19.

To install it in a fully offline environment, follow these steps:

1. Unzip the <RPS>\thirdparty\flit_core-3.10.1.zip file.
2. Run `py -m pip install . --no-build-isolation` from the <RPS>\thirdparty\flit_core-3.10.1 directory.
3. Unzip the <RPS>\thirdparty\wheel-0.45.0.zip file.
4. Run `py -m pip install . --no-build-isolation` from the <RPS>\thirdparty\wheel-0.45.0 directory.

Also, add the flag `--no-build-isolation` for the upcoming commands when installing in an offline environment. For example, `py -m pip install . --no-build-isolation`

Important

Install the following third-party libraries in the order they appear and unzip the files in Administrator mode.

3. Unzip the <RPS>\thirdparty\more-itertools-10.1.0.zip file.
4. Run `py -m pip install .` from the <RPS>\thirdparty\more-itertools-10.1.0 directory.
5. Unzip the <RPS>\thirdparty\jaraco.functools-4.0.0.zip file.
6. Run `py -m pip install .` from the <RPS>\thirdparty\jaraco.functools-4.0.0 directory.
7. Unzip the <RPS>\thirdparty\cheroot-10.0.0.zip file.
8. Run `py -m pip install .` from the <RPS>\thirdparty\cheroot-10.0.0 directory.
9. Unzip the <RPS>\thirdparty\web.py-0.62.zip file.
10. Run `py -m pip install .` from the <RPS>\thirdparty\web.py-0.62 directory.
11. Unzip the <RPS>\thirdparty\pyparsing-3.1.1.zip file.
12. Run `py -m pip install .` from the <RPS>\thirdparty\pyparsing-3.1.1 directory.
13. Unzip the <RPS>\thirdparty\httplib2-0.22.0.zip file.
14. Run `py -m pip install .` from the <RPS>\thirdparty\httplib2-0.22.0 directory.
15. Unzip the <RPS>\thirdparty\six-1.16.0.zip file.
16. Run `py -m pip install .` from the <RPS>\thirdparty\six-1.16.0 directory.
17. Unzip the <RPS>\thirdparty\python-dateutil-2.8.2.zip file.

18. Run `py -m pip install .` from the `<RPS>\thirdparty\python-dateutil-2.8.2` directory.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).
2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `libffi-devel` (`yum install libffi-devel`).
4. Install OpenSSL.
 - For 8.5.500.11 or lower versions, install OpenSSL version 1.1.1.
 - For 8.5.500.13 or higher versions, install OpenSSL 3.0.13. Download OpenSSL 3.0.13 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-3.0.13 --openssldir=/usr/home/openssl-3.0.13 --libdir=lib no-shared`
5. Install 64 bit Python 3.11.5.
 - For 8.5.500.11 or lower versions, compile with OpenSSL 1.1.1 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
 - For 8.5.500.13 or higher versions, compile with OpenSSL 3.0.13 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-3.0.13 --enable-optimizations`
6. Install the **RPS IP** with the installer.

Important

Offline installation is supported starting from RPS 8.5.500.19.

To install it in a fully offline environment, follow these steps:

1. Unzip the `<RPS>\thirdparty\flit_core-3.10.1.zip` file.
2. Run `py -m pip install . --no-build-isolation` from the `<RPS>\thirdparty\flit_core-3.10.1` directory.
3. Unzip the `<RPS>\thirdparty\wheel-0.45.0.zip` file.
4. Run `py -m pip install . --no-build-isolation` from the `<RPS>\thirdparty\wheel-0.45.0` directory.

Also, add the flag `--no-build-isolation` for the upcoming commands when installing in an offline environment. For example, - `py -m pip install . --no-build-isolation`

Important

Install the following third-party libraries in the order they appear.

7. Untar the <RPS>/thirdparty/more-itertools-10.1.0.tar.gz file.
8. Run `python3 -m pip install .` from the <RPS>/thirdparty/more-itertools-10.1.0 directory.
9. Untar the <RPS>/thirdparty/jaraco.functools-4.0.0.tar.gz file.
10. Run `python3 -m pip install .` from the <RPS>/thirdparty/jaraco.functools-4.0.0 directory.
11. Untar the <RPS>/thirdparty/cheroot-10.0.0.tar.gz file.
12. Run `python3 -m pip install .` from the <RPS>/thirdparty/cheroot-10.0.0 directory.
13. Untar the <RPS>/thirdparty/web.py-0.62.tar.gz file.
14. Run `python3 -m pip install .` from the <RPS>/thirdparty/web.py-0.62 directory.
15. Untar the <RPS>/thirdparty/pyparsing-3.1.1.tar.gz file.
16. Run `python3 -m pip install .` from the <RPS>/thirdparty/pyparsing-3.1.1 directory.
17. Untar the <RPS>/thirdparty/httplib2-0.22.0.tar.gz file.
18. Run `python3 -m pip install .` from the <RPS>/thirdparty/httplib2-0.22.0 directory.
19. Untar the <RPS>/thirdparty/six-1.16.0.tar.gz file.
20. Run `python3 -m pip install .` from the <RPS>/thirdparty/six-1.16.0 directory.
21. Untar the <RPS>/thirdparty/python-dateutil-2.8.2.tar.gz file.
22. Run `python3 -m pip install .` from the <RPS>/thirdparty/python-dateutil-2.8.2 directory.

Important

- GIR does not support direct upgrade of RPS from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip install` command as mentioned above.

Configuring Recording Processor Script

This section describes how to configure the Recording Processor Script for your environment.

Configuring High Availability

Recording Processor Cluster

RPS now provides High Availability support using multiple instances of RPS (all active). These active/active instances must be accessed through an HA proxy or load balancer. In this mode, each RPS is responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner based on the load it receives. Each Recording Processor is responsible for fetching metadata from all ICON DB Servers. Media Control Platform

(MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In each Recording Processor's **rpconfig.cfg** configuration file, in the **[processing]** section, set the following options:
 - **get_from_http_before_posting** = 1
 - **mode** = active
2. Ensure that all Recording Processor instances have the *same* network related configuration.

Important

Genesys recommends that multiple Recording Processor instances be deployed on a single host to optimize the available CPU and take advantage of parallel processing. Multiple Recording Processor instances can then be deployed on other hosts as needed.

3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the **Recording Processor URI** parameter to the load balancer's URL.
4. Configure the load balancer to balance traffic to the Recording Processor instances.

The following is an example configuration section that is needed for setting up an Apache load balancer for a three-instance Recording Processor cluster.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
    BalancerMember http://<IP address of the active1 Recording Processor server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the active2 Recording Processor Server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the active3 Recording Processor Server>:<active
Recording Processor port>
</Proxy>
```

Important

SpeechMiner version 8.5.2 or later is required for the Recording Processor cluster support to work properly.

Recording Processor Script Active/Backup HA

RPS can also provide High Availability support by using two RPS instances (active and backup) accessed through an HA proxy or load balancer in failover mode. In this mode, the active RPS is always responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner, and the backup instance is responsible for receiving and temporarily storing metadata if the active instance is unavailable. Once the active instance recovers, the balancer will direct clients to the active instance, and the backup instance will

send any stored data to the active instance for metadata processing. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In the active Recording Processor's **rpconfig.cfg** configuration file located in the **[processing]** section, set the **mode** parameter to active.
2. In the backup Recording Processor's **rpconfig.cfg** configuration file:
 - Set the **mode** parameter to backup.
 - In the **[processing]** section, set the **post_uri** parameter to `http://<active_rp_ip>:<active_rp_port>/api/contact-centers/%s/recordings/`.
3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the Recording Processor URI parameter to the failover load balancer's URL.
4. Configure the load balancer to direct traffic to the active Recording Processor instance first and to the backup instance if an error/failure occurs.

The following is an example configuration section that is needed for setting up an Apache load balancer in failover mode for Recording HA support.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
    BalancerMember http://<IP address of the active Recording Processor server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the backup Recording Processor Server>:<active
Recording Processor port> status=H
</Proxy>
```

For more information about how to use Genesys Administrator Extension to configure your Contact Center, see the [Genesys Administrator Extension Help](#).

Configure Passwords

Important

In a Linux or Windows environment, RPS supports reading the environment variables for password related configuration parameters in order to avoid storing the password in plain-text in the configuration file. When both are available, the environment variables take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

HTCC_PASSWORD - maps to the existing configuration parameter under the **htcc** section, password value.
AUTH_PASSWORD - maps to the existing configuration parameter under the **auth** section, password value.
CONFIG_SERVER_PASSWORD - maps to the existing configuration parameter under the **config_server** section, password value.
<ICON_ID>_DB_INFO_PASSWORD - maps to the existing configuration parameter under the

<ICON_ID>_db_info section, password value, where <ICON_ID> refers to the ICON instance listed in the icon_db_servers section.

For example, if you have VCCSIPSwitch: icon1 the environment variable that corresponds to the icon1_db_info password is icon1_DB_INFO_PASSWORD.

In a Windows environment only, the Recording Processor Script (RPS) can store passwords in the Windows Vault instead of in the **rpconfig.cfg** file or requiring the use of environment variables.

For example, run the following command for the Recording Processor Script credentials located at <Recording Processor Directory>\rp. This command will prompt the user to enter valid values for the password/key configuration parameters and stores the passwords in the encrypted file named **rp.secret**:

Command to store:

```
encryptPassword.bat -password <password_string>
```

Where <password_string> is a comma-delimited series of key/value pairs, use the format <environment variable name 1>=<environment variable value 1>,<environment variable name 2>=<environment variable value 2>,<environment variable name 3>=<environment variable value 3>, and so on. Note that space is not allowed in <password_string>.

For example:

```
encryptPassword.bat -password "HTCC_PASSWORD=somepassword1, AUTH_PASSWORD=somepassword2, CONFIG_SERVER_PASSWORD=somepassword3, ICON1_DB_INFO_PASSWORD=somepassword4, ICON2_DB_INFO_PASSWORD=somepassword5"
```

Important

Passwords used with this command cannot contain a comma or an equals sign.

Configure the Configuration Server Connection

To configure the Configuration Server connection, set the following parameters in the **[config_server]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
application_name	Empty	Specifies the name of the RPS application object in the Configuration Server, when using RPS as a third party server application.
hostname	<ip address>	Specifies the IP address of the primary Configuration Server.
port	2020	Specifies the port of the primary

Parameter Name	Default Value	Description
		Configuration Server.
username	default	Specifies the Configuration Server username.
password	password	Specifies the Configuration Server password. Note: The password can be overridden by the CONFIG_SERVER_PASSWORD environment variable.
backup_host	Empty	Specifies the IP address of the backup Configuration Server.
backup_port	Empty	Specifies the backup port of the backup Configuration Server.

Important

Recording Processor Script does not support a secure connection to the Configuration Server.

Configuring the Server Port

In the **[rp_server]** section of the **rpconfig.cfg** file, set the **port** parameter.

Important

You can also set the "port" parameter using the command line with the --port command line argument. The command line argument takes precedence over the configuration file value.

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri	http://<Web Services IP>:<Web Services Port>	Specifies the Base URI for accessing the Interaction Recording Web Services (Web Services) API.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services)

Parameter Name	Default Value	Description
		account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note: The password can be overridden by the HTCC_PASSWORD environment variable.</p>

Each Interaction Recording Web Services (Web Services) instance must have a region associated with it. Set the region parameter in the [metadata] section of the `rpconfig.cfg` file to match the region associated with Interaction Recording Web Services (Web Services) instance set to receive the Recording Processor's metadata.

Configuring Cross-Site Request Forgery (CSRF) Protection

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled, set the following parameter in the **[htcc]** section of the **rpconfig.cfg** file:

- **csrfp** = 1

Configuring the Connection to SpeechMiner

To configure the SpeechMiner Connection:

1. In the IVR Profile, set the recording destinations to point to the SpeechMiner interaction receiver:
 - a. Login to Genesys Administrator Extension, and navigate to **Configuration > System > Configuration Manager**.
 - b. Under **Voice Platform**, select **Voice Platform Profiles**.
 - c. Click on the IVR Profile for which you want to set the recording destination.
 - d. Select the **Recording** tab.
 - e. In the **SpeechMiner Interaction Receiver** field, enter the URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile. For example, `https://<SpeechMiner IP>/interactionreceiver`.
 - f. In the **SpeechMiner Interaction Receiver Authorization Header** field, enter the authorization information (username:password) required to connect to the SpeechMiner service used by the RPS. For example, `user:password`.

Important

The values of these options must match the corresponding configuration options in the SpeechMiner system.

Configuring Failed Message Files

The Recording processor can backup messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner. These files are located in the **<recording processor dir>\failed** folder.

In the **rpconfig.cfg** configuration file, add the following parameter:

```
[processing]
backup_failed_metadata = 1
```

Configuring the Agent Hierarchies

Recording Processor Script uses the agent hierarchy information to set the access control information for recordings within the recording metadata. Refer to [Access Control for Genesys Interaction Recording Users](#) to configure this appropriately.

Configuring Basic Authorization

In the **rpconfig.cfg** configuration file, set the following parameters:

```
[auth]
# Basic Authentication username and password. Set username blank to disable.
username = rp_username
password = rp_password
```

Important

- The username and password must match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.
- The password can be overridden by the **AUTH_PASSWORD** environment variable.

Configuring After Call Work

Recording Processor can collect After Call Work (ACW) customized data from ICON.

In the **rpconfig.cfg** file, in the **[processing]** section, add the following parameters:

- **enable_acw**—Set it to 1.
- **acw_threshold_minutes**—Set it to the maximum time to wait for the customized attached data.

Important

- If Call Customized Attached Data is still not available in the ICON database after *acw_threshold_minutes*, the RPS will stop collecting customized data for this recording and write it to the database.
- If *enable_acw* is set to 0, ACW customized data will not be included.
- If disposition code is required in the metadata, you must set **enable_acw** and **acw_threshold_minutes** using Recording Processor configuration. The disposition code is part of the user data collected during ACW. For this reason, **enable_acw** must be enabled in the Recording Processor. If it is not enabled, the data will not be collected. If the disposition code must be collected from the Recording Processor, configure the following to include the disposition code for recording:

```
[processing] enable_acw=1
[metadata] acw_threshold_minutes=5.
```

Where 5 is the maximum time (in minutes) to wait for the disposition code. In the ICON configuration, the **EventData** parameter in the **custom-states** section, must include `char,DispositionCode` and **store-event-data** must be set to `conf` to collect the attached data:

```
[custom-states] store-event-data=conf      EventData=char,DispositionCode
```

For additional information, refer to the [ICON Deployment Guide](#).

Configuring ICON for Recording Processor

Important

When configuring Recording Processor to connect to a primary and backup ICON Database in HA mode, two separate DB Servers must be used. The DB Servers must run in an active/active pair mode.

To configure ICON, edit the **rpconfig.cfg** configuration file as follows:

1. Configure the switches:

Add a configuration option for each switch name under the **[icon_db_servers]** section. You can specify more than two ICON databases per SIP Switch configuration. For example:

```
[icon_db_servers]
SIP_Switch1: icon1
SIP_Switch2: icon2, icon2Backup
SIP_Switch3: icon3, icon4, icon5, icon6
```

Important

In the above example, **SIP_Switch3** has 4 ICON databases. The Recording Processor Script (RPS) keeps track of the ICON database instance currently used. If the current database instance becomes unavailable, RPS will

attempt the operation in the next database.

The configuration option name must match the exact name of the switch as configured in the Genesys configuration. The primary and backup ICON names must be unique, but do not have to match anything in the Genesys configuration.

2. Configure the ICON Connection Settings:

- For each unique ICON specified in the first step, create a new section using the following syntax: `<ICON_ID>_db_info`, where `<ICON_ID>` corresponds to the values defined in the **[icon_db_servers]** section above.
- **dbengine** must be `mssql`, `oracle`, `db2`, or `postgres`.
- **dbserver_host** and **dbserver_port** specify the host and port information for the Genesys DB Server.
- **dbms** specifies the host where the database resides.

The following is an example using the values for **SIP_Switch1** and **SIP_Switch2** from step 1:

```
[icon1_db_info]
dbserver_host = vm221.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2_db_info]
dbserver_host = vm222.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2Backup_db_info]
dbserver_host = vm223.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon_oracle_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host/Oracle SID>
```

```
dbengine = oracle

[icon_postgres_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname = <database name>
dbms = <database host>
dbengine = postgres

[icon_db2_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host>
dbengine = db2
```

Important

- For Oracle or DB2 implementations, the **dbname** parameter must be left blank or empty.
- The password can be overridden by the **<ICON_ID>_DB_INFO_PASSWORD** environment variable.

In the example above, the RPS will use the connection properties in section **[icon1_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch1. The RPS will use the connection properties in section **[icon2_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch2. In the case of SIP_Switch2, the RPS will use the connection settings in **[icon2Backup_db_info]** if the primary ICON (icon2) is unavailable when recording metadata is being processed.

Configure how to Filter Metadata from ICON

The Recording Processor supports the ability to filter specific attached data fields (based on the key name), such as attached data and After Call Work (ACW) customized data retrieved from the ICON database. This support prevents specific metadata from reaching additional GIR related components (for example, SpeechMiner).

The following two sections describe how to:

- **Filter attached data.**
- **Filter ACW.**

Important

- Verify that the following items are not removed from the filter. Removing these items may cause errors in GIR:
 - GRECORD_PARTITIONS
 - GRECORD_PROGRAM
 - GSIP_REC_FN
- When running SpeechMiner, you must include Workspace Web Edition (WWE) in the attached_data_filter and acw_custom_data_filter Recording Processor configuration values. For example:


```
[filter]
attached_data_filter=^ORSI:|^WWE
acw_custom_data_filter=^ORSI:|^WWE
```

Filter Attached Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **attached_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out attached data whose key name matches the pattern.

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **attached_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude key names that should not be filtered out (for example, like GRECORD_PARTITIONS).

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
attached_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the default
value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Filter ACW Related Custom Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **acw_custom_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out ACW whose key name matches the pattern.

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **acw_custom_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude ACW that should not be filtered out (for example, like **GRECORD_PARTITIONS**).

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
acw_custom_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the
default value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Configuring SSL for Recording Processor

To configure SSL:

Configure HTTPS on the Primary Recording Processor Server

1. Create a self-signed certificate and private key for the Recording Processor host. For example, on RHEL run: `openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem`
2. In the **rp_server** section of the Recording Processor's configuration file, set the following parameters:
 - **ssl_certificate**—To point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - **ssl_private_key**—To point to the private key file. For example, `cert228.pem`.
3. Give the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section [Genesys Voice Platform 8.5 User's Guide](#).

4. Restart Recording Processor.

Configure the HTTPS connection to Interaction Recording Web Services (Web Services)

1. Set up HTTPS on Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). See the [Genesys Security Deployment Guide](#).
2. Get the corresponding certificate for the Interaction Recording Web Services (Web Services) server. Set the **caCertificate** option in your Interaction Recording Web Services application (see **caCertificate** if you're using a Web Services application).
3. In the **[htcc]** section of the Recording Processor configuration file, set **base_uri** parameter to use https.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection to SpeechMiner

1. Set up HTTPS on SpeechMiner. See the [Genesys Security Deployment Guide](#).
2. Set the **disable_ssl_certificate_validation** parameter in the **[speechminer]** section of the Recording Processor configuration to a value of 1.
3. Using Genesys Administrator Extension on the Recording tab of the IVR Profile, modify the SpeechMiner Interaction Receiver field use https as the protocol in the URL.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA cert file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection from the backup Recording Processor to the primary Recording Processor

1. Configure HTTPS on the primary Recording Processor.
2. Get the corresponding PEM certificate for the Web Services server.
3. In the **[processing]** section of the Recording Processor configuration file, set the **post_uri** parameter

to use https as the protocol in the URL.

4. In the **client** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS on the backup Recording Processor Server

Follow the same procedure used for the Primary Recording Processor Server using a new certificate and private key for the Backup Recording Processor's server.

Configuring the IVR Profile

Using Genesys Administrator Extension, configure the following parameters on the **Recording** tab of the IVR Profile:

1. **Recording Processor URI**—The URI that the Media Control Platform (MCP) uses to post the metadata of the audio recording after the recording is complete. For example, `http:// <Recording Processor Host>/api/contact-centers/<Contact Cente Domain Name>/recordings/`.

Important

The value for the URI must always end with a forward slash (/).

2. **SpeechMiner Interaction Receiver**—The URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile.
3. **SpeechMiner Interaction Receiver Authorization Header**—The authorization information required to connect to the SpeechMiner service used by the RPS. For example, `<SpeechMiner Webserver Username>:<SpeechMiner Webserver Password>`.

For more information, see the [Configuring GVP](#).

Configuring the Recording Processor Using Genesys Administrator Extension (Optional)

The Recording Processor uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Processor as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Processor as a "third party server" application in

Genesys Administrator Extension. For more information, see the "Using the Management Layer" section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring RPS to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest RPS.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Processor Script of type Third Party Server.
4. Create a new application (for example, myRPS) in Genesys Administrator Extension using this new application template.
5. Set the Command Line parameter (for example, C:\Python311\python.exe).
6. Set the Host parameter in the application's server info to the correct Host object.
7. Set the Working Directory parameter to the <Recording Processor Install Directory>\rp directory. For example, /opt/genesys/Recording_Processor_Script_8.5/rp/.
8. Set the Command Line Arguments parameter to the appropriate values. For example, recording_process.py --config-file=/opt/genesys/Recording_Processor_Script_8.5/rp/rpconfig.cfg.
Refer to the [Starting the Recording Processor Script](#) section for additional command line parameters
9. Make sure that LCA has permission to read and write to the Recording Processor installation directory and Recording Processor log directory.
10. Save the configuration changes.
11. Ensure that the Configuration Server parameters in the Recording Processor configuration file are set appropriately. Refer to **Configure the Configuration Server Connection** tab on this page.

Important

The Recording Processor does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

For more information about the RPS options, see [Genesys Interaction Recording Options Reference](#).

Starting the Recording Processor Script

To launch the RPS, run the following command from the <Recording Processor Install Directory>:

```
<python executable> recording_process.py --config-file=rpconfig.cfg
```

Use the following command line when you want to run multiple instances of RPS on the same machine:

```
<python executable> recording_process.py --config-file=rpconfig.cfg --id=1 --port=8889
```

For each RPS instance, assign a unique id (--id parameter) and port number (--port).

Important

- --port defines the server port opened by the RPS process.
- --id represents the suffix of the:
 - application_name in the configuration file. For example, if application_name is defined in the configuration file as **RecordingProcessorScript** and --id 2 is specified in the command line, then the application object named **RecordingProcessorScript_2** will be used to start the program.
 - log files
 - metadata json files created in the failed folder
 - database file created by the process

By default the RPS log file is stored in the working directory. This can be changed by specifying a preexisting folder in the logfile_path parameter in the log file section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingProcessor
```

Recording Processor Script (Python 3) RHEL 7

Important

Voice Processor, a multi-threaded microservice based on the Node.JS platform, is an alternative to the Recording Processor Script (RPS).

- For information on deploying Voice Processor, see [Deploying Voice Processor](#).
- To migrate from an existing RPS deployment to Voice Processor, see [Migrating from RPS to Voice Processor](#).
- RPS is not supported for deployments integrated with SIP Cluster. If your deployment uses SIP Cluster, you must use Voice Processor.

For new deployments, Genesys recommends using Voice Processor instead of RPS.

Prerequisites

Before installing and configuring the RPS, you must have the following prerequisites:

- An **Interaction Recording Web Services** 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.
- A **Recording Crypto Server** 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.

Installing Recording Processor Script

Installing on Windows

1. Install 64 bit Python 3.11.5 from the **Python** website.
2. Install the **RPS IP** with the installer. **Note:** Install the following third-party libraries in the order they appear and unzip the files in Administrator mode.
3. Unzip the <RPS>\thirdparty\more-itertools-10.1.0.zip file.
4. Run `py -m pip install .` from the <RPS>\thirdparty\more-itertools-10.1.0 directory.
5. Unzip the <RPS>\thirdparty\jaraco.functools-4.0.0.zip file.
6. Run `py -m pip install .` from the <RPS>\thirdparty\jaraco.functools-4.0.0 directory.
7. Unzip the <RPS>\thirdparty\cheroot-10.0.0.zip file.
8. Run `py -m pip install .` from the <RPS>\thirdparty\cheroot-10.0.0 directory.
9. Unzip the <RPS>\thirdparty\web.py-0.62.zip file.
10. Run `py -m pip install .` from the <RPS>\thirdparty\web.py-0.62 directory.
11. Unzip the <RPS>\thirdparty\pyparsing-3.1.1.zip file.
12. Run `py -m pip install .` from the <RPS>\thirdparty\pyparsing-3.1.1 directory.
13. Unzip the <RPS>\thirdparty\httplib2-0.22.0.zip file.
14. Run `py -m pip install .` from the <RPS>\thirdparty\httplib2-0.22.0 directory.
15. Unzip the <RPS>\thirdparty\six-1.16.0.zip file.
16. Run `py -m pip install .` from the <RPS>\thirdparty\six-1.16.0 directory.
17. Unzip the <RPS>\thirdparty\python-dateutil-2.8.2.zip file.
18. Run `py -m pip install .` from the <RPS>\thirdparty\python-dateutil-2.8.2 directory.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).

2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `libffi-devel` (`yum install libffi-devel`).
4. Install OpenSSL 1.1.1.
 - For RHEL 7:
 1. Download OpenSSL 1.1.1 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-1.1.1 --openssldir=/usr/home/openssl-1.1.1`
 2. Add OpenSSL lib path in `LD_LIBRARY_PATH`. Example command - `export LD_LIBRARY_PATH=/usr/home/openssl-1.1.1/lib:$LD_LIBRARY_PATH`
5. Install 64 bit Python 3.11.5 compiled with OpenSSL 1.1.1 from the [Python](#) website.
 - While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
6. Install the **RPS IP** with the installer. **Note:** Install the following third-party libraries in the order they appear.
7. Untar the `<RPS>/thirdparty/more-itertools-10.1.0.tar.gz` file.
8. Run `python3 -m pip install .` from the `<RPS>/thirdparty/more-itertools-10.1.0` directory.
9. Untar the `<RPS>/thirdparty/jaraco.functools-4.0.0.tar.gz` file.
10. Run `python3 -m pip install .` from the `<RPS>/thirdparty/jaraco.functools-4.0.0` directory.
11. Untar the `<RPS>/thirdparty/cheroot-10.0.0.tar.gz` file.
12. Run `python3 -m pip install .` from the `<RPS>/thirdparty/cheroot-10.0.0` directory.
13. Untar the `<RPS>/thirdparty/web.py-0.62.tar.gz` file.
14. Run `python3 -m pip install .` from the `<RPS>/thirdparty/web.py-0.62` directory.
15. Untar the `<RPS>/thirdparty/pyparsing-3.1.1.tar.gz` file.
16. Run `python3 -m pip install .` from the `<RPS>/thirdparty/pyparsing-3.1.1` directory.
17. Untar the `<RPS>/thirdparty/httplib2-0.22.0.tar.gz` file.
18. Run `python3 -m pip install .` from the `<RPS>/thirdparty/httplib2-0.22.0` directory.
19. Untar the `<RPS>/thirdparty/six-1.16.0.tar.gz` file.
20. Run `python3 -m pip install .` from the `<RPS>/thirdparty/six-1.16.0` directory.
21. Untar the `<RPS>/thirdparty/python-dateutil-2.8.2.tar.gz` file.
22. Run `python3 -m pip install .` from the `<RPS>/thirdparty/python-dateutil-2.8.2` directory.

Important

- GIR does not support direct upgrade of RPS from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip install` command as mentioned above.

Configuring Recording Processor Script

This section describes how to configure the Recording Processor Script for your environment.

Configuring High Availability

Recording Processor Cluster

RPS now provides High Availability support using multiple instances of RPS (all active). These active/active instances must be accessed through an HA proxy or load balancer. In this mode, each RPS is responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner based on the load it receives. Each Recording Processor is responsible for fetching metadata from all ICON DB Servers. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In each Recording Processor's **rpconfig.cfg** configuration file, in the **[processing]** section, set the following options:
 - **get_from_httpc_before_posting** = 1
 - **mode** = active
2. Ensure that all Recording Processor instances have the *same* network related configuration.

Important

Genesys recommends that multiple Recording Processor instances be deployed on a single host to optimize the available CPU and take advantage of parallel processing. Multiple Recording Processor instances can then be deployed on other hosts as needed.

3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the **Recording Processor URI** parameter to the load balancer's URL.
4. Configure the load balancer to balance traffic to the Recording Processor instances.

The following is an example configuration section that is needed for setting up an Apache load balancer for a three-instance Recording Processor cluster.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
  BalancerMember http://<IP address of the active1 Recording Processor server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active2 Recording Processor Server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active3 Recording Processor Server>:<active
Recording Processor port>
```

</Proxy>

Important

SpeechMiner version 8.5.2 or later is required for the Recording Processor cluster support to work properly.

Recording Processor Script Active/Backup HA

RPS can also provide High Availability support by using two RPS instances (active and backup) accessed through an HA proxy or load balancer in failover mode. In this mode, the active RPS is always responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner, and the backup instance is responsible for receiving and temporarily storing metadata if the active instance is unavailable. Once the active instance recovers, the balancer will direct clients to the active instance, and the backup instance will send any stored data to the active instance for metadata processing. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In the active Recording Processor's **rpconfig.cfg** configuration file located in the **[processing]** section, set the **mode** parameter to active.
2. In the backup Recording Processor's **rpconfig.cfg** configuration file:
 - Set the **mode** parameter to backup.
 - In the **[processing]** section, set the **post_uri** parameter to `http://<active_rp_ip>:<active_rp_port>/api/contact-centers/%s/recordings/`.
3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the Recording Processor URI parameter to the failover load balancer's URL.
4. Configure the load balancer to direct traffic to the active Recording Processor instance first and to the backup instance if an error/failure occurs.

The following is an example configuration section that is needed for setting up an Apache load balancer in failover mode for Recording HA support.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
    BalancerMember http://<IP address of the active Recording Processor server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the backup Recording Processor Server>:<active
Recording Processor port> status=H
</Proxy>
```

For more information about how to use Genesys Administrator Extension to configure your Contact Center, see the [Genesys Administrator Extension Help](#).

Configure Passwords

Important

In a Linux or Windows environment, RPS supports reading the environment variables for password related configuration parameters in order to avoid storing the password in plain-text in the configuration file. When both are available, the environment variables take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

HTCC_PASSWORD - maps to the existing configuration parameter under the **htcc** section, password value.

AUTH_PASSWORD - maps to the existing configuration parameter under the **auth** section, password value.

CONFIG_SERVER_PASSWORD - maps to the existing configuration parameter under the **config_server** section, password value.

<ICON_ID>_DB_INFO_PASSWORD - maps to the existing configuration parameter under the **<ICON_ID>_db_info** section, password value, where **<ICON_ID>** refers to the ICON instance listed in the **icon_db_servers** section.

For example, if you have VCCSIPSwitch: icon1 the environment variable that corresponds to the icon1_db_info password is icon1_DB_INFO_PASSWORD.

In a Windows environment only, the Recording Processor Script (RPS) can store passwords in the Windows Vault instead of in the **rpconfig.cfg** file or requiring the use of environment variables.

For example, run the following command for the Recording Processor Script credentials located at **<Recording Processor Directory>\rp**. This command will prompt the user to enter valid values for the password/key configuration parameters and stores the passwords in the encrypted file named **rp.secret**:

Command to store:

```
encryptPassword.bat -password <password_string>
```

Where **<password_string>** is a comma-delimited series of key/value pairs, use the format **<environment variable name 1>=<environment variable value 1>,<environment variable name 2>=<environment variable value 2>,<environment variable name 3>=<environment variable value 3>**, and so on. Note that space is not allowed in **<password_string>**.

For example:

```
encryptPassword.bat -password "HTCC_PASSWORD=somepassword1, AUTH_PASSWORD=somepassword2, CONFIG_SERVER_PASSWORD=somepassword3, ICON1_DB_INFO_PASSWORD=somepassword4, ICON2_DB_INFO_PASSWORD=somepassword5"
```

Important

Passwords used with this command cannot contain a comma or an equals sign.

Configure the Configuration Server Connection

To configure the Configuration Server connection, set the following parameters in the **[config_server]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
application_name	Empty	Specifies the name of the RPS application object in the Configuration Server, when using RPS as a third party server application.
hostname	<ip address>	Specifies the IP address of the primary Configuration Server.
port	2020	Specifies the port of the primary Configuration Server.
username	default	Specifies the Configuration Server username.
password	password	Specifies the Configuration Server password. Note: The password can be overridden by the CONFIG_SERVER_PASSWORD environment variable.
backup_host	Empty	Specifies the IP address of the backup Configuration Server.
backup_port	Empty	Specifies the backup port of the backup Configuration Server.

Important

Recording Processor Script does not support a secure connection to the Configuration Server.

Configuring the Server Port

In the **[rp_server]** section of the **rpconfig.cfg** file, set the **port** parameter.

Important

You can also set the "port" parameter using the command line with the --port command line argument. The command line argument takes precedence over the configuration file value.

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri	http://<Web Services IP>:<Web Services Port>	Specifies the Base URI for accessing the Interaction Recording Web Services (Web Services) API.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	Specifies the password used to access the Interaction Recording Web Services (Web Services) account. Note: The password can be overridden by the HTCC_PASSWORD environment variable.

Each Interaction Recording Web Services (Web Services) instance must have a region associated with it. Set the region parameter in the [metadata] section of the rpconfig.cfg file to match the region associated with Interaction Recording Web Services (Web Services) instance set to receive the Recording Processor's metadata.

Configuring Cross-Site Request Forgery (CSRF) Protection

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled, set the following parameter in the **[htcc]** section of the **rpconfig.cfg** file:

- **csrfp = 1**

Configuring the Connection to SpeechMiner

To configure the SpeechMiner Connection:

1. In the IVR Profile, set the recording destinations to point to the SpeechMiner interaction receiver:
 - a. Login to Genesys Administrator Extension, and navigate to **Configuration > System > Configuration Manager**.
 - b. Under **Voice Platform**, select **Voice Platform Profiles**.
 - c. Click on the IVR Profile for which you want to set the recording destination.
 - d. Select the **Recording** tab.
 - e. In the **SpeechMiner Interaction Receiver** field, enter the URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile. For example, `https://<SpeechMiner IP>/interactionreceiver`.

- f. In the **SpeechMiner Interaction Receiver Authorization Header** field, enter the authorization information (username:password) required to connect to the SpeechMiner service used by the RPS. For example, user:password.

Important

The values of these options must match the corresponding configuration options in the SpeechMiner system.

Configuring Failed Message Files

The Recording processor can backup messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner. These files are located in the **<recording processor dir>\failed** folder.

In the **rpconfig.cfg** configuration file, add the following parameter:

```
[processing]
backup_failed_metadata = 1
```

Configuring the Agent Hierarchies

Recording Processor Script uses the agent hierarchy information to set the access control information for recordings within the recording metadata. Refer to [Access Control for Genesys Interaction Recording Users](#) to configure this appropriately.

Configuring Basic Authorization

In the **rpconfig.cfg** configuration file, set the following parameters:

```
[auth]
# Basic Authentication username and password. Set username blank to disable.
username = rp_username
password = rp_password
```

Important

- The username and password must match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.
- The password can be overridden by the **AUTH_PASSWORD** environment variable.

Configuring After Call Work

Recording Processor can collect After Call Work (ACW) customized data from ICON.

In the **rpconfig.cfg** file, in the **[processing]** section, add the following parameters:

- **enable_acw**—Set it to 1.
- **acw_threshold_minutes**—Set it to the maximum time to wait for the customized attached data.

Important

- If Call Customized Attached Data is still not available in the ICON database after *acw_threshold_minutes*, the RPS will stop collecting customized data for this recording and write it to the database.
- If *enable_acw* is set to 0, ACW customized data will not be included.
- If disposition code is required in the metadata, you must set **enable_acw** and **acw_threshold_minutes** using Recording Processor configuration. The disposition code is part of the user data collected during ACW. For this reason, **enable_acw** must be enabled in the Recording Processor. If it is not enabled, the data will not be collected. If the disposition code must be collected from the Recording Processor, configure the following to include the disposition code for recording:
[processing] enable_acw=1
[metadata] acw_threshold_minutes=5. Where 5 is the maximum time (in minutes) to wait for the disposition code. In the ICON configuration, the **EventData** parameter in the **custom-states** section, must include char,DispositionCode and **store-event-data** must be set to conf to collect the attached data:
[custom-states] store-event-data=conf EventData=char,DispositionCode
For additional information, refer to the [ICON Deployment Guide](#).

Configuring ICON for Recording Processor

Important

When configuring Recording Processor to connect to a primary and backup ICON Database in HA mode, two separate DB Servers must be used. The DB Servers must run in an active/active pair mode.

To configure ICON, edit the **rpconfig.cfg** configuration file as follows:

1. Configure the switches:

Add a configuration option for each switch name under the **[icon_db_servers]** section. You can specify more than two ICON databases per SIP Switch configuration. For example:

```
[icon_db_servers]
SIP_Switch1: icon1
SIP_Switch2: icon2, icon2Backup
SIP_Switch3: icon3, icon4, icon5, icon6
```

Important

In the above example, **SIP_Switch3** has 4 ICON databases. The Recording Processor Script (RPS) keeps track of the ICON database instance currently used. If the current database instance becomes unavailable, RPS will attempt the operation in the next database.

The configuration option name must match the exact name of the switch as configured in the Genesys configuration. The primary and backup ICON names must be unique, but do not have to match anything in the Genesys configuration.

2. Configure the ICON Connection Settings:

- For each unique ICON specified in the first step, create a new section using the following syntax: `<ICON_ID>_db_info`, where `<ICON_ID>` corresponds to the values defined in the **[icon_db_servers]** section above.
- **dbengine** must be `mssql`, `oracle`, `db2`, or `postgres`.
- **dbserver_host** and **dbserver_port** specify the host and port information for the Genesys DB Server.
- **dbms** specifies the host where the database resides.

The following is an example using the values for **SIP_Switch1** and **SIP_Switch2** from step 1:

```
[icon1_db_info]
dbserver_host = vm221.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2_db_info]
dbserver_host = vm222.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2Backup_db_info]
dbserver_host = vm223.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon_oracle_db_info]
dbserver_host = <host>
dbserver_port = <port>
```



```
username = <username>
password = <password>
dbname =
dbms = <database host/Oracle SID>
dbengine = oracle

[icon_postgres_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname = <database name>
dbms = <database host>
dbengine = postgres

[icon_db2_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host>
dbengine = db2
```

Important

- For Oracle or DB2 implementations, the **dbname** parameter must be left blank or empty.
- The password can be overridden by the **<ICON_ID>_DB_INFO_PASSWORD** environment variable.

In the example above, the RPS will use the connection properties in section **[icon1_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch1. The RPS will use the connection properties in section **[icon2_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch2. In the case of SIP_Switch2, the RPS will use the connection settings in **[icon2Backup_db_info]** if the primary ICON (icon2) is unavailable when recording metadata is being processed.

Configure how to Filter Metadata from ICON

The Recording Processor supports the ability to filter specific attached data fields (based on the key name), such as attached data and After Call Work (ACW) customized data retrieved from the ICON database. This support prevents specific metadata from reaching additional GIR related components (for example, SpeechMiner).

The following two sections describe how to:

- **Filter attached data.**
- **Filter ACW.**

Important

- Verify that the following items are not removed from the filter. Removing these items may cause errors in GIR:
 - RECORD_PARTITIONS
 - RECORD_PROGRAM
 - GSIP_REC_FN
- When running SpeechMiner, you must include Workspace Web Edition (WWE) in the attached_data_filter and acw_custom_data_filter Recording Processor configuration values. For example:


```
[filter]
attached_data_filter=^ORSI|^WWE
acw_custom_data_filter=^ORSI|^WWE
```

Filter Attached Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **attached_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out attached data whose key name matches the pattern.

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **attached_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude key names that should not be filtered out (for example, like RECORD_PARTITIONS).

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
attached_data_filter_exception = ^RECORD_PARTITIONS$      ; (Note: this is the default
value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Filter ACW Related Custom Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **acw_custom_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out ACW whose key name matches the pattern.

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **acw_custom_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude ACW that should not be filtered out (for example, like GRECORD_PARTITIONS).

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
acw_custom_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the
default value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Configuring SSL for Recording Processor

To configure SSL:

Configure HTTPS on the Primary Recording Processor Server

1. Create a self-signed certificate and private key for the Recording Processor host. For example, on RHEL run: `openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem`
2. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—To point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228.pem`.
3. Give the self-signed certificate PEM file to any MCP client that needs to validate the certificate during

the SSL handshake. See the "Enable Secure Communication" section [Genesys Voice Platform 8.5 User's Guide](#).

4. Restart Recording Processor.

Configure the HTTPS connection to Interaction Recording Web Services (Web Services)

1. Set up HTTPS on Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). See the [Genesys Security Deployment Guide](#).
2. Get the corresponding certificate for the Interaction Recording Web Services (Web Services) server. Set the **caCertificate** option in your Interaction Recording Web Services application (see **caCertificate** if you're using a Web Services application).
3. In the **[htcc]** section of the Recording Processor configuration file, set **base_uri** parameter to use https.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection to SpeechMiner

1. Set up HTTPS on SpeechMiner. See the [Genesys Security Deployment Guide](#).
2. Set the **disable_ssl_certificate_validation** parameter in the **[speechminer]** section of the Recording Processor configuration to a value of 1.
3. Using Genesys Administrator Extension on the Recording tab of the IVR Profile, modify the SpeechMiner Interaction Receiver field use https as the protocol in the URL.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA cert file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection from the backup Recording Processor to the primary Recording Processor

1. Configure HTTPS on the primary Recording Processor.

2. Get the corresponding PEM certificate for the Web Services server.
3. In the **[processing]** section of the Recording Processor configuration file, set the **post_uri** parameter to use https as the protocol in the URL.
4. In the **client** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS on the backup Recording Processor Server

Follow the same procedure used for the Primary Recording Processor Server using a new certificate and private key for the Backup Recording Processor's server.

Configuring the IVR Profile

Using Genesys Administrator Extension, configure the following parameters on the **Recording** tab of the IVR Profile:

1. **Recording Processor URI**—The URI that the Media Control Platform (MCP) uses to post the metadata of the audio recording after the recording is complete. For example, `http:// <Recording Processor Host>/api/contact-centers/<Contact Cente Domain Name>/recordings/`.

Important

The value for the URI must always end with a forward slash (/).

2. **SpeechMiner Interaction Receiver**—The URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile.
3. **SpeechMiner Interaction Receiver Authorization Header**—The authorization information required to connect to the SpeechMiner service used by the RPS. For example, `<SpeechMiner Webserver Username>:<SpeechMiner Webserver Password>`.

For more information, see the [Configuring GVP](#).

Configuring the Recording Processor Using Genesys Administrator Extension (Optional)

The Recording Processor uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Processor as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Processor as a "third party server" application in Genesys Administrator Extension. For more information, see the "Using the Management Layer" section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring RPS to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest RPS.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Processor Script of type Third Party Server.
4. Create a new application (for example, myRPS) in Genesys Administrator Extension using this new application template.
5. Set the Command Line parameter (for example, C:\Python311\python.exe).
6. Set the Host parameter in the application's server info to the correct Host object.
7. Set the Working Directory parameter to the <Recording Processor Install Directory>\rp directory. For example, /opt/genesys/Recording_Processor_Script_8.5/rp/.
8. Set the Command Line Arguments parameter to the appropriate values. For example, recording_process.py --config-file=/opt/genesys/Recording_Processor_Script_8.5/rp/rpconfig.cfg.
Refer to the [Starting the Recording Processor Script](#) section for additional command line parameters
9. Make sure that LCA has permission to read and write to the Recording Processor installation directory and Recording Processor log directory.
10. Save the configuration changes.
11. Ensure that the Configuration Server parameters in the Recording Processor configuration file are set appropriately. Refer to **Configure the Configuration Server Connection** tab on this page.

Important

The Recording Processor does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

For more information about the RPS options, see [Genesys Interaction Recording Options Reference](#).

Starting the Recording Processor Script

To launch the RPS, run the following command from the <Recording Processor Install Directory>:

```
<python executable> recording_process.py --config-file=rpconfig.cfg
```

Use the following command line when you want to run multiple instances of RPS on the same machine:

```
<python executable> recording_process.py --config-file=rpconfig.cfg --id=1 --port=8889
```

For each RPS instance, assign a unique id (--id parameter) and port number (--port).

Important

- --port defines the server port opened by the RPS process.
- --id represents the suffix of the:
 - application_name in the configuration file. For example, if application_name is defined in the configuration file as **RecordingProcessorScript** and --id 2 is specified in the command line, then the application object named **RecordingProcessorScript_2** will be used to start the program.
 - log files
 - metadata json files created in the failed folder
 - database file created by the process

By default the RPS log file is stored in the working directory. This can be changed by specifying a preexisting folder in the logfile_path parameter in the log file section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingProcessor
```

Recording Processor Script Legacy (Python 2) Deprecated

Important

Recording Processor Script Legacy (based on Python 2) has been discontinued as of March 31, 2024.

Important

Voice Processor, a multi-threaded microservice based on the Node.js platform, is an alternative to the Recording Processor Script (RPS).

- For information on deploying Voice Processor, see [Deploying Voice Processor](#).
- To migrate from an existing RPS deployment to Voice Processor, see [Migrating from RPS to Voice Processor](#).
- RPS is not supported for deployments integrated with SIP Cluster. If your deployment

uses SIP Cluster, you must use Voice Processor.

For new deployments, Genesys recommends using Voice Processor instead of RPS.

Genesys Interaction Recording (GIR) needs the Recording Processor Script (RPS) to manage the recording metadata between Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner.

Prerequisites

Before installing and configuring the RPS, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) (or [Web Services](#) if you're using version 8.5.210.02 or earlier) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.

Installing Recording Processor Script

Installing on Windows

1. Install 32 bit Python 2.7.5 or latest 2.7.x release from the [Python](#) website.
2. Install the RPS IP.
3. Unzip the <RPS>\thirdparty\httplib2-0.8.zip file.
4. From the newly created directory, run: `python setup.py install`.
5. Unzip the <RPS>\thirdparty\setuptools-1.3.2.zip file.
6. From the newly created directory, run: `python setup.py install`.
7. Unzip the <RPS>\thirdparty\python-dateutil-1.5.zip file.
8. From the newly created directory, run: `python setup.py install`.
9. Unzip the <RPS>\thirdparty\web.py-0.37.zip file.
10. From the newly created directory, run: `python setup.py install`.
11. Download pyOpenSSL for Windows (32 bit) from the [Python pyOpenSSL](#) site.
12. Install pyOpenSSL by running `pyOpenSSL-0.12.1.win32-py2.7.exe`.

Important

Installing pyOpenSSL (the previous steps) is optional, and is only required if an HTTPS server is needed for Recording Processor Script to receive metadata. Without it, only an HTTP server is supported.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).
2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `openssl-devel` (`yum install openssl-devel.x86_64`).
4. Install Python 2.7.5 or latest 2.7.x release from the [Python](#) website:
 - Genesys recommends that newer versions of Python are installed separately from existing versions (do not update).
 - See the [Example](#) below for an example of how to install CPython 2.7.6 on RHEL5.
5. Install/deploy the RPS IP.
6. Install `httplib2-0.8`:
 - a. Untar `httplib2-0.8.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `setuptools-1.3.2`:
 - a. Untar `setuptools-1.3.2.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `python-dateutil-1.5`:
 - a. Untar `python-dateutil-1.5.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `web.py-0.37`:
 - a. Untar `web.py-0.37.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `pyOpenSSL-0.12`:
 - a. Download `pyOpenSSL-0.12.tar.gz` from the [pyOpenSSL 0.12](#) site.
 - b. Untar the downloaded file, `pyOpenSSL-0.12.tar.gz` to the RPS installation directory.
 - c. From the newly created directory, run the following command to build the library: `python setup.py build`.
 - d. Install the library, run: `python setup.py install`.

Important

Installing pyOpenSSL (the previous steps) is optional, and is only required if an HTTPS server is needed for Recording Processor Script to receive metadata. Without it, only an HTTP server is supported.

RPS on RHEL8 cannot build pyOpenSSL-0.12. You must download **openssl-1.0.1e.tar.gz** from <https://ftp.openssl.org/source/old/1.0.1>, build openssl-1.0.1e, proceed to build Python 2.7.18 normally, and then build pyOpenSSL-0.12 while also updating LD_LIBRARY_PATH to point to the openssl-1.0.1e libraries.

Example: Installing CPython 2.7.6 on RHEL5 (64bit)

The following instructions are intended as an example only. A specific system or environment may require different steps when installing CPython 2.7.6 on RHEL5 (64bit):

1. Verify that `zlib-devel` is installed on the OS (`yum install zlib-devel`).
2. Verify that `sqlite dev` is installed on the OS (`yum install sqlite-devel.x86_64`).
3. Verify that `openssl devel` is installed on the OS (`yum install openssl-devel.x86_64`).
4. Download CPython 2.7.6 source from the [Python](#) site.
5. Untar compressed source.
6. Run `./configure --enable-ipv6`.
7. Run `"make altinstall"` (this should prevent the overwriting of any existing versions).

Upgrading Recording Processor Script

1. Stop the RPS process.
2. Stop the RPS application.
3. Back up the RPS configuration file (`rpconfig.cfg`) and the `sqlite` file from the `\rp` directory (`rpqueue.db`).
4. Rename the existing installation folder name to `<folder name>.<old.current_date>` or something similar.
5. Uninstall the RPS component.
6. Install the new RPS component.
7. Copy the `rpconfig.cfg` and `rpqueue.db` files from the previous version into the `\rp` folder inside the new installation directory.
8. Start the new RPS application.
9. Repeat the above steps for additional RPS instances.

Configuring Recording Processor Script

This section describes how to configure the Recording Processor Script for your environment.

Configuring High Availability

Recording Processor Cluster

RPS now provides High Availability support using multiple instances of RPS (all active). These active/active instances must be accessed through an HA proxy or load balancer. In this mode, each RPS is responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner based on the load it receives. Each Recording Processor is responsible for fetching metadata from all ICON DB Servers. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In each Recording Processor's **rpconfig.cfg** configuration file, in the **[processing]** section, set the following options:
 - **get_from_http_before_posting** = 1
 - **mode** = active
2. Ensure that all Recording Processor instances have the *same* network related configuration.

Important

Genesys recommends that multiple Recording Processor instances be deployed on a single host to optimize the available CPU and take advantage of parallel processing. Multiple Recording Processor instances can then be deployed on other hosts as needed.

3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the **Recording Processor URI** parameter to the load balancer's URL.
4. Configure the load balancer to balance traffic to the Recording Processor instances.

The following is an example configuration section that is needed for setting up an Apache load balancer for a three-instance Recording Processor cluster.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
  BalancerMember http://<IP address of the active1 Recording Processor server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active2 Recording Processor Server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active3 Recording Processor Server>:<active
Recording Processor port>
</Proxy>
```

Important

SpeechMiner version 8.5.2 or later is required for the Recording Processor cluster support to work properly.

Recording Processor Script Active/Backup HA

RPS can also provide High Availability support by using two RPS instances (active and backup) accessed through an HA proxy or load balancer in failover mode. In this mode, the active RPS is always responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner, and the backup instance is responsible for receiving and temporarily storing metadata if the active instance is unavailable. Once the active instance recovers, the balancer will direct clients to the active instance, and the backup instance will send any stored data to the active instance for metadata processing. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In the active Recording Processor's **rpconfig.cfg** configuration file located in the **[processing]** section, set the **mode** parameter to active.
2. In the backup Recording Processor's **rpconfig.cfg** configuration file:
 - Set the **mode** parameter to backup.
 - In the **[processing]** section, set the **post_uri** parameter to `http://<active_rp_ip>:<active_rp_port>/api/contact-centers/%s/recordings/`.
3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the Recording Processor URI parameter to the failover load balancer's URL.
4. Configure the load balancer to direct traffic to the active Recording Processor instance first and to the backup instance if an error/failure occurs.

The following is an example configuration section that is needed for setting up an Apache load balancer in failover mode for Recording HA support.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
  BalancerMember http://<IP address of the active Recording Processor server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the backup Recording Processor Server>:<active
Recording Processor port> status=H
</Proxy>
```

For more information about how to use Genesys Administrator Extension to configure your Contact Center, see the [Genesys Administrator Extension Help](#).

Configure Passwords

Important

In a Linux or Windows environment, RPS supports reading the environment variables for password related configuration parameters in order to avoid storing the password in plain-text in the configuration file. When both are available, the environment variables take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

HTCC_PASSWORD - maps to the existing configuration parameter under the **htcc** section, password value.

AUTH_PASSWORD - maps to the existing configuration parameter under the **auth** section, password value.

CONFIG_SERVER_PASSWORD - maps to the existing configuration parameter under the **config_server** section, password value.

<ICON_ID>_DB_INFO_PASSWORD - maps to the existing configuration parameter under the **<ICON_ID>_db_info** section, password value, where **<ICON_ID>** refers to the ICON instance listed in the **icon_db_servers** section.

For example, if you have VCCSIPSwitch: icon1 the environment variable that corresponds to the icon1_db_info password is icon1_DB_INFO_PASSWORD.

In a Windows environment only, the Recording Processor Script (RPS) can store passwords in the Windows Vault instead of in the **rpconfig.cfg** file or requiring the use of environment variables.

For example, run the following command for the Recording Processor Script credentials located at **<Recording Processor Directory>\rp**. This command will prompt the user to enter valid values for the password/key configuration parameters and stores the passwords in the encrypted file named **rp.secret**:

Command to store:

```
encryptPassword.bat -password <password_string>
```

Where **<password_string>** is a comma-delimited series of key/value pairs, use the format **<environment variable name 1>=<environment variable value 1>,<environment variable name 2>=<environment variable value 2>,<environment variable name 3>=<environment variable value 3>**, and so on. Note that space is not allowed in **<password_string>**.

For example:

```
encryptPassword.bat -password "HTCC_PASSWORD=somepassword1, AUTH_PASSWORD=somepassword2, CONFIG_SERVER_PASSWORD=somepassword3, ICON1_DB_INFO_PASSWORD=somepassword4, ICON2_DB_INFO_PASSWORD=somepassword5"
```

Important

Passwords used with this command cannot contain a comma or an equals sign.

Configure the Configuration Server Connection

To configure the Configuration Server connection, set the following parameters in the **[config_server]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
application_name	Empty	Specifies the name of the RPS application object in the Configuration Server, when using RPS as a third party server application.
hostname	<ip address>	Specifies the IP address of the primary Configuration Server.
port	2020	Specifies the port of the primary Configuration Server.
username	default	Specifies the Configuration Server username.
password	password	Specifies the Configuration Server password. Note: The password can be overridden by the CONFIG_SERVER_PASSWORD environment variable.
backup_host	Empty	Specifies the IP address of the backup Configuration Server.
backup_port	Empty	Specifies the backup port of the backup Configuration Server.

Important

Recording Processor Script does not support a secure connection to the Configuration Server.

Configuring the Server Port

In the **[rp_server]** section of the **rpconfig.cfg** file, set the **port** parameter.

Important

You can also set the "port" parameter using the command line with the --port command line argument. The command line argument takes precedence over the configuration file value.

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri	http://<Web Services IP>:<Web Services Port>	Specifies the Base URI for accessing the Interaction Recording Web Services (Web Services) API.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	Specifies the password used to access the Interaction Recording Web Services (Web Services) account. Note: The password can be overridden by the HTCC_PASSWORD environment variable.

Each Interaction Recording Web Services (Web Services) instance must have a region associated with it. Set the region parameter in the [metadata] section of the rpconfig.cfg file to match the region associated with Interaction Recording Web Services (Web Services) instance set to receive the Recording Processor's metadata.

Configuring Cross-Site Request Forgery (CSRF) Protection

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled, set the following parameter in the **[htcc]** section of the **rpconfig.cfg** file:

- **csrfp = 1**

Configuring the Connection to SpeechMiner

To configure the SpeechMiner Connection:

1. In the IVR Profile, set the recording destinations to point to the SpeechMiner interaction receiver:
 - a. Login to Genesys Administrator Extension, and navigate to **Configuration > System > Configuration Manager**.
 - b. Under **Voice Platform**, select **Voice Platform Profiles**.
 - c. Click on the IVR Profile for which you want to set the recording destination.
 - d. Select the **Recording** tab.
 - e. In the **SpeechMiner Interaction Receiver** field, enter the URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile. For example, `https://<SpeechMiner IP>/interactionreceiver`.

- f. In the **SpeechMiner Interaction Receiver Authorization Header** field, enter the authorization information (username:password) required to connect to the SpeechMiner service used by the RPS. For example, user:password.

Important

The values of these options must match the corresponding configuration options in the SpeechMiner system.

Configuring Failed Message Files

The Recording processor can backup messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner. These files are located in the **<recording processor dir>\failed** folder.

In the **rpconfig.cfg** configuration file, add the following parameter:

```
[processing]
backup_failed_metadata = 1
```

Configuring the Agent Hierarchies

Recording Processor Script uses the agent hierarchy information to set the access control information for recordings within the recording metadata. Refer to [Access Control for Genesys Interaction Recording Users](#) to configure this appropriately.

Configuring Basic Authorization

In the **rpconfig.cfg** configuration file, set the following parameters:

```
[auth]
# Basic Authentication username and password. Set username blank to disable.
username = rp_username
password = rp_password
```

Important

- The username and password must match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.
- The password can be overridden by the **AUTH_PASSWORD** environment variable.

Configuring After Call Work

Recording Processor can collect After Call Work (ACW) customized data from ICON.

In the **rpconfig.cfg** file, in the **[processing]** section, add the following parameters:

- **enable_acw**—Set it to 1.
- **acw_threshold_minutes**—Set it to the maximum time to wait for the customized attached data.

Important

- If Call Customized Attached Data is still not available in the ICON database after *acw_threshold_minutes*, the RPS will stop collecting customized data for this recording and write it to the database.
- If *enable_acw* is set to 0, ACW customized data will not be included.
- If disposition code is required in the metadata, you must set **enable_acw** and **acw_threshold_minutes** using Recording Processor configuration. The disposition code is part of the user data collected during ACW. For this reason, **enable_acw** must be enabled in the Recording Processor. If it is not enabled, the data will not be collected. If the disposition code must be collected from the Recording Processor, configure the following to include the disposition code for recording:
[processing] enable_acw=1
[metadata] acw_threshold_minutes=5. Where 5 is the maximum time (in minutes) to wait for the disposition code. In the ICON configuration, the **EventData** parameter in the **custom-states** section, must include *char,DispositionCode* and **store-event-data** must be set to *conf* to collect the attached data:
[custom-states] store-event-data=conf EventData=char,DispositionCode
For additional information, refer to the [ICON Deployment Guide](#).

Configuring ICON for Recording Processor

Important

When configuring Recording Processor to connect to a primary and backup ICON Database in HA mode, two separate DB Servers must be used. The DB Servers must run in an active/active pair mode.

To configure ICON, edit the **rpconfig.cfg** configuration file as follows:

1. Configure the switches:

Add a configuration option for each switch name under the **[icon_db_servers]** section. You can specify more than two ICON databases per SIP Switch configuration. For example:

```
[icon_db_servers]
SIP_Switch1: icon1
SIP_Switch2: icon2, icon2Backup
SIP_Switch3: icon3, icon4, icon5, icon6
```

Important

In the above example, **SIP_Switch3** has 4 ICON databases. The Recording Processor Script (RPS) keeps track of the ICON database instance currently used. If the current database instance becomes unavailable, RPS will attempt the operation in the next database.

The configuration option name must match the exact name of the switch as configured in the Genesys configuration. The primary and backup ICON names must be unique, but do not have to match anything in the Genesys configuration.

2. Configure the ICON Connection Settings:

- For each unique ICON specified in the first step, create a new section using the following syntax: `<ICON_ID>_db_info`, where `<ICON_ID>` corresponds to the values defined in the **[icon_db_servers]** section above.
- **dbengine** must be `mssql`, `oracle`, `db2`, or `postgres`.
- **dbserver_host** and **dbserver_port** specify the host and port information for the Genesys DB Server.
- **dbms** specifies the host where the database resides.

The following is an example using the values for **SIP_Switch1** and **SIP_Switch2** from step 1:

```
[icon1_db_info]
dbserver_host = 10.0.0.221
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2_db_info]
dbserver_host = 10.0.0.222
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2Backup_db_info]
dbserver_host = 10.0.0.223
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon_oracle_db_info]
dbserver_host = <host>
dbserver_port = <port>
```

```
username = <username>
password = <password>
dbname =
dbms = <database host/Oracle SID>
dbengine = oracle

[icon_postgres_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname = <database name>
dbms = <database host>
dbengine = postgres

[icon_db2_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host>
dbengine = db2
```

Important

- For Oracle or DB2 implementations, the **dbname** parameter must be left blank or empty.
- The password can be overridden by the **<ICON_ID>_DB_INFO_PASSWORD** environment variable.

In the example above, the RPS will use the connection properties in section **[icon1_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch1. The RPS will use the connection properties in section **[icon2_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch2. In the case of SIP_Switch2, the RPS will use the connection settings in **[icon2Backup_db_info]** if the primary ICON (icon2) is unavailable when recording metadata is being processed.

Configure how to Filter Metadata from ICON

The Recording Processor supports the ability to filter specific attached data fields (based on the key name), such as attached data and After Call Work (ACW) customized data retrieved from the ICON database. This support prevents specific metadata from reaching additional GIR related components (for example, SpeechMiner).

The following two sections describe how to:

- **Filter attached data.**
- **Filter ACW.**

Important

- Verify that the following items are not removed from the filter. Removing these items may cause errors in GIR:
 - RECORD_PARTITIONS
 - RECORD_PROGRAM
 - GSIP_REC_FN
- When running SpeechMiner, you must include Workspace Web Edition (WWE) in the attached_data_filter and acw_custom_data_filter Recording Processor configuration values. For example:


```
[filter]
attached_data_filter=^ORSI|^WWE
acw_custom_data_filter=^ORSI|^WWE
```

Filter Attached Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **attached_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out attached data whose key name matches the pattern.

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

4. Add a new option called **attached_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude key names that should not be filtered out (for example, like RECORD_PARTITIONS).

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
attached_data_filter_exception = ^RECORD_PARTITIONS$      ; (Note: this is the default
value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

5. Restart the Recording Processor.

Filter ACW Related Custom Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **acw_custom_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out ACW whose key name matches the pattern.

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

4. Add a new option called **acw_custom_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude ACW that should not be filtered out (for example, like GRECORD_PARTITIONS).

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
acw_custom_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the
default value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

5. Restart the Recording Processor.

Configuring SSL for Recording Processor

To configure SSL:

Configure HTTPS on the Primary Recording Processor Server

1. Make sure pyOpenSSL is installed.
2. Create a self-signed certificate and private key for the Recording Processor host. For example, on RHEL run: `openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem`
3. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—To point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228.pem`.
4. Give the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section [Genesys Voice Platform 8.5 User's Guide](#).

5. Restart Recording Processor.

Configure the HTTPS connection to Interaction Recording Web Services (Web Services)

1. Set up HTTPS on Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). See the [Genesys Security Deployment Guide](#).
2. Get the corresponding certificate for the Interaction Recording Web Services (Web Services) server. Set the **caCertificate** option in your Interaction Recording Web Services application (see **caCertificate** if you're using a Web Services application).
3. In the **[htcc]** section of the Recording Processor configuration file, set **base_uri** parameter to use https.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file. See the <Python27 install directory>\Lib\site-packages\httplib2\cacerts.txt file for an example.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection to SpeechMiner

1. Set up HTTPS on SpeechMiner. See the [Genesys Security Deployment Guide](#).
2. Set the **disable_ssl_certificate_validation** parameter in the **[speechminer]** section of the Recording Processor configuration to a value of 1.
3. Using Genesys Administrator Extension on the Recording tab of the IVR Profile, modify the SpeechMiner Interaction Receiver field use https as the protocol in the URL.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA cert file. See the <Python27 install directory>\Lib\site-packages\httplib2\cacerts.txt file for an example.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection from the backup Recording Processor to the primary Recording Processor

1. Configure HTTPS on the primary Recording Processor.
2. Get the corresponding PEM certificate for the Web Services server.

3. In the **[processing]** section of the Recording Processor configuration file, set the **post_uri** parameter to use https as the protocol in the URL.
4. In the **client** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file. See the <Python27 install directory>\Lib\site-packages\httplib2\cacerts.txt file for an example.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS on the backup Recording Processor Server

Follow the same procedure used for the Primary Recording Processor Server using a new certificate and private key for the Backup Recording Processor's server.

Configuring the IVR Profile

Using Genesys Administrator Extension, configure the following parameters on the **Recording** tab of the IVR Profile:

1. **Recording Processor URI**—The URI that the Media Control Platform (MCP) uses to post the metadata of the audio recording after the recording is complete. For example, http:// <Recording Processor Host>/api/contact-centers/<Contact Center Domain Name>/recordings/.

Important

The value for the URI must always end with a forward slash (/).

2. **SpeechMiner Interaction Receiver**—The URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile.
3. **SpeechMiner Interaction Receiver Authorization Header**—The authorization information required to connect to the SpeechMiner service used by the RPS. For example, <SpeechMiner Webserver Username>:<SpeechMiner Webserver Password>.

For more information, see the [Configuring GVP](#).

Configuring the Recording Processor Using Genesys Administrator Extension (Optional)

The Recording Processor uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Processor as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Processor as a "third party server" application in Genesys Administrator Extension. For more information, see the "Using the Management Layer" section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring RPS to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest RPS.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Processor Script of type Third Party Server.
4. Create a new application (for example, myRPS) in Genesys Administrator Extension using this new application template.
5. Set the Command Line parameter (for example, C:\Python27\python.exe).
6. Set the Host parameter in the application's server info to the correct Host object.
7. Set the Working Directory parameter to the <Recording Processor Install Directory>\rp directory. For example, /opt/genesys/Recording_Processor_Script_8.5/rp/.
8. Set the Command Line Arguments parameter to the appropriate values. For example, recording_process.py --config-file=/opt/genesys/Recording_Processor_Script_8.5/rp/rpconfig.cfg.
Refer to the [Starting the Recording Processor Script](#) section for additional command line parameters
9. Make sure that LCA has permission to read and write to the Recording Processor installation directory and Recording Processor log directory.
10. Save the configuration changes.
11. Ensure that the Configuration Server parameters in the Recording Processor configuration file are set appropriately. Refer to **Configure the Configuration Server Connection** tab on this page.

Important

The Recording Processor does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

For more information about the RPS options, see [Genesys Interaction Recording Options Reference](#).

Starting the Recording Processor Script

To launch the RPS, run the following command from the <Recording Processor Install Directory>:

```
<python executable> recording_process.py --config-file=rpconfig.cfg
```

Use the following command line when you want to run multiple instances of RPS on the same machine:


```
<python executable> recording_process.py --config-file=rpconfig.cfg --id=1 --port=8889
```

For each RPS instance, assign a unique id (--id parameter) and port number (--port).

Important

- --port defines the server port opened by the RPS process.
- --id represents the suffix of the:
 - application_name in the configuration file. For example, if application_name is defined in the configuration file as **RecordingProcessorScript** and --id 2 is specified in the command line, then the application object named **RecordingProcessorScript_2** will be used to start the program.
 - log files
 - metadata json files created in the failed folder
 - database file created by the process

By default the RPS log file is stored in the working directory. This can be changed by specifying a preexisting folder in the logfile_path parameter in the log file section of the configuration file. For example:

```
logfile_path = C:\logs\recordingProcessor
```

Deploying Voice Processor

Genesys Interaction Recording (GIR) needs Voice Processor to process recording metadata from Media Control Platform (MCP), combine this metadata with data collected from Genesys Info Mart (GIM), and forward the result to Interaction Recording Web Services (RWS) and SpeechMiner Interaction Receiver (SM IR). During this process, recordings from multiple call legs are combined into a single interaction.

Important

If you are not using Voice Processor, Recording Processor Script (RPS) can be used. However, for new deployments, Genesys recommends using Voice Processor instead of RPS.

This topic contains the following sub-topics:

- [Prerequisites](#)
- [Preparing your Docker environment](#)
- [Configuring Voice Processor](#)
- [Deploying Voice Processor to Docker](#)
- [Miscellaneous Docker tips](#)
- [Migrating from RPS to Voice Processor](#)
- [Monitoring and troubleshooting information](#)

Comparison of Voice Processor and RPS

Voice Processor is a multi-threaded microservice based on the Node.js platform and it replaces the Python-based Recording Processor Script (RPS). The key advantage of Voice Processor is that since Node.js is a multi-threaded platform, a single Voice Processor instance can handle an incoming recording post load equivalent to 30-40 instances of RPS (20+ recordings per second). Therefore, a single instance should be sufficient for most customers. For customers with extremely high volumes, or who require redundancy, the Voice Processor can be run behind a load balancer similar to the existing RPS deployments.

Another benefit is that, in the event of outages that prevent posting of recordings to RWS and SpeechMiner Interaction Receiver, the Voice Processor automatically retries these posts for up to 40 days. As a result, you do not have to manually recover recordings if you resolve the downstream outage within that period.

The Voice Processor retrieves additional metadata from Genesys Info Mart (GIM) instead of Interaction Concentrator Database (ICON). The format and contents of the metadata posted by the Voice Processor to RWS and SM IR do not differ significantly from the format and contents posted by

RPS. However, there are some differences that may impact third-party integrations that download recording metadata from RWS, Recording Backup Service (RCBS), or from SpeechMiner. You must consider the following differences in the format and contents of the metadata:

- Name of the **eventID** property in the **eventData** list is different.
- Metadata that is meant to be internal-use only is not posted by the Voice Processor.
- As Genesys Info Mart is a data warehouse that is updated on a periodic basis through ICON data, the arrival of recordings in SpeechMiner is slightly delayed when compared to RPS.

Prerequisites

- Docker version 17.12.1-ce or higher running on a x86_64 Linux host.
- Ansible 2.6 or higher installed on the Docker host and the deployment is using Red Hat Enterprise Linux 7.
- Ansible 2.13 or higher installed on the Docker host and the deployment is using Red Hat Enterprise Linux 8.
- PostgreSQL 12.11 or higher.
- Genesys Info Mart 8.5 or higher installed on Microsoft SQL Server or PostgreSQL. For information on the system requirements for GIM, see [Genesys Info Mart Requirements](#).
- Interaction Recording Web Services (RWS) 8.5.201.90 or higher.
- SpeechMiner 8.5 or higher if you are using SpeechMiner. We recommend that you install Speechminer before deploying Voice Processor.

We recommend that you have the following details before proceeding with deployment:

- Host name, port, database name for Genesys Info Mart, and user (read-only) credentials.
- Host name and port for the Interaction Recording Web Services (RWS), and Operation Admin (ops) credentials.
- Configuration Manager credentials for an account with access to the IVR Profile.
- Host name, port, and credentials needed to post to SpeechMiner Interaction Receiver. These details are required only for new installations.

Preparing your Docker environment

Extracting installation files

You can download the Docker image from the Genesys customer portal. The Docker image is a .tar file that contains the installation and configuration files required to set up and run the Voice Processor. Extract and copy the files from the image using the following steps:

1. Load the Docker image.

```
zcat <.tar file> | docker load
```

2. View the list of Docker images and make a note of the newly loaded image.

```
docker image ls
```

3. Add a custom tag to the docker image for your reference.

```
docker tag <image ID> <tag>
```

4. Copy the files from the image.

```
id=$(docker run --rm -dt <image> cat) && docker cp $id:/rps/compose . && docker stop $id
```

Now you have the sample configuration files in **./compose/defaults**, an Ansible playbook in **./compose** to help you set up and run the Voice Processor, and an SQL file for database setup. You will need these files for installing and configuring the Voice Processor.

Tip

You can refer to the custom tag of your docker image when setting parameters in the configuration files such as **settings-override.yml**, **secrets.yml**, and **docker-config.yml**.

Setting up Docker

Docker on the host must be running in swarm mode, but a multi-host swarm is not required. A new network inside Docker is used for the deployment. Docker swarm and its network can be set up this way:

```
docker swarm init
docker network create gir_vp --driver overlay --scope swarm
```

If the network that you created collides with an existing network in your environment, you can define the network's IP range like this:

```
docker network create --subnet 10.99.99.0/24 --gateway 10.99.99.1 --scope swarm gir_vp
```

Docker configs and secrets

Configs and secrets are Docker objects available in Docker swarm mode for storing run-time container configuration files and are mounted inside the container at run-time. The key difference between the two is that secrets are encrypted in Docker when at rest.

Docker logs

Docker container logs are typically stored under **/var/lib/docker/containers**, but container logs can be accessed simply by `docker logs <container name>`. Logs are rotated based on run-time configurations.

Configuring Voice Processor

This section contains the following sub-sections:

- [PostgreSQL database configuration](#)
- [Service level configuration](#)
- [Genesys Voice Platform profile configuration](#)
- [Tenant level configuration](#)
- [GIM DB ETL configuration](#)

PostgreSQL database configuration

The Voice Processor requires a service-specific database that tracks work in progress items. This database runs on a PostgreSQL server. Set up the database using the following steps:

1. Create a database in your PostgreSQL server for the Voice Processor.
2. Create a PostgreSQL user and grant all privileges to the database that you created in the previous step.
3. Assign a password to the user that does not contain a backward slash (\) or quotation marks, as they might cause issues later.
4. Make a note of the database name, user name, and password — they are needed when configuring the Voice Processor in later steps.
5. Confirm that the **standard_conforming_strings** parameter of the PostgreSQL server is set to on (default).

Important

- GIR Voice Processor must have a separate PostgreSQL DB from Config Server since the Voice Processor PostgreSQL DB requires the `standard_conforming_strings` setting to be on and the Config Server PostgreSQL DB requires the `standard_conforming_strings` setting to be off.
- Voice Processor supports connections to PostgreSQL DB when `password_encryption` is set to `md5` or `scram-sha-256` in the **postgresql.conf** file.
- When using a PostgreSQL DB for Genesys Info Mart, if `password_encryption` is set to `scram-sha-256` in the **postgresql.conf** file, the Genesys Info Mart version must be 8.5.016.04 or higher.

6. Run the provided script, `create_node_rps_tables_v2.sql`, against this new database to provision it.

Important

To avoid possible conflicts with their settings requirements, Genesys recommends not hosting the Voice Processor and Configuration Server databases on the same PostgreSQL instance.

Service level configuration

You can follow the instructions provided with the configuration files available in the default directory. You can copy the provided configuration files and make changes to your copies. We recommend that you use a version control repository to store your configurations. Add the PostgreSQL database, user name, and password to the **nodeRpsDb** setting in your copies of **settings-override.yml** and **secrets.yml**.

Voice Processor database settings

To enable TLS connection to the Voice Processor database, set the **ssl** parameter to **true** and configure the **trustedCA** parameter under **nodeRpsDb** in **settings-override.yml**.

```
nodeRpsDb:
  database: <database name>
  host: <db server hostname>
  port: <db port>
  user: <db user>
  ssl: < true / false >
  trustedCA: false / true / "<path to root certificate>"
```

The **ssl** parameter is optional and its default value is **false**. When you set it to **true**, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to **true**, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is **false**.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is **true**.
- Authenticate the server certificate against the specified root authorities. Set **vpdb_ca_cert** in your copy of **docker-config.yml** with the **<path to root certificate>** value.

Voice Processor HTTPS settings

The **rwsBaseUri** setting in **settings-override.yml** supports HTTPS. For example:

```
https://<RWS hostname>:<RWS port>
```

To use HTTPS on the Voice Processor service API, set **https** to **true** in your copy of **docker-config.yml**. You must provide the server private key, public key, and path to the files.

```
https: true
tls:
  privkey: <path to the private key file>
  pubkey: <path to the public key file>
```

MCP post basic authentication

Add the following lines to the **settings-override.yml** file to enable basic authentication for the endpoint used by the MCP to post recording metadata:

```
authUsername: "<basic auth username>"
authPassword: "<basic auth password>"
```

If you add these options, you must also configure the Voice Platform profile option, **recording client.callrec_authorization**, in the **[gvp.service-parameters]** section to match these credentials. As basic authentication involves sending the credentials in plain text format, we strongly recommend that you use TLS for maximum security. Note that the other Voice Processor endpoints are not authenticated. Therefore, you must install the Voice Processor behind a firewall or API gateway to restrict access. You can obtain a summary of endpoints exposed by the Voice Processor service by accessing:

```
http://<GIR VP hostname>:<port>/apidoc
```

Setting the Voice Processor Docker image

Add the following line to the **docker-config.yml** file to specify the Voice Processor docker image that was imported:

```
image: <image ID>:<tag>
```

To find the values for <image ID>:<tag>, use the `docker images` command. An example is given below:

```
[root@CENTOS7-CloudVM defaults]# docker images
REPOSITORY          TAG                 IMAGE ID
CREATED             SIZE
voice-processor_v9000025  latest            7320945e8b25
21 months ago       979MB
pureengage-docker-production.jfrog.io/gir_rp_nodejs  9.0.000.04.023    7320945e8b25
21 months ago       979MB
[root@CENTOS7-CloudVM defaults]#
```

Genesys Voice Platform profile configuration

Use HTTPS protocol in the Voice Processor URL when HTTPS is enabled in the Voice Processor service API.

```
recordingclient.callrec_dest = fixed,https://<VP hostname>:<VP port>/api/contact-centers/<CCID>/recordings/
```

Use HTTPS protocol in the SpeechMiner Interaction Receiver URL when HTTPS is enabled on the SpeechMiner Interaction Receiver. When using HTTPS for the SpeechMiner URL, by default, the Voice Processor does not validate SpeechMiner server certificate. You can set **sm_ca_cert** in your copy of **docker-config.yml** with the <path to root certificate> value to authenticate the server certificate against the specified root authorities.

```
recordingclient.rp.speechminer_uri: fixed,https://<Speechminer backend
hostname>/interactionreceiver/
```

Tenant level configuration

As the Voice Processor is designed to support Genesys cloud multi-tenancy model, settings that may vary from tenant to tenant are stored in an RWS group settings called **rps-provisioning**:

Important

You need an Ops Admin user account to access these settings. For more information on how to update settings in RWS, see [Settings API](#).

You must specify the Ops Admin user name and password in your copy of **secrets.yml**. The tenant level configuration values are set to the RWS group settings **rps-provisioning** using HTTP POST. For example:

```
curl -u <Ops admin user>:<password> -X POST -H "Content-Type: application/json"
<rwsBaseUri>/api/v2/ops/contact-centers/<ccid>/settings/rps-provisioning -d @rps-
settings.json
```

Where **rps-settings.json** contains settings like: eventDataFilters, gimDb, rwsPostRecBaseUri and others.

To confirm the Voice Processor per tenant settings, use HTTP GET. For example:

```
curl -u <Ops admin user>:<password> -X GET "<rwsBaseUri>/api/v2/ops/contact-
centers/<ccid>/settings/rps-provisioning?location=*&ignoreParentLocations=false"
```

GIM database

You must provide information needed to access the tenant's GIM database. To enable TLS connection to the GIM database, set the **ssl** parameter to true and configure the **trustedCA** parameter under GIM database settings in tenant level configuration.

```
{
  "name": "gimDb",
  "value": {
    "primary": {
      "host": "<GIM server hostname>",
      "port": "<GIM server port (default 5432 for Postgres, 1433 for MS SQL)>",
      "user": "< DB user name >",
      "database": "<database name",
      "password": "<DB user password>",
      "dbType": "<postgres or mssql, default postgres>",
      "ssl": < true / false >,
      "trustedCA": false / true / "<path to root certificate>",
    },
    "backup": {
      < same settings as for primary >
    }
  }
}
```

The **ssl** parameter is optional and its default value is false. When you set it to true, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to true, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is false.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is true
- Authenticate the server certificate against the specified root authorities by performing the following steps:
 1. Set **gim_ca_cert** in your copy of **docker-config.yml** with the <path to root certificate> value.
 2. Set **trustedCA** to /rps/rpsdata/gimCA in GIM database settings to be posted to tenant level configuration.

The **backup** parameter is optional. You can omit it if there is only one GIM database available.

RWS posting

You must specify the RWS instance to which recordings are posted. As this is a region-based setting, multi-regional deployments can ensure that recording data stays within the jurisdictional boundaries. The Voice Processor instance selects the location identified through the **nodePath** of the RWS server from which the setting is retrieved or the nearest matching parent. The **backup** parameter is optional. The URL supports HTTPS. When using HTTPS for the RWS URL, by default, the Voice Processor does not validate RWS server certificate. You can set **rws_ca_cert** in your copy of **docker-config.yml** with the <path to root certificate> value to authenticate the server certificate against the specified root authorities.

For example, the following setting applies to all Voice Processor instances:

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

The following setting would override the above global setting for Voice Processor instances that retrieved the setting from an RWS node with **nodePath** /US or /US/* :

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/US",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

Event filtering

You can use filters to remove unwanted data from the recording metadata. The event filtering settings are similar to RPS except the mechanism of how the default filters are disabled.

```
{
  "name": "eventDataFilters",
  "value": {
    "attachedDataFilter": "regex for new attached data filter",
    "attachedDataFilterException": "regex for new attached data filter exception",
  }
}
```

```

    "acwCustomDataFilter": "regexp for new ACW data filter",
    "acwCustomDataFilterException": "regexp for new ACW data filter exception"

    -- or, to disable the default filters or filter exceptions --

    "disableAttachedDataFilter": true,
    "disableAttachedDataFilterException": true,
    "disableAcwCustomDataFilter": true,
    "disableAcwCustomDataFilterException": true
  }
}

```

The default filters are:

- **attachedDataFilter:** ^ORSI:|^WWE|^PegAG
- **attachedDataFilterException:** ^(GRECORDER_(PARTITIONS|PROGRAM)|GSR_STATE|GSIP_REC_FN)\$
- **acwCustomDataFilter:** ^ORSI:|^WWE|^PegAG
- **acwCustomDataFilterException:** ^(GRECORDER_(PARTITIONS|PROGRAM)|GSR_STATE|GSIP_REC_FN)\$

Complete after-call work (ACW) threshold

The ACW threshold indicates how long the Voice Processor waits, in minutes, following the end of an interaction to update custom data. Custom data entered by agents after this interval is not added to recording metadata. The default value is zero.

```

{
  "name": "acwThresholdMinutes",
  "value": <ACW wait interval in minutes>
}

```

GIM DB ETL configuration

You must configure the GIM ETL application properly to ensure recording metadata is posted from the Voice Processor to SpeechMiner in a timely manner.

The **etl-start-time**, **etl-end-time**, and **etl-timezone** options in the **[schedule]** section are used to configure a daily maintenance period during which population of GIM data is paused for maintenance purpose. New recordings posted to the Voice Processor during this period are not processed and they are held temporarily in a database until the maintenance period finishes and the relevant GIM data becomes available. You must configure the **maintain-start-time** option such that the GIM ETL maintenance job begins and completes during the maintenance period.

The **etl-frequency** option in the **[schedule]** section is used to specify the cycle time of the GIM ETL jobs that populate the recording metadata used by the Voice Processor. We recommend that you use the default value of one minute. Note that any time longer than 3 minutes may cause subsequent delays in recording posts. If a longer **etl-frequency** setting is used, then the value of the Voice Processor service setting, **rpsInitialInteractionTimeout**, should be increased accordingly.

The **user-event-data-timeout** option in the **[gim-etl]** section is used to ensure that custom attached data entered during after-call work is captured. You can increase the default value of one hour if your agents will spend more than a few minutes in after-call work.

Important

Consult Genesys before setting non-default values for the following options.

The **max-call-duration**, **merge-failed-is-link-timeout**, and **extract-data-stuck-threshold** options in the **[gim-etl]** section must be configured properly to ensure completeness of the call metadata recorded in GIM. For more information on these options, see [Operations-Related Options for Genesys Info Mart](#).

Deploying and Starting Voice Processor

Deploy Voice Processor to the newly configured Docker swarm using Ansible, referencing your copies of the default configuration files. This step also starts Voice Processor.

Before deploying, the **settings-override.yml** and **secrets.yml** files (or the yaml files you have designated to provide these settings) have several mandatory parameters that must be configured, as described below.

In the **settings-override.yml** file, the following parameters are required:

- **rwsBaseUri** – Specifies the address of the RWS cluster that will provide Voice Processor with contact center settings, including tenant-specific configurations such as Genesys Info Mart (GIM) database information and ACW Wait Time. Example: `http://some-rws-host.com:8090`
- **region** – Controls the region section in the metadata POSTed to RWS. This must match the **crRegion** setting of the RWS cluster to which recordings are posted. Example: `usa`
- **nodeRpsDb** – This section specifies the Voice Processor Persistence Database, which is required for storing recording metadata while Voice Processor is processing them.
 - **database** – Database on the host that will hold recording metadata. Example: `noderrpsdb`
 - **host** – Host of the Persistence Database. Example: `noderrpsdb.com`
 - **port** – Port that the Persistence Database is listening on. Example: `5432`

In the **secrets.yml** file, the following parameters are required:

- **nodeRpsDb** – This section specifies the credentials to the Voice Processor Persistence Database.
 - **user** – The user name to connect to the Persistence Database.
 - **password** – The password to connect to the Persistence Database.
- **rwsUserName** – The user name for authenticating with RWS.
- **rwsPassword** – The password for authenticating with RWS.

For an example of how the yaml files should be structured, you can refer to the default yaml files that were included with Voice Processor. These files are located at **<INSTALL_DIR>/defaults/**, where **<INSTALL_DIR>** is the location where you extracted the installation files to during the [Preparing](#)

your Docker environment step.

After configuring the default configuration files, deploy and start Voice Processor:

```
ansible-playbook \
  -e docker_config=<defaults directory of Your Docker Configuration Yaml (e.g. docker-
  config.yaml)> \
  -e logger_config=<defaults directory of Your Logger Configuration Yaml (e.g. logger-
  config.yaml)> \
  -e settings_override=<defaults directory of Your Voice Processor Configuration Yaml
  (e.g. settings-override.yaml)> \
  -e secrets=<defaults directory of Your Voice Processor Secrets Yaml (e.g.
  secrets.yaml)> \
  gir-vp-playbook.yaml
```

Important

If the above options are not specified, then the .yaml files in **./compose/defaults** will be used.

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }'
girvp.company.com/api/active-version
```

Validating

1. Place a call to an agent or a test agent that is configured for recording.
2. Verify that the call arrives at the SpeechMiner UI. It should take 5 to 15 minutes depending on your configured ACW wait setting.
3. Assuming that live traffic is not recorded, you can use the health check endpoint `<domain:port>/api/status?verbose=1`. The items `recordingsInProgress` or the MCP Post operational status can be helpful in determining whether or not the recording is arriving at the Voice Processor. This also helps you to isolate GVP configuration problems from problems with the Voice Processor service. If a load balancer is used, the node serving the health check may not be the one that handled the recording. Therefore, several health checks may be required to cover the whole cluster.

Upgrading

Docker object configurations and secrets cannot be upgraded. We recommend that you remove the stack, update the required configurations, and redeploy.

```
docker stack rm <gir_vp>
ansible-playbook \
```

```
-e docker_config=mydocker.yml \  
-e logger_config=mylogger.yml \  
-e settings_override=mysettings.yml \  
-e secrets=mysecrets.yml \  
gir-vp-playbook.yml
```

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }'  
girvp.company.com/api/active-version
```

Miscellaneous Docker tips

- To view network details:

```
docker network inspect <network_name>
```

- To view a list of your swarm stacks:

```
docker stack ls
```

- To view a list containers in your stack:

```
docker stack ps <gir_vp>
```

- To view the container logs:

```
docker logs <container name>
```

- To remove everything to start again:

```
docker stack rm gir_vp  
docker network rm <network_name>  
docker swarm leave --force
```

Migrating from RPS to Voice Processor

This section explains how to migrate from an existing RPS deployment to Voice Processor.

Prerequisites

- Voice Processor is fully deployed

- The following Voice Processor dependent components are working as expected:
 - Interaction Recording Web Services (RWS)
 - Genesys Info Mart Database
 - SpeechMiner Interaction Receiver
 - Voice Processor Database

Migrating procedure

You can migrate from RPS to Voice Processor by changing the IVR profile, even if a Load Balancer is being used for RPS.

You must configure the **Recording Processor URI** parameter in the **Recording** tab of the IVR profile using Genesys Administrator Extension (GAX). This URI is used by Media Control Platform (MCP) to post metadata of the audio recording after the recording is complete. You must change this parameter to ensure that MCP posts metadata to the Voice Processor instead of RPS. For example:

```
http://<Voice Processor Host>:<Voice Processor Port>/api/contact-centers/<Contact Center Domain Name>/recordings/
```

The value for the URI must always end with a forward slash (/). For more information, see [Deploying Genesys Voice Platform for GIR](#).

Important

We recommend that you save the original value that is needed if rollback becomes necessary.

Validation

1. Place a test call.
2. After 15 to 20 minutes, check the SpeechMiner UI for the test call recording that should appear for the test agent.
3. RPS should no longer receive any call data.

Rollback

Restore the original value of the **Recording Processor URI** parameter in the IVR profile.

Shutting down RPS

Before shutting down RPS, recover any lost recordings after the RPS has processed existing calls. For more information, see [Recovering Metadata for SpeechMiner](#). After ensuring that the Voice Processor is processing data as expected, shut RPS down. If there are any GIR ICONs used by RPS, shut them down as well.

Monitoring and troubleshooting information

The Voice Processor provides detailed health and performance information on the endpoint `<domain:port>/api/status`.

The following optional query parameters allow you to request specific health reports:

- `?verbose=1` provides a summary for each tenant and service.
- `?ccid=<HTCC ID>` provides a detailed report for a single tenant.
- `?service=<service name>` provides a detailed report for a single service. The following services are available for querying:
 - `persistence` provides information about the health and performance of the Voice Processor database.
 - `ccSettings` provides a status on the connection to RWS and the validity of the RWS settings.
 - `gim` provides health and performance information of the GIM database.
 - `rws` provides a status on RWS in the context of posting recording metadata.
 - `sm` provides health and performance information on data posted to SpeechMiner Interaction Receiver.
 - `schedRecovery` provides health and performance information on the internal scheduled recovery service which retries failed tasks periodically.
 - `mcpPosts` provides health and performance information on handling of incoming posts from MCP.

For example, if posts are not reaching SpeechMiner, a query to `/api/status?verbose=1` should provide sufficient information to isolate the problem. Additionally, you can provide a snapshot of the output from this query when you contact Genesys Customer Care for assistance.

GIR Voice Processor deployment using Podman

This deployment is applicable for Voice Processor 9.0.000.39 or higher for installing GIR VP using Podman.

Currently GIR Voice Processor is deployed using Docker swarm in Premise and through Elastic Container Service in AWS. Due to known issues with Docker Swarm and Docker in RHEL 8, GIR Voice Processor is moving towards Podman for deploying Voice Processor.

Limitations

- Currently there are no alternatives for Docker swarm features with Podman. We may have to use a load balancer for load balancing with Podman.
- Network mode *overlay* is not supported in Podman, we will use *host* network mode.

Prerequisites

- Confirm Podman 4.9 or higher.
- Confirm Podman-compose 1.2.0 or higher.
 - `podman --version`
- PostgreSQL - one of the following:
 - From Voice Processor 9.0.000.40 (or higher), install PostgreSQL 16.4.
 - From Voice Processor 9.0.000.39 (or lower), install PostgreSQL 12.11.
- Genesys Info Mart 8.5 or higher installed on Microsoft SQL Server or PostgreSQL. For information on the system requirements for GIM, see [Genesys Info Mart Requirements](#).
- Interaction Recording Web Services (RWS) 8.5.205.32 or higher.
- SpeechMiner 8.5 or higher if you are using SpeechMiner. We recommend that you install Speechminer before deploying Voice Processor.

We recommend that you have the following details before proceeding with deployment:

- Host name, port, database name for Genesys Info Mart, and user (read-only) credentials.
- Host name and port for the Interaction Recording Web Services (RWS), and Operation Admin (ops) credentials.
- Configuration Manager credentials for an account with access to the IVR Profile.

- Host name, port, and credentials needed to post to SpeechMiner Interaction Receiver. These details are required only for new installations.

Preparing Podman environment

Extracting installation files

You can download the GIR VP image from the Genesys customer portal. The GIR VP image is a .tar file that contains the installation and configuration files required to set up and run the Voice Processor. Extract and copy the files from the image using the following steps:

1. Load the GIR VP image.
`zcat <.tar file> | podman load`
2. View the list of container images and make a note of the newly loaded image.
`podman image ls`
3. Add a custom tag to the image for your reference.
`podman tag <image ID> <tag>`
4. Copy the files from the image.
`id=$(podman run --rm -dt <image> cat) && podman cp $id:/rps/compose . && podman stop $id`

In the above command, `--rm` option will delete the image file after creating the Podman container.

Now you have the sample configuration files in **./compose/defaults**, an Ansible playbook in **./compose** to help you set up and run the Voice Processor, and an SQL file for database setup. You will need these files for installing and configuring the Voice Processor.

Podman logs

Podman logs location can be find using:

```
podman inspect --format='{{.HostConfig.LogConfig.Path}}' <container-id>
```

Podman container logs can be accessed simply by `podman logs <container name>`.

Configuring Voice Processor

This section contains the following sub-sections:

- [PostgreSQL database configuration](#)
- [Service level configuration Documentation:CR:Solution:VP:8.5.2](#)
- [Genesys Voice Platform profile configuration Documentation:CR:Solution:VP:8.5.2](#)
- [Tenant level configuration Documentation:CR:Solution:VP:8.5.2](#)
- [GIM DB ETL configuration Documentation:CR:Solution:VP:8.5.2](#)

PostgreSQL database configuration

The Voice Processor requires a service-specific database that tracks work in progress items. This database runs on a PostgreSQL server. Set up the database using the following steps:

1. Create a database in your PostgreSQL server for the Voice Processor.
2. Create a PostgreSQL user and grant all privileges to the database that you created in the previous step.
3. Assign a password to the user that does not contain a backward slash (\) or quotation marks, as they might cause issues later.
4. Make a note of the database name, user name, and password — they are needed when configuring the Voice Processor in later steps.
5. Confirm that the **standard_conforming_strings** parameter of the PostgreSQL server is set to on (default).

Important

- GIR Voice Processor must have a separate PostgreSQL DB from Config Server since the Voice Processor PostgreSQL DB requires the `standard_conforming_strings` setting to be on and the Config Server PostgreSQL DB requires the `standard_conforming_strings` setting to be off.
- Voice Processor supports connections to PostgreSQL DB when `password_encryption` is set to md5 or scram-sha-256 in the **postgresql.conf** file.
- When using a PostgreSQL DB for Genesys Info Mart, if `password_encryption` is set to scram-sha-256 in the **postgresql.conf** file, the Genesys Info Mart version must be 8.5.016.04 or higher.

6. Run the provided script, `create_node_rps_tables_v2.sql`, against this new database to provision it.

Important

To avoid possible conflicts with their settings requirements, Genesys recommends not hosting the Voice Processor and Configuration Server databases on the same PostgreSQL instance.

Service level configuration

You can follow the instructions provided with the configuration files available in the default directory. You can copy the provided configuration files and make changes to your copies. We recommend that you use a version control repository to store your configurations. Add the PostgreSQL database, user name, and password to the **nodeRpsDb** setting in your copy of **settings-override.json**.

Voice Processor database settings

To enable TLS connection to the Voice Processor database, set the **ssl** parameter to **true** and configure the **trustedCA** parameter under **nodeRpsDb** in **settings-override.json**.

```
nodeRpsDb:
  database: <database name>
  host: <db server hostname>
  port: <db port>
  user: <db user>
  ssl: < true / false >
  trustedCA: false / true / "<path to root certificate>"
```

The **ssl** parameter is optional and its default value is **false**. When you set it to **true**, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to **true**, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is **false**.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is **true**.
- Authenticate the server certificate against the specified root authorities. Set **vpdb_ca_cert** in your copy of **settings-override.json** with the **<path to root certificate>** value.
- While enabling TLS on the Voice Processor service, please validate that Podman has access to certificate files.

Voice Processor HTTPS settings

The **rwsBaseUri** setting in **settings-override.json** supports HTTPS. For example:

```
https://<RWS hostname>:<RWS port>
```

To use HTTPS on the Voice Processor service API, set **https** to **true** in your copy of **settings-override.json**. You must provide the server private key, public key, and path to the files.

```
https: true
tls:
  privkey: <path to the private key file>
  pubkey: <path to the public key file>
```

MCP post basic authentication

Add the following lines to the **settings-override.json** file to enable basic authentication for the endpoint used by the MCP to post recording metadata:

```
authUsername: "<basic auth username>"
authPassword: "<basic auth password>"
```

If you add these options, you must also configure the Voice Platform profile option, **recording client.callrec_authorization**, in the **[gvp.service-parameters]** section to match these credentials. As basic authentication involves sending the credentials in plain text format, we strongly

recommend that you use TLS for maximum security. Note that the other Voice Processor endpoints are not authenticated. Therefore, you must install the Voice Processor behind a firewall or API gateway to restrict access. You can obtain a summary of endpoints exposed by the Voice Processor service by accessing: `http://<GIR VP hostname>:<port>/apidoc`

Setting the GR Voice Processor image

To find the values for `<image ID>:<tag>`, use the `podman image ls` command.

Genesys Voice Platform profile configuration

Use HTTPS protocol in the Voice Processor URL when HTTPS is enabled in the Voice Processor service API.

```
recordingclient.callrec_dest = fixed,https://<VP hostname>:<VP port>/api/contact-centers/<CCID>/recordings/
```

Use HTTPS protocol in the SpeechMiner Interaction Receiver URL when HTTPS is enabled on the SpeechMiner Interaction Receiver. When using HTTPS for the SpeechMiner URL, by default, the Voice Processor does not validate SpeechMiner server certificate. You can set **sm_ca_cert** in your copy of **settings-override.json** with the `<path to root certificate>` value to authenticate the server certificate against the specified root authorities.

```
recordingclient.rp.speechminer_uri: fixed,https://<Speechminer backend hostname>/interactionreceiver/
```

Tenant level configuration

As the Voice Processor is designed to support Genesys cloud multi-tenancy model, settings that may vary from tenant to tenant are stored in an RWS group settings called **rps-provisioning**:

Important

You need an Ops Admin user account to access these settings. For more information on how to update settings in RWS, see [Settings API](#).

You must specify the Ops Admin user name and password in your copy of **settings-override.json**. The tenant level configuration values are set to the RWS group settings **rps-provisioning** using HTTP POST. For example:

```
curl -u <Ops admin user>:<password> -X POST -H "Content-Type: application/json" <rwsBaseUri>/api/v2/ops/contact-centers/<ccid>/settings/rps-provisioning -d @rps-settings.json
```

Where **rps-settings.json** contains settings like: `eventDataFilters`, `gimDb`, `rwsPostRecBaseUri` and others.

To confirm the Voice Processor per tenant settings, use HTTP GET. For example:

```
curl -u <Ops admin user>:<password> -X GET "<rwsBaseUri>/api/v2/ops/contact-centers/<ccid>/settings/rps-provisioning?location=*&ignoreParentLocations=false"
```

GIM database

You must provide information needed to access the tenant's GIM database. To enable TLS connection to the GIM database, set the **ssl** parameter to true and configure the **trustedCA** parameter under GIM database settings in tenant level configuration.

```
{
  "name": "gimDb",
  "value": {
    "primary": {
      "host": "<GIM server hostname>",
      "port": "<GIM server port (default 5432 for Postgres, 1433 for MS SQL)>",
      "user": "<DB user name >",
      "database": "<database name>",
      "password": "<DB user password>",
      "dbType": "<postgres or mssql, default postgres>",
      "ssl": "< true / false >",
      "trustedCA": false / true / "<path to root certificate>",
    },
    "backup": {
      < same settings as for primary >
    }
  }
}
```

The **ssl** parameter is optional and its default value is false. When you set it to true, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to true, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is false.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is true
- Authenticate the server certificate against the specified root authorities by performing the following steps:
 1. Set **gim_ca_cert** in your copy of **settings-override.json** with the <path to root certificate> value.
 2. Set **trustedCA** to /rps/rpsdata/gimCA in GIM database settings to be posted to tenant level configuration.

The **backup** parameter is optional. You can omit it if there is only one GIM database available.

RWS posting

You must specify the RWS instance to which recordings are posted. As this is a region-based setting, multi-regional deployments can ensure that recording data stays within the jurisdictional boundaries. The Voice Processor instance selects the location identified through the nodePath of the RWS server from which the setting is retrieved or the nearest matching parent. The **backup** parameter is optional. The URL supports HTTPS. When using HTTPS for the RWS URL, by default, the Voice Processor does not validate RWS server certificate. You can set **rws_ca_cert** in your copy of **settings-override.json** with the <path to root certificate> value to authenticate the server certificate against the specified root authorities.

For example, the following setting applies to all Voice Processor instances:

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

The following setting would override the above global setting for Voice Processor instances that retrieved the setting from an RWS node with nodePath /US or /US/* :

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/US",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

Event filtering

You can use filters to remove unwanted data from the recording metadata. The event filtering settings are similar to RPS except the mechanism of how the default filters are disabled.

```
{
  "name": "eventDataFilters",
  "value": {
    "attachedDataFilter": "regex for new attached data filter",
    "attachedDataFilterException": "regex for new attached data filter exception",
    "acwCustomDataFilter": "regex for new ACW data filter",
    "acwCustomDataFilterException": "regex for new ACW data filter exception"

    -- or, to disable the default filters or filter exceptions --

    "disableAttachedDataFilter": true,
    "disableAttachedDataFilterException": true,
    "disableAcwCustomDataFilter": true,
    "disableAcwCustomDataFilterException": true
  }
}
```

The default filters are:

- **attachedDataFilter:** ^ORSI:|^WWE|^PegAG
- **attachedDataFilterException:** ^(GRECORD_(PARTITIONS|PROGRAM)|GSRS_STATE|GSIP_REC_FN)\$
- **acwCustomDataFilter:** ^ORSI:|^WWE|^PegAG
- **acwCustomDataFilterException:** ^(GRECORD_(PARTITIONS|PROGRAM)|GSRS_STATE|GSIP_REC_FN)\$

Complete after-call work (ACW) threshold

The ACW threshold indicates how long the Voice Processor waits, in minutes, following the end of an interaction to update custom data. Custom data entered by agents after this interval is not added to

recording metadata. The default value is zero.

```
{
  "name": "acwThresholdMinutes",
  "value": <ACW wait interval in minutes>
}
```

GIM DB ETL configuration

You must configure the GIM ETL application properly to ensure recording metadata is posted from the Voice Processor to SpeechMiner in a timely manner.

The **etl-start-time**, **etl-end-time**, and **etl-timezone** options in the **[schedule]** section are used to configure a daily maintenance period during which population of GIM data is paused for maintenance purpose. New recordings posted to the Voice Processor during this period are not processed and they are held temporarily in a database until the maintenance period finishes and the relevant GIM data becomes available. You must configure the **maintain-start-time** option such that the GIM ETL maintenance job begins and completes during the maintenance period.

The **etl-frequency** option in the **[schedule]** section is used to specify the cycle time of the GIM ETL jobs that populate the recording metadata used by the Voice Processor. We recommend that you use the default value of one minute. Note that any time longer than 3 minutes may cause subsequent delays in recording posts. If a longer **etl-frequency** setting is used, then the value of the Voice Processor service setting, **rpsInitialInteractionTimeout**, should be increased accordingly.

The **user-event-data-timeout** option in the **[gim-etl]** section is used to ensure that custom attached data entered during after-call work is captured. You can increase the default value of one hour if your agents will spend more than a few minutes in after-call work.

Important

Consult Genesys before setting non-default values for the following options.

The **max-call-duration**, **merge-failed-is-link-timeout**, and **extract-data-stuck-threshold** options in the **[gim-etl]** section must be configured properly to ensure completeness of the call metadata recorded in GIM. For more information on these options, see [Operations-Related Options for Genesys Info Mart](#).

Deploying and Starting Voice Processor

Deploy Voice Processor to the newly configured Podman image, referencing your copies of the default configuration files. This step also starts Voice Processor.

Before deploying, the **settings-override.json** file (or the yaml files you have designated to provide these settings) have several mandatory parameters that must be configured, as described below.

In the **settings-override.json** file, the following parameters are required:

- **rwsBaseUri** – Specifies the address of the RWS cluster that will provide Voice Processor with contact center settings, including tenant-specific configurations such as Genesys Info Mart (GIM) database information and ACW Wait Time. Example: `<code>http://some-rws-host.com:8090</code>`

- **region** – Controls the region section in the metadata POSTed to RWS. This must match the **crRegion** setting of the RWS cluster to which recordings are posted. Example: usa
- **nodeRpsDb** – This section specifies the Voice Processor Persistence Database, which is required for storing recording metadata while Voice Processor is processing them.
 - **database** – Database on the host that will hold recording metadata. Example: noderpsdb
 - **host** – Host of the Persistence Database. Example: noderpsdb.com
 - **port** – Port that the Persistence Database is listening on. Example: 5432
 - **user** – The user name to connect to the Persistence Database.
 - **password** – The password to connect to the Persistence Database.
- **rwsUsername** – The user name for authenticating with RWS.
- **rwsPassword** – The password for authenticating with RWS.

For an example of how the yaml files should be structured, you can refer to the default yaml files that were included with Voice Processor. These files are located at **<INSTALL_DIR>/defaults/**, where **<INSTALL_DIR>** is the location where you extracted the installation files to during the **Preparing your Podman environment** step.

After configuring the default configuration files, deploy and start Voice Processor:

```
sudo ansible-playbook \
  -e docker_config=<defaults directory of Your Docker Configuration Yaml (e.g. docker-
  config.yaml)> \
  -e logger_config=<defaults directory of Your Logger Configuration Yaml (e.g. logger-
  config.yaml)> \
  -e settings_override=<defaults directory of Your Voice Processor Configuration Yaml
  (e.g. settings-override.yaml)> \
  -e secrets=<defaults directory of Your Voice Processor Secrets Yaml (e.g.
  secrets.yaml)> \
  -e service_user=<user name under which gir vp service is running>
  -e service_group=<group name under which gir vp service is running>
  gir-vp-playbook.yml
```

Important

If the above options are not specified, then the .yaml files in **./compose/defaults** will be used.

Before starting the service, ensure proper permissions are set for the certificates and working directory:

1. Set ownership and permissions for the certificate directory:


```
sudo chown -R 1101:1101 <working_directory>/rpsdata
sudo chmod 755 <working_directory>/rpsdata
```
2. Set permissions for key files:


```
sudo chmod 644 <working_directory>/rpsdata/privkey
sudo chmod 644 <working_directory>/rpsdata/pubkey
```


3. Restart the service after permission changes:
`sudo systemctl restart girvp`

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }' girvp.company.com/api/active-version
```

GIR VP container will run as a Systemctl service: `girvp.service`.

When a container stopped, Systemctl will start a new Podman container:

```
sudo systemctl status girvp.service
```

Validating

1. Place a call to an agent or a test agent that is configured for recording.
2. Verify that the call arrives at the SpeechMiner UI. It should take 5 to 15 minutes depending on your configured ACW wait setting.
3. Assuming that live traffic is not recorded, you can use the health check endpoint `<domain:port>/api/status?verbose=1`. The items *recordingsInProgress* or the MCP Post operational status can be helpful in determining whether or not the recording is arriving at the Voice Processor. This also helps you to isolate GVP configuration problems from problems with the Voice Processor service. If a load balancer is used, the node serving the health check may not be the one that handled the recording. Therefore, several health checks may be required to cover the whole cluster.

Upgrading

Docker object configurations and secrets cannot be upgraded. We recommend that you remove the container, update the required configurations, and redeploy.

```
podman container rm <gir_vp>
```

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example `curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }' <hostname/Ip address>:8889/api/active-version`

Miscellaneous Podman tips

- To view a list of running containers:

```
podman ps
```

- To view a list of all containers (started/stopped):

```
podman ps -a
```

- To view the container logs:

```
podman logs <container name>
```

- To remove everything to start again:

```
podman stop <gir_vp> podman container rm <gir_vp>
```

Load balancing with Podman

GIR VP supports load balancing with Nginx and httpd. To set up load balancing in a Premise environment, see [Setting up the Load Balancer in a Single-Tenant Environment](#).

Deploying Genesys Voice Platform for GIR

Genesys Voice Platform (GVP) provides the media services, including IVR, that GIR needs to record contact center interactions.

Installing GVP

Install and configure the GVP solution as described in the [GVP 8.5 Deployment Guide](#). You can learn more about GVP [here](#).

Configuring GVP

GVP uses four components and functions that require additional configuration to enable recording for GIR:

- [Resource Manager](#)
- [IVR Profile](#)
- [Logical Resource Group](#)
- [Media Control Platform \(MCP\)](#)

Resource Manager

1. In the GVP Resource Manager application, configure the following parameters:

Section Name	Parameter Name	Value
rm	conference-sip-error-respcode	Set to 503.
	resource-unavailable-respcode	Set to 603
monitor	sip.proxy.releaseconfonfailure	Set to false.

2. For each GVP shared tenant, a separate tenant is required by Resource Manager. Create a gateway resource for each tenant RM tenant using the SIP Server source address.

IVR Profile

Important

By default the profile named record is used for recording purposes. For IVR recording, the recording parameters associated with the record profile are combined with the

existing IVR profile that is used for the IVR functionality. For additional information, refer to [IVR Recording](#).

1. In Genesys Administrator Extension, navigate to **Configuration > Voice Platform**, select **Voice Platform Profiles**, and click **New**.
2. On the **General** tab, enter the following parameters:
 - **Name** (Genesys recommends naming it record)
 - **Display Name**
 - **Description**
3. On the **Options** tab,
 - Configure for basic authorization:
 - In the **[gvp.service-parameters]** section, set the **recordingclient.callrec_authorization** parameter to fixed, rp_username:rp_password.

Important

The rp_username:rp_password value must be the same username and password that are configured for authorization in the Recording Processor Script or Voice Processor. For more information, see [Recording Processor Script](#) or [Voice Processor](#).

- Configure the following in the **[gvp.service-parameters]** section to set the bitrate and to determine if MP3 recording is mono or stereo:
 - For 8 kbit/s mono:**
 - Set the recordingclient.gvp.config.mpc.mp3.bitrate parameter to fixed,8.
 - Set the recordingclient.channels parameter to fixed,1.
 - For 16 kbit/s stereo:**
 - Set the recordingclient.gvp.config.mpc.mp3.bitrate parameter to fixed,16.
 - Set the recordingclient.channels parameter to fixed,2.
 - For 32 kbit/s stereo:**
 - Set the recordingclient.gvp.config.mpc.mp3.bitrate parameter to fixed,32.
 - Set the recordingclient.channels parameter to fixed,2.
- 4. On the **Recording** tab, add the Recording Certificates, and set the parameters. **[+] Show the table describing the parameters.**

Section	Parameter Name	Description
Recording Destinations	Storage Destination	The path for recording storage on the WebDAV Server. For example, http://<webdav>/recordings.
	Storage HTTP Authorization Header	The credentials for the WebDAV Server. The format is username:password. This field is visible only if the Storage Destination begins with either http or https.

Section	Parameter Name	Description
	Recording Processor URI	The URI that MCP uses to post the metadata of the audio recording after the recording is complete. MCP uses HTTP POST to send the metadata to the Recording Processor or Voice Processor. The format for this parameter is: http://<Recording Processor Host>/api/contact-centers//recordings/. Note: The value for the URI must always end with a forward slash (/).
	SpeechMiner Interaction Receiver	Specifies the URL that points to the SpeechMiner Interaction Receiver responsible for accepting metadata from the Recording Processor Script or Voice Processor for this profile, for example, http://<SpeechMiner Host>/interactionreceiver.
	SpeechMiner Interaction Receiver Authorization Header	Specifies the credentials required to connect to the SpeechMiner Interaction Receiver used by the Recording Processor Script or Voice Processor associated with this profile. The format is username:password, where the username and password are the Interaction Receiver credentials. Note: The user and password value must be the same as the username and password configured in both of the following sections: <ul style="list-style-type: none"> • Configuring SpeechMiner settings in RWS. • Step 5 of Configuring SpeechMiner users.
Speech Analytics Parameters Note: Leave these parameters empty unless you have purchased and enabled speech analytics mode on SpeechMiner; otherwise, recording may not operate correctly.	SpeechMiner Destination	Specifies the URL that points to the SpeechMiner Interaction Receiver responsible for accepting analytics files for this profile, for example, http://<SpeechMiner Host>/interactionreceiver. This is an optional parameter and should be left empty if

Section	Parameter Name	Description
		speech analytics is not enabled.
	SpeechMiner HTTP Authorization Header	Specifies the credentials required to connect to the SpeechMiner Interaction Receiver used for accepting analytics files for this profile. The format is username:password, where the username and password are the Interaction Receiver credentials for analytics. This field is visible only if the SpeechMiner Destination begins with either http or https.
Additional Recording Parameters	Recording Storage MIME Type	The audio file type used for the storage recording. Set to audio/mp3.
	Recording Alert Tone Source (Optional)	The URI of the audio tone. For example, http://example.com/tone.wav.
Recording File Name Template	File Name Template	<p>Specifies the name of the template used for generating the MSML recording. When left blank, the default value is \$id\$. Choose any, or all of the following parameters:</p> <ul style="list-style-type: none"> • ID—The unique identifier of the template. • Date Time—The date and time of the call in which the recording is started. The date and time is sent in ISO format with UTC time. The ISO format is YYYY-MM-DDTHH:MM:SSZ • MCP Date Time—The local date and time of the call in which the recording is started. The local time follows the MCP instance where the recording is taking place. • SIP Server Application Name—The SIP Server application name in which the recording is started. • Call UUID—The call UUID of the call in which the

Section	Parameter Name	Description
		<p>recording is started.</p> <ul style="list-style-type: none"> • ANI—The ANI information of the call in which the recording is started. • Connection ID—The TLib Connection ID of the call in which the recording is started. • DNIS—The DNIS information of the call in which the recording is started. • Agent ID—The agent ID of the DN of the call in which the recording is started. If the recording has not started because the DN or Agent ID has not logged in, this parameter will not be present. <p>For example, if DNIS, ANI and Agent ID are selected, the File Name Template is set to \$dnis\$_\$ani\$_\$agentId\$.</p> <p>Note: Using too many parameters could exceed the 260 characters limit for a Windows file name.</p>

Using multiple locations

A Recording IVR profile enables you to set up a separate voice recording storage location, per data center location, based on the SIP Server geo-location. To use this functionality, create a separate IVR Profile for each geo-location, as follows:

1. Set the following parameters in the **[gvp.general]** section:
 - **service-type**=record
 - **geo-location** (that is, the geo-location that identifies SIP Server location).
2. For each new IVR Profile configure separate Recording Destinations:
 - Storage Destination: Set to the recording storage location for the corresponding data center.
 - Recording Processor URI: Set to the Recording Processor or Voice Processor address for the corresponding data center.
 - SpeechMiner Interaction Receiver: Set to the SpeechMiner Interaction Receiver in the primary data center.

For additional information about multiple data center locations, refer to the [Multiple data center locations](#) page.

Logical Resource Group

A single Media Control Platform (MCP) pool can be used to provide all types of media services including call recording. A dedicated Logical Resource Group can also be used for call recording.

1. Modify a Logical Resource Group to include call recording:
 - Set the **service-types** option to `voicexml;conference;announcement;cpd;media;recordingclient`.
2. Create a new Logical Resource Group. In the **[gvp.lrg]** section, set the following parameters:

Parameter Name	Value
service-types	recordingclient
load-balance-scheme	round-robin
monitor-method	option
port-usage-type	in-and-out
resource-confmaxsize	-1

Important

If using a dedicated Logical Resource Group, ensure that the `recordingclient` value is removed from the MCP pool's **service-types** parameter. For example, set the service type to `voicexml;conference;announcement;cpd;media`.

Media Control Platform

1. Ensure that the Media Control Platform (MCP) instances are included on the **Connections** tab of the Resource Manager Application object.
2. In the **[mpc]** section, set the **default_audio_format** parameter to ULAW, or ALAW, depending on the G711 settings.
3. In the **[mpc]** section, set the **mediamgr.recordwritetimeinterval** parameter to 10000 (10 seconds). The default value is 1000 milliseconds(1 second).
4. In the **[mpc]** section, set the **recordpostretrybackoff** parameter based on the time required to initialize the Recording Processor Script (RPS) or Voice Processor, which depends on the number of agents in the deployment, and how long it takes to retrieve agent information from the configuration environment through the Configuration Server. The initialization time can be determined by examining the RPS log or Voice Processor log and looking for an entry containing "INFO Recording processor is listening on" which indicates that the RPS or Voice Processor is fully initialized. Genesys recommends that the value be set to approximately half the time required for this initialization to complete. For example, if it takes 200 seconds for RPS or Voice Processor initialization to complete, **recordpostretrybackoff** should be set to 100 seconds. Note that this parameter is specified in milliseconds.

Important

- When assigning the MCP(s) for handling call recording, the IP address and Port must match the details of the MCP. Set the **max ports** option to double the number of calls that you want to handle with the MCP. One port is used per stream in the call, one for the customer leg and one for the caller leg. If **max ports** is set to 1000, the MCP can handle 500 calls.
- If screen recording is used, make sure the clock is synchronized to the same time as the agent desktop machines where the Screen Recording Service is installed.
- The **[mpc].recordnumparallelpost** parameter is set to 30 by default and it does not need to be changed during normal operation. However, in a scenario where MCP is posting high number of files to Recording Processor Script (RPS) or Voice Processor and WebDAV, it is recommended to set the value of this parameter based on the sizing calculation: $value2/value1$ where:
 - *value1*: The number of concurrent uploads the WebDAV is able to handle
 - *value2*: The number of recordings that MCP will be posting to WebDAV

For more information about the GVP and Media Server options, see the [Media Control Platform](#).

Encrypting and Provisioning Certificates

Before you configure encryption certificates for voice and screen recordings, you must generate the following keys and certificates:

- A certificate for the Certificate Authority (CA) in .pem format.
- A recording certificate (also known as public key) in .pem X.509 RSA format.
- A recording private key in .pem format.

Important

It is your responsibility to store your private keys and certificates, including the expired ones. You must also back up your keystore, keystore password, certificates and private keys in a secure location offsite to protect against site level disasters. When Genesys Interaction Recording encryption is enabled, loss of the keystore and private key would result in loss of recording files.

While renewing the certificates, keep your old certificates under Administration - Recording Certificates and provision the new certificates using the instructions provided in this section. This will ensure the playback of recordings encrypted with the older certificates without any issues.

Generating the Certificates and Keys

This certificate must meet the following requirements:

- 2048 bit RSA (or higher; please align encryption strength requirements with your IT Security)
- x509 certificate
- PEM format
- The certificate must be signed by a trusted third-party CA, self signed or signed by your own private CA
- If using a third-party CA, the certificate signing request provided to the third-party CA must contain the Subject Name, Serial Number, Subject DN, and Issuer DN. You might be contacted by the third-party CA who might ask for additional information
- The certificate validity period of the certificate determines when the next certificate needs to be generated for renewal

The following OpenSSL command to generate certificate signing request and private key is an example:

```
openssl req -nodes -newkey rsa:2048 -keyout private_key.pem -out cert.req -days <validity period>
```

The system prompts for DN fields to be filled in. You must fill in all of them. See the table below for

the details.

DN Field	Explanation	Example
Common Name	Name of your Recording Solution	Interaction Recording
Organization	The exact legal name of your organization. Do not abbreviate your organization name.	Monster & Sons, Inc.
Organization Unit	Section of the organization.	Robot Repairs
City or Locality	The city where your organization is legally located.	Pleasant Hill
State or Province	Full state or province where your organization is legally located.	California
Country	The two-letter ISO abbreviation for your country.	US

The files will have the following:

- `private_key.pem`— the private key that is used to decrypt the recordings. It must be kept safe and should not be shared.
- `cert.req`— the certificate signing request for the third-party CA that signs the request and provides the public key certificate to be used to encrypt the recordings.

Chained Certificates

Genesys recommends that the recording certificate that you want to use for Genesys Interaction Recording encryption be signed by a single trusted third-party CA.

Important

Chained certificates are certificates where the trusted third-party CA is used to sign the intermediate CA certificate, and the intermediate CA certificate is then used to sign the user certificate.

To set up a chained certificate:

1. Upload the certificate using Genesys Administrator Extension.
2. Obtain the CA file and place it in the MCP's local directory—for example, `/genesys/mcp/certificates/<tenant name>/<ca-file>`. Note that the CA file given here should be the bundle of all the intermediate CA's and the root CA in specific order—for example, `cat crt_inter3.pem crt_inter2.pem crt_inter1.pem root_ca.pem > ca.pem`. When you create a bundle from separate certificates, take note that these certificates might sometimes have additional information that should not be in the final bundle file. If this is the case, the above command (`cat`) will not work, and the information should be copied using an editor that opens the file using the Unix end of line. The information that should be taken starts from:
-----BEGIN CERTIFICATE-----
and finished with the line:
-----END CERTIFICATE-----

3. Configure the CA file path in IVR profile. In the `gvp.service-parameters` section, set the `recordingclient.gvp.config.mpc.mediamgr.CA_file` parameter to `fixed,/genesys/mcp/certificates/<tenant name>/<ca-file>`

For Call Recordings

A Recording Certificate binds a public encryption key to a particular recorded message identity.

Important

- When configuring encryption, backup of the private key is your responsibility. If the private key becomes lost or corrupt, any recording encrypted using that key will become unusable.
- If screen recording is also used in the deployment, it is required that a screen recording certificate is also provisioned. Otherwise, the Recording Muxer Script will not be able to mux the call recording and screen recording together, if the call recording is encrypted but the screen recording is not encrypted.

The following steps describe how to configure encryption for voice recordings:

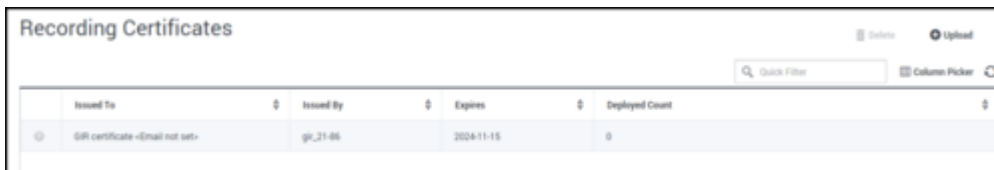
Prerequisites

- A certificate for the Certificate Authority (CA) in .pem format—for example, `ca_cert.pem`.
 - A recording certificate (also known as public key) in .pem format—for example, `02_gir_cert.pem`.
 - A recording private key in .pem format—for example, `02_gir_priv_key.pem`.
1. On the machine where the Recording Crypto Server is installed, place the Certificate Authority (`ca_cert.pem`) in the `<Recording Crypto Server Install Directory>\RCS` directory.
 2. Edit the **rcs.properties** file:
 - a. Change the value of the **cacertstorepath** parameter to `ca_cert.pem`.
 - b. Set the value of the **cacertstorepassword** parameter to the valid password.
 3. Restart the Recording Crypto Server.
 4. Using Recording Plug-in for GAX, edit all your Media Control Platforms (MCP):
 - On the **Options** tab of each MCP application object, in the **[mpc]** section, set the **mediamgr.CA_file** parameter to the location of the Certificate Authority file (for example, `c:\keystore\ca_cert.pem`).
 5. Restart all the MCP instances.

For an example of a certificate, see [Sample Certificate and Key File Generation](#). You are now ready to upload and deploy your certificates to complete the encryption process.

To upload a new certificate:

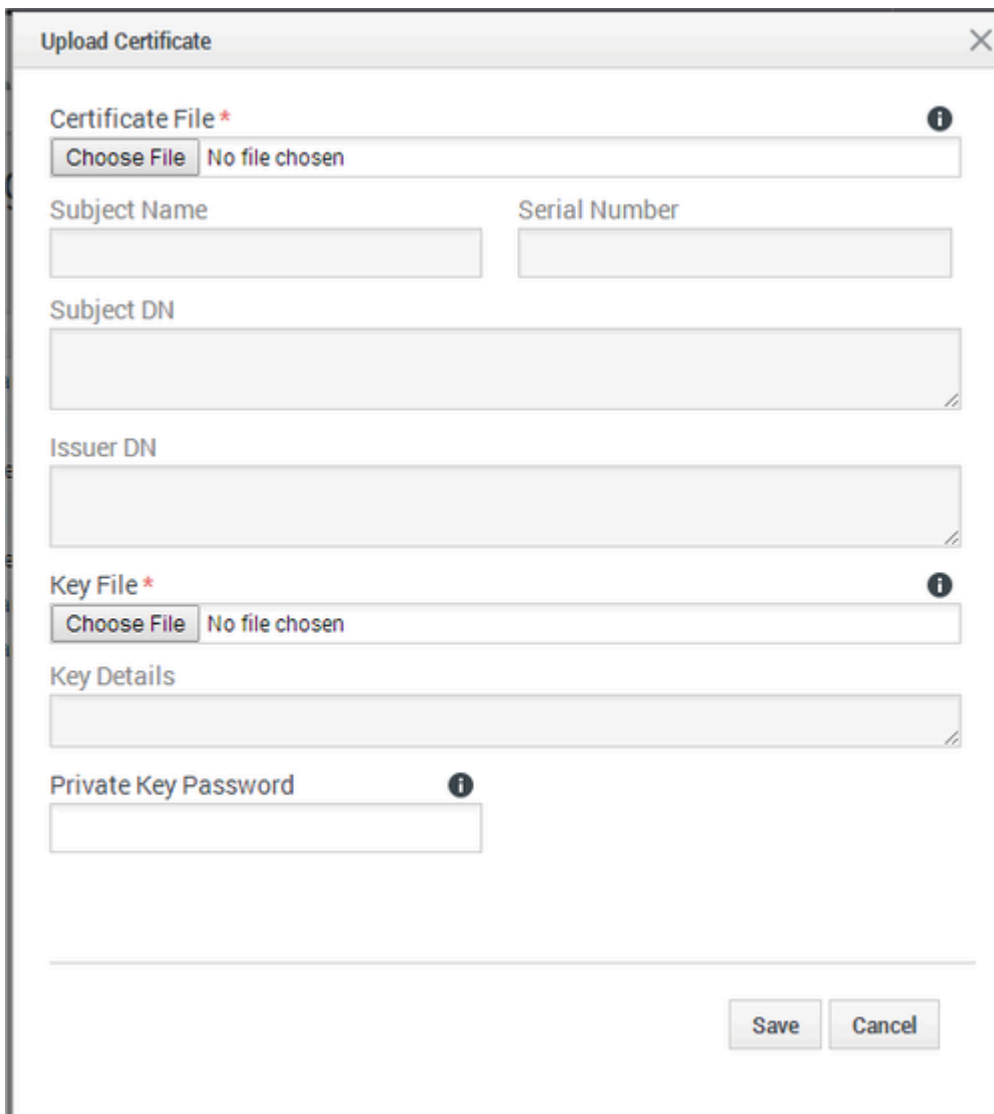
1. Log in to Genesys Administrator Extension, and navigate to **Administration > Certificates**.



The screenshot shows the 'Recording Certificates' panel in the Genesys Administrator Extension. It features a table with columns: Issued To, Issued By, Expires, and Deployed Count. A single row is visible with the following data: Issued To: 'GRI certificate (Email not set)', Issued By: 'gr_21-06', Expires: '2024-11-15', and Deployed Count: '0'. Above the table, there are buttons for 'Delete' and 'Upload', a 'Quick Filter' search bar, and a 'Column Picker' icon.

Issued To	Issued By	Expires	Deployed Count
GRI certificate (Email not set)	gr_21-06	2024-11-15	0

2. On the **Recording Certificates** panel, click **Upload**.



The 'Upload Certificate' dialog box contains several input fields and buttons. At the top, there's a 'Certificate File' section with a 'Choose File' button and 'No file chosen' text. Below this are fields for 'Subject Name', 'Serial Number', 'Subject DN', and 'Issuer DN'. The 'Subject Name' and 'Serial Number' fields are linked, as are 'Subject DN' and 'Issuer DN'. Next is a 'Key File' section with another 'Choose File' button and 'No file chosen' text. Below that is a 'Key Details' field. At the bottom is a 'Private Key Password' field. 'Save' and 'Cancel' buttons are at the bottom right.

Upload Certificate

Certificate File *

Choose File No file chosen

Subject Name Serial Number

Subject DN

Issuer DN

Key File *

Choose File No file chosen

Key Details

Private Key Password

Save Cancel

3. On the **Upload Certificate** panel, in the **Certificate File** section, click **Choose File**.
4. Select the appropriate file. This file must contain an X.509 RSA certificate in PEM format. The **Subject Name**, **Serial Number**, **Subject DN**, and **Issuer DN** fields automatically populate.
5. In the **Key File** section, click **Choose File**.

6. Select the appropriate file. The file must contain an RSA private key in PEM format. The encoding can be in either OpenSSL RSA private key or PKCS8 format. The **Key Details** field automatically populates.

Upload Certificate [X]

Certificate File * [i]
Choose File cert1.pem

Subject Name
Cert1 User <cert1.user@genesyslab.com>

Serial Number
1

Subject DN
C=CA,ST=Ontario,L=Markham,O=Genesys Telecommunications Laboratories,CN=Cert1 User,E=cert1.user@genesyslab.com

Issuer DN
C=CA,ST=Ontario,L=Markham,O=Genesys Telecommunications Laboratories,CN=Certificate Administrator,E=cert.admin@genesyslab.com

Key File * [i]
Choose File cert1.key

Key Details
Openssl format private key file

Private Key Password * [i]
[Empty Password Field]

[Save] [Cancel]

7. If the private key file is encrypted, enter the **Private Key Password**.
8. Click **Save**.

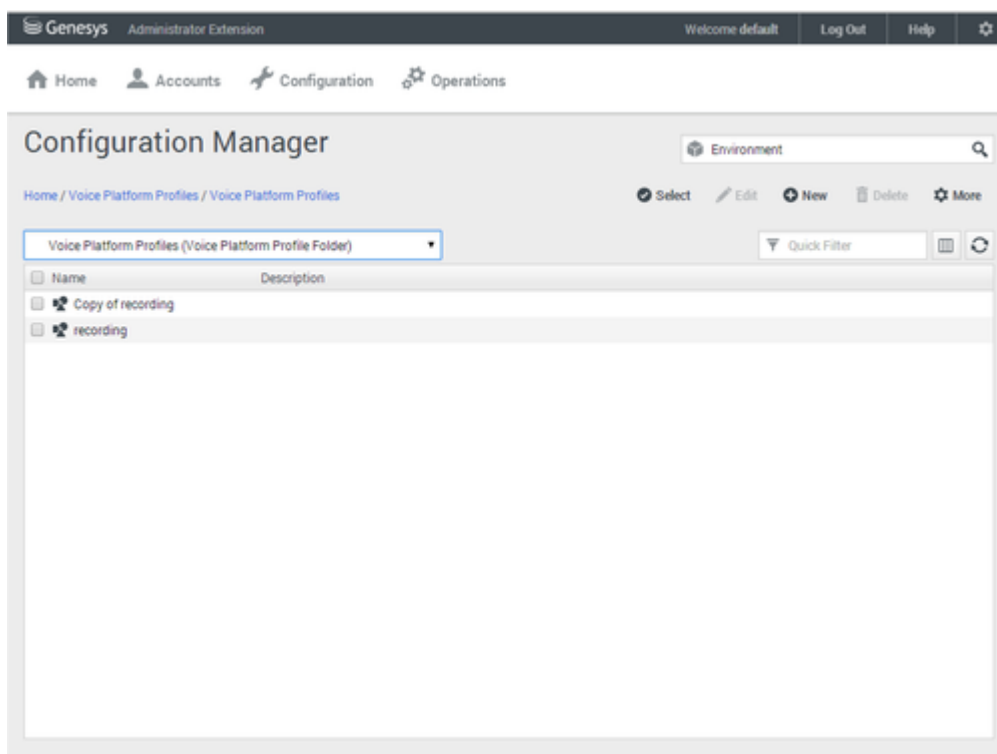
Important

- If you Upload and/or delete recording certificates in one Genesys Administrator Extension session, these changes are not reflected in another Genesys Administrator Extension session. You must log out and log in again to the second Genesys Administrator Extension session.

- If Recording Crypto Server (RCS) is restarted when a Genesys Administrator Extension user is logged in, the next Genesys Administrator Extension operation involving RCS fails because the RCS session saved by the Recording Plug-in for GAX does not exist. RCS will return a 401 "RCS is not available" error. The user must log out, and log in again when receiving the 401 "RCS is not available" error.

To deploy a new certificate:

1. Log in to Genesys Administrator Extension, and navigate to **Configuration > Configuration Manager > Voice Platform Profiles**.



2. From the **Voice Platform Profiles** screen, click the profile that you want to add the certificate to.

The screenshot displays the Genesys Administrator Extension interface. The top navigation bar includes the Genesys logo, 'Administrator Extension', and links for 'Welcome default', 'Log Out', 'Help', and a settings icon. Below this is a secondary navigation bar with icons for 'Home', 'Accounts', 'Configuration', and 'Operations'. The main content area is titled 'Configuration Manager' and shows a breadcrumb trail: 'Home / Voice Platform Profiles / Voice Platform Profiles / recording Properties'. To the right of the breadcrumb are icons for 'Clone', 'Delete', and 'Move To'. On the left, a sidebar contains a list of tabs: 'General', 'Options', 'Permissions', and 'Recording'. The 'Recording' tab is selected. The main form area contains the following fields and controls:

- Name ***: A text input field containing the value 'recording'.
- Display Name ***: A text input field containing the value 'Recording'.
- Description**: A large text area for additional details.
- Tenant**: A dropdown menu currently showing 'Environment'.
- State Enabled**: A checkbox that is checked.

At the bottom of the form are three buttons: 'Save', 'Apply', and 'Cancel'.

3. Select the **Recording** tab.

The screenshot shows the Genesys Administrator Extension Configuration Manager interface. The left sidebar contains a menu with 'General', 'Options', 'Permissions', and 'Recording'. The main content area is titled 'Configuration Manager' and shows the breadcrumb 'Home / Voice Platform Profiles / Voice Platform Profiles / recording Properties'. The 'Recording' tab is selected in the sidebar. The main content area is divided into three sections: 'Recording Certificates', 'Recording Destinations', and 'Additional Recording Parameters'. The 'Recording Certificates' section has an 'Add' button and a table with columns 'Issued To', 'Issued By', and 'Expires'. The 'Recording Destinations' section has fields for 'Storage Destination', 'Recording Processor URI', 'SpeechMiner Destination', 'SpeechMiner Interaction Receiver', and 'SpeechMiner Interaction Receiver Authorization Header'. The 'Additional Recording Parameters' section has radio buttons for 'Recording Storage MIME Type' (audio/wav and audio/mp3) and a field for 'Recording Alert Tone Source'. At the bottom, there are 'Save', 'Apply', and 'Cancel' buttons.

4. Click **Add**.
5. From the **Select Certificate** screen, select the certificate you want to add to the IVR Profile, and click **Add**.

Genesys Administrator Extension

Home Accounts Configuration Operations

Configuration Manager

Home / Voice Platform Profiles / Voice Platform Profiles / recording Properties

General Options Permissions **Recording**

Recording Certificates

Issued To	Issued By	Expires
Cert1 User <cert1.user@genesyslab.com>	Certificate Administrator	2024-04-32

Add Remove

Recording Destinations

Storage Destination

Recording Processor URI

SpeechMiner Destination

SpeechMiner Interaction Receiver

SpeechMiner Interaction Receiver Authorization H...

username password

Additional Recording Parameters

Recording Storage MIME Type

☐ audio/wav

☒ audio/mp3

Recording Alert Tone Source

Recording File Name Template

File Name Template

☐ ID ☐ SIP Server Application Name ☐ ANI ☐ DNIS

☐ Date Time ☐ Call UUID ☐ Connection ID ☐ Agent ID

☐ MCP Date Time

Save Apply Cancel

6. Click **Save**.

Important

In Genesys Administrator Extension, do not open **certificates-n** (where **n** is 1, 2, 3, and so on) options using the **Options** tab of the IVR Profile for editing. If opened for editing and saved without making any changes, the certificate will be corrupted. Instead, always use the **Recording** tab of the IVR Profile for certificate administration. To fix this issue, remove the certificate using the **Recording** tab of the IVR profile, add it again, and then save.

For Screen Recordings

Assigning Certificates

To assign a new certificate:

1. Using Genesys Administrator Extension, in the header, go to **Administration > Screen Recording Certificates**.
2. On the **Screen Recording Certificates** panel, click **Add**.
3. From the **Select Certificate** window, perform one of the following actions:
 - Select the check box next to the appropriate certificate, and click **Add**.
 - Click **Cancel** to discard any changes.
4. Perform one of the following actions:
 - Click the **Save** button to accept the changes.
 - Click the **Cancel** button to discard the changes.

Setting up the Decryption Proxy

1. Configure the Recording Crypto Server (RCS) locations that Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) uses for encrypted screen recordings:
 - For a single location:
 - a. Using a text editor, create the **create_single_location** file using the following command:

```
{
  "name": "decrypt-uri-prefix",
  "location": "/",
  "value": "<rcs uri>/rcs"
}
```

Important

Replace `<rcs uri>` with the appropriate value.

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_single_location http://<Web Services
Server>:8080/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
```

- For multiple locations:
 - a. Using a text editor, create the **create_first_location** file using the following command:

```
{
  "name": "decrypt-uri-prefix",
  "location": "<node_location>",
  "value": "<rcs uri>/rcs"
}
```

```
}
```

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_first_location http://<Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>/settings/screen-recording --header "Content-Type: application/json"; echo
```

Important

Replace <node_location> with the appropriate value. The values for the <node_location> are similar to the **nodePath** settings in the Interaction Recording Web Services (Web Services) **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier, refer to the **nodePath** setting in the **server-settings.yaml** file instead), but allow a hierarchical representation. For example, an Interaction Recording Web Services (Web Services) node uses a **decrypt-uri-prefix** setting with a location of "/US" if the **nodePath** set to "/US/AK" or "/US/HI".

- c. Repeat steps a and b for each location required.

For more information on the properties of these group settings, see [Interaction Recording Web Services Group Settings](#).

Important

If you upload and/or delete recording certificates in one Genesys Administrator Extension session, these changes are not reflected in another Genesys Administrator Extension session. You must log out and log in again to the second Genesys Administrator Extension session.

Deploying the Screen Recording Service

Genesys Interaction Recording (GIR) requires that a Screen Recording Service (SRS) be installed on each Agent's desktop to enable the Agent to capture what is happening on the screen at the time of an active interaction.

The procedures on this page show how to download, install, configure and test the Screen Recording Service.

Important

- For blended agents that are configured to support the handling of both voice and non-voice interactions, GIR will perform screen recording of voice interactions only.
- If the Screen Recording Service is restarted while a recording is in progress or when trying to close a recording, an extra **vlc.exe** process might be left running in the system. If this happens, use Task Manager to stop any remaining **vlc.exe** processes.

Prerequisites

The following list provides you with the requirements you need to successfully deploy the Screen Recording Service (SRS):

- Before you can install and use the SR Service on your desktop, you must have the following information ready at hand. Your IT department or Genesys Professional can help you get this information.
 - Access to Workspace Web Edition (WWE) or Workspace Desktop Edition (WDE)
 - The software (minimum version 8.5.302.10)
- When the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) server is deployed with a GoDaddy, without including the G2-to-G1 cross certificate (that is, the intermediate certificate), you must perform one of the following manual workarounds:
 - Download the G2-to-G1 certificate from https://certs.godaddy.com/repository/gd_bundle-g2-g1.crt.
 - Include the G2-to-G1 cross certificate in the server side certificate (concatenated with the server certificate).
 - Import the G2-to-G1 cross certificate into the SYSTEM (Local Machine) Root CA certificate store.
- Verify that the client machine meets the following minimum specifications:
 - Pentium Dual Core CPU
 - 2 GB RAM (800 MB available for the SR Service)
 - A minimum of 5 GB of available space (in total) for the SR Service installation and working space.

- When the Interaction Recording Web Services server (or Web Services if you are using version 8.5.210.02 or earlier) is deployed with a self-signed certificate, you must import the certificate to **Trusted Root Certification Authorities** for the current user (for example, My User account) and local agent's workstation (for example, Compute Account), from the Certificates Microsoft Management Console (MMC) snap-in .
- If you are running Bria 4 on Windows 7, you must enable Windows Aero. If you do not enable Windows Aero, the Screen Recording Service may fail to capture the Bria 4 application.
- Verify the client machine is synchronized to the same time as the machine on which Media Control Platform (MCP) is installed.
- Starting from version 8.5.500.19, the Microsoft Visual C++ 2015-2022 Redistributable (x64) must be installed on the machine.

Installation considerations

After verifying that your system meets the basic prerequisites, you should consider the following:

- The recommended installation procedure will install the Screen Recording Service's self-signed PFX certificates to the root certificates store. For more information, see [Creating Self-Signed Certificates](#).
- When required use one of the following options to query the Screen Recording Service (SR Service) version:
 - Run the following command line `wmic datafile where name='C:\\<Installation Directory>\\GenesysServiceHandler.exe`.
 - Open the web browser and navigate to <https://127.0.0.1/version> if the SR Service is deployed with HTTPS enabled or <http://127.0.0.1:8080/version> if the SR Service is running as HTTP.
- Proxy support for outbound connections from SRS can be enabled either with or without authentication support.
 - The parameters used to configure the SRS Proxy are available in [Advanced configuration for the Screen Recording Service](#).
- When a proxy is used it may interfere with the SR Service operation. The SR Service runs as an HTTP server and relies on an incoming socket connection to correctly identify the agent's windows session. If the HTTP requests are forwarded by a proxy, the SR Service may not be able to correctly identify the user session in a multi-user environment. With a single user, the SR Service will rollback to the currently active windows session.
When a proxy is used it is recommended that localhost (127.0.0.1) connections be excluded from the proxy settings.
When the proxy is an internal system service (like an Antivirus\Firewall), it is recommended that the SRS related processes (SrsProcess.exe and GenesysServiceHandler.exe) be added to the security software exception\white list.
- The Screen Recording Service can be used by a Citrix client. The following Citrix configurations are supported:
 - Citrix XenApp 7.x or Citrix XenDesktop 7.x running under Windows Server 2019
- In a Citrix environment (for Genesys SR Service 8.5.230.23 and later), SRS only supports a single session per remote PC (Session Sharing is not supported).
- In a Citrix environment, Genesys recommends to configure the `screen-recording.client.max-attempts` parameter to 15 to avoid ping timeout from Workspace Web Edition (WWE) during upgrade,

as upgrade usually takes longer duration in Citrix environment. For more information on the screen-recording.client.max-attempts parameter, see [Integrating with Workspace Web Edition](#).

- In a Citrix environment (for Genesys SR Service 8.5.370.85 and later), the SR Service can be configured to work with Citrix's Virtual Loopback feature.
 - Configure the authenticationHost parameter so that the SR Service uses a loopback IP address that is outside of the range being used by the Citrix Virtual Loopback Feature. See [Advanced Configuration for the Screen Recording Service](#) for more details on how to configure the authenticationHost parameter.
- If SRS is deployed on a Citrix VDA, you need to disable Citrix API hooks for vlc.exe by creating the following registry values. For more information, see [How to Disable Citrix API Hooks on a Per-application Basis](#).
 - **Keys:**
HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook64
 - **Value Name:** ExcludedImageNames
 - **Type:** REG_SZ
 - **Value:** vlc.exe
- If the IPv4 SRS authenticationHost parameter is configured to something other than 127.0.0.1, then use that IP address instead of 127.0.0.1 in the above URLs. See [Advanced Configuration for the Screen Recording Service](#) for more details.
- If the IPv4 SRS authenticationHost parameter is configured to something other than 127.0.0.1, and SRS is configured to use HTTPS, then use that IP address when creating self-signed certificates. See [Creating Self-Signed Certificates to support IP Loopback Addresses other than 127.0.0.1](#) for more details.
- The SR Service can be used in a VMware Horizon environment. The following VMware Horizon configuration is supported:
 - VMware Horizon 7 running under Windows Server 2019
- If you are using [Workspace Web Edition](#) or [Workspace Desktop Edition](#) and the SR Service with Genesys Softphone in a VDI environment (such as Citrix Xenapp), you must configure the screen-recording.client.address option to point to the SRS Loopback address.

Screen Recording Service - operating systems

The Screen Recording Service is supported on the following operating systems in a non-Citrix mode:

- Windows 10 (64-bit)
- Windows 11

The Screen Recording Service is supported on the following operating systems for Citrix support:

- Windows Server 2022
- Windows Server 2019

The Screen Recording Service is supported on the following operating system for VMware Horizon support:

- Windows Server 2022
- Windows Server 2019

Recommended screen resolutions

Genesys has tested the Screen Recording Service under the following recommended screen resolutions. If you use the Screen Recording Service on a computer with a different screen resolution than listed above, you should do a field validation of the Screen Recording Service in your setup to ensure that it is working properly. If you encounter unexpected results, Genesys recommends that you set your screen resolution to one of the recommended and tested resolutions listed below.

Warning

If a field validation has been completed against an earlier version using a non-supported resolution, there is no guarantee that resolution will continue to work on upgrades to new releases. Only supported resolutions are continually tested against each new version.

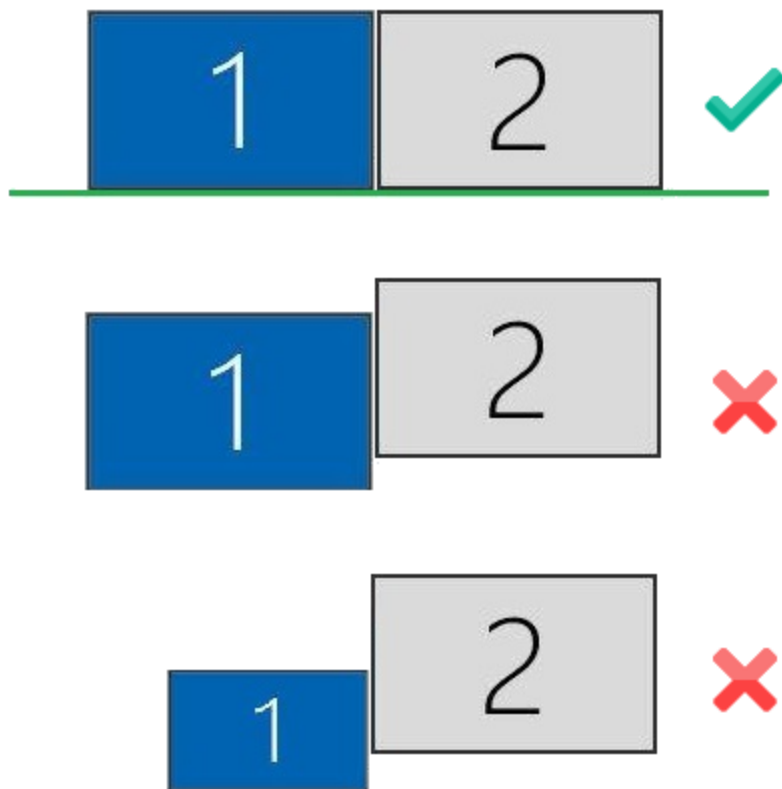
Single Monitor:

- 1024 x 768
- 1280 x 720
- 1600 x 1200
- 1920 x 1080

Dual Monitor:

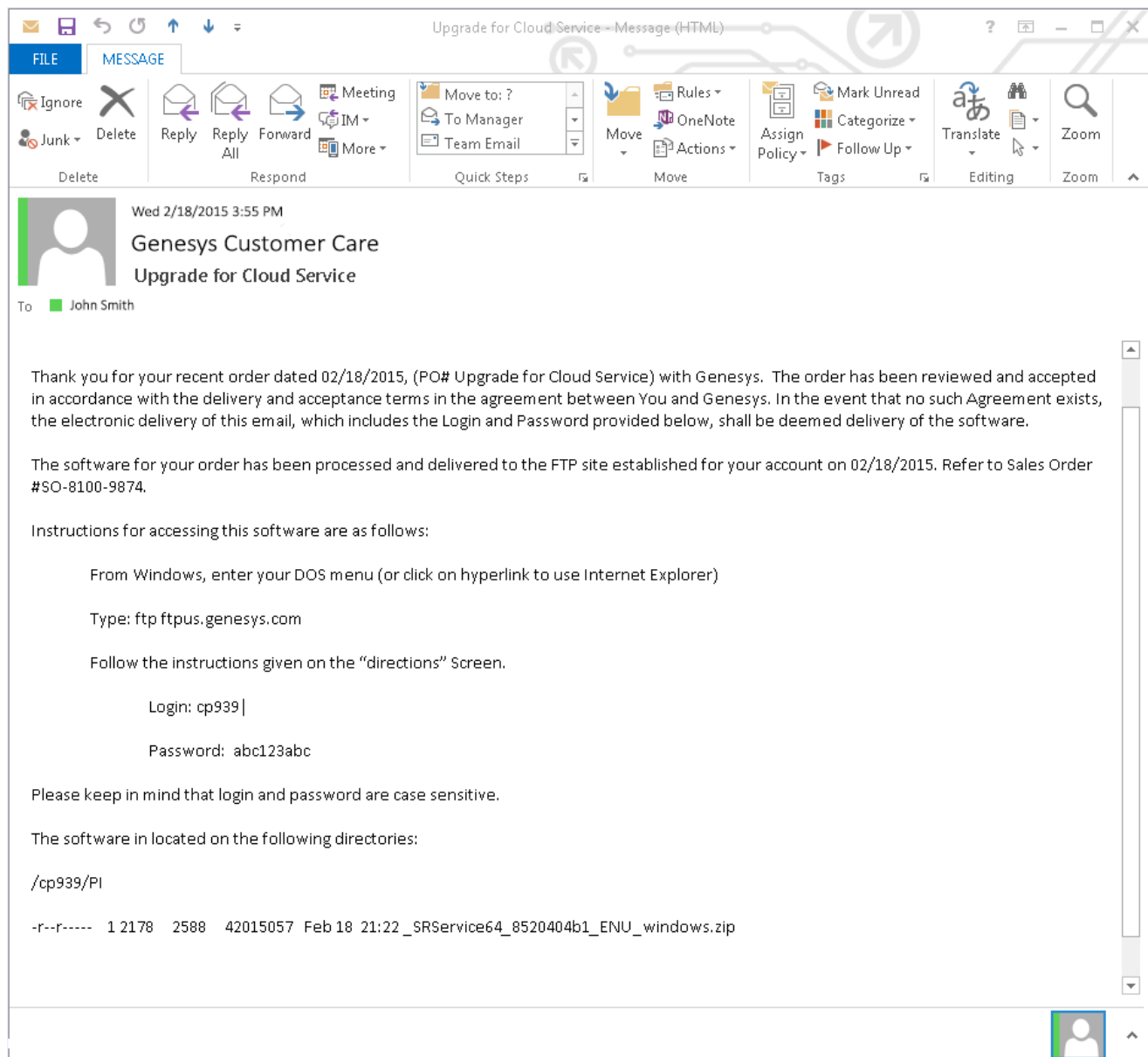
- Side-by-side 1024 x 768 + 1024 x 768
- Side-by-side 1280 x 720 + 1280 x 720
- Side-by-side 1600 x 1200 + 1600 x 1200
- Side-by-side 1920 x 1080 + 1920 x 1080

When using dual monitors, set both displays to the same resolution and arrange them side-by-side (*not* offset) in your display settings, as shown here:



Using dual monitors in a non-recommended configuration can result in errors.

Get your software



Find the email you received from Genesys with the details about your software (it will look similar to the example), and using your favorite FTP client—for example, Filezilla, connect with the credentials listed in the email.

Download the zipped file to a temporary folder on your computer.

Installing your software for the first time

There are two ways to install the SR Service by using:

- **Installation Wizard - for version 8.5.3 and later**
- **Command Prompt**

Important

- To install the Screen Recording Service you must have Administrator privileges.
- Firefox users must close the browser before installing the Screen Recording Service. If Firefox is open while Screen Recording Service is being installed, restart the browser after the installation is completed.

Installing the SR Service for the first time with the installation wizard

This installation procedure is for version 8.5.3 and later.

1. Locate the setup.exe and double-click its icon. The installation wizard is activated.
2. Select one of the following options and click **Next**:
 - **Standard**: Installation will not collect the user's input and proceeds with the default values.
 - **Advanced**: Installation will collect the user's input only for specific configuration parameters.
 - **Customized**: Installation will collect the user's input for all the required parameters.

Important

- For SRS versions 8.5.345.24 and later, selecting the **Standard** mode of installation installs SRS in the HTTPS mode. For SRS versions below 8.5.345.24, this option installs SRS in the HTTP mode.
- The **Use HTTPS self-signed certificates** option is configurable only when the **Advanced** or **Customized** mode of installation is selected. When the **Use HTTPS self-signed certificates** option is selected, SRS uses the HTTPS mode. When this option is not selected, SRS uses the HTTP mode.
- If you select **Use HTTPS self-signed certificates**, you must also specify the following:
 - Base URL for allowed Server Host Names: `https://*.genesyscloud.com`
 - GWS Server URL: `https://<server_name>:443`

3. Select **Use an existing configuration file** (optional) to copy the configuration of one machine, to all other installations of the SR Service on different machines in the same deployment. In the **Location** field, enter the location of the existing configuration file and click **Next**.
4. Select **Use HTTPS self-signed certificates** (SRS uses HTTPS mode). Enter the **Base URL for allowed Server Host Names** and the **GWS Server URL** (see the note, above).
5. For the **Select Certificate Validation** option, select one of the following options: **Do Not Validate the certificate option**, **Validate the certificate using Windows certificate store**, or **Validate the certificate using self-signed certificate**.
6. Verify that the location in the **Destination Folder**, is the correct location (that is, the location where the SR Service will be installed) for the SR Service. If it is not the correct location, enter the correct location and click **Next**.
7. Click **Install**, to complete the first time installation.

Installing the SR Service for the first time with the command prompt

1. Open a command prompt, and type `cd` to change directories to the installation folder.
2. At the prompt, enter the following command and press **Enter**:

```
setup.exe /s /z"-s '<C:\genesys_silent.ini>' -sl '<setup log file name>' -t '<setup wizard log file name>'"
```

For more information, refer to the [Advanced Configuration for the Screen Recording Service](#) section.

Important

- Set the configured `genesys_silent.ini` file path in the command line. Use the absolute path for the input file parameters.
For example, run `setup.exe /s /z"-s 'c:\genesys_silent.ini' -sl 'c:\setup.log' -t 'c:\setup_wizard.log'"`
- The `genesys_silent.ini` file must be configured when using command line silent installation and an unused parameter must be commented out in the `genesys_silent.ini` file. The standard **genesys_silent.ini** file is included with the installation package.
 - The **genesys_silent.ini** file provides all possible configuration parameters along with a description of each.
 - The file lists all the parameters with placeholders.
 - Verify that the unused configuration parameters are either deleted or commented.
 - Verify that the configuration file contains at least the following parameters:
[SRServer]
InstallationType=Standard
[IPCommon]
InstallPath=<Absolute path where the SRS needs to be installed>
[MaintMode]
Mode=FirstInstall

- For SRS versions 8.5.345.24 and later, HTTPS is used by default for the HTTP server within SRS. For the earlier versions of SRS, HTTP is used by default. To change this mode, you must use `InstallationType=Advanced`, `CertificateValidation=UseWinCertStore` and then set `HTTPS=true` or `HTTPS=false`.
- For additional security options, consult a Genesys Professional.
- During the installation process, the antivirus program may block the installation when the installation process detects that the antivirus program is attempting to make system changes. In this scenario, the user will have to unblock the installation program to continue the installation.

Verify the installation

Use Windows Explorer to locate the directory where you installed the software. For example, `C:\Program Files (x86)\GCTI\Genesys SR Service \Logs\GSR`. Once you see the folder is there, restart your computer to confirm that the service starts automatically.

To verify the version installed, browse to `https://127.0.0.1/version` or `http://127.0.0.1:8080/version`.

Enable Screen Recording

In addition to installing and configuring the Screen Recording Service, the agent desktop application must be configured appropriately to support screen recording, and screen recording must be enabled within Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). For further details, refer to the [Enable Screen Recording](#) section.

Test the service and validate the installation

After installation, use Windows Services to confirm that the Genesys SR Service is 'Started'. Check the startup log file as follows:

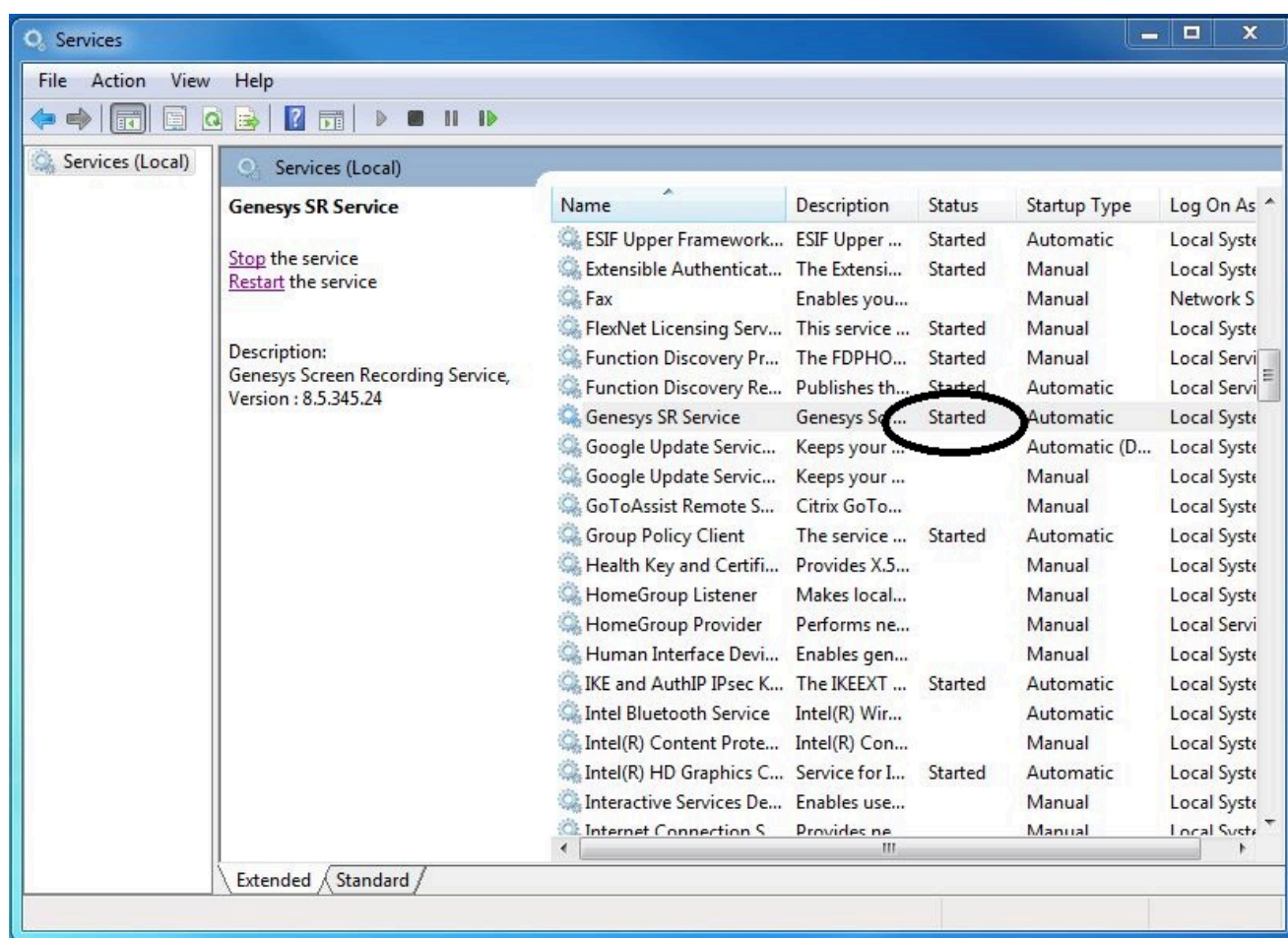
1. Open the `C:\Program Files (x86)\GCTI\Genesys SR Service \Logs\GSR.log` file, and make sure that something similar to the following lines are included (with the version reflecting the version you have just installed):
`ServiceHandler: Running Version:8.5.230.23, IP:135.39.66.17, OS:win32`
2. Make sure that the `C:\Program Files (x86)\GCTI\Genesys SR Service \Logs\GSR.log` file contains no errors

or exceptions.

3. Use the agent desktop to login as an agent that has been configured to have their voice interactions recorded. When the **recordingWhen** parameter is not set to off, the screens will also be recorded when the Screen Recording Service is running. Once logged-in as an agent, request an inbound call to that agent, or use the agent desktop to initiate an outbound call (For example, to a cell phone). Keep the interaction active for 10-20 seconds, and then disconnect the call. Proceed with step 4 to review the log file.
4. After the test, review the C:\Program Files (x86)\GCT\Genesys SR Service \Logs\GSR.log for the following line: Uploader: Upload of file <file-name-of-media> was successful.

Tip

If your installation is unsuccessful, contact your Genesys Professional.



Upgrading the Screen Recording Service

Screen Recording Service can be upgraded manually or automatically. Both types of upgrades assume a functional existing deployment of Screen Recording Service. If the functionality of the existing deployment is in question, it is recommended to look for and stop the service, delete the previous installation folder and proceed as though this is the first time deploying the software. Contact your Genesys Professional if you are not sure if the software is working.

Manual upgrade from any version to 8.5.302.10

1. Create a backup copy of the C:\Genesys\SRC directory and name the backup directory C:\Genesys\SRC.backup.
2. Unzip your new software in a temporary directory (for example, C:\temp).
3. Update the **.ini** file. Access the temporary directory and type the following command in a command prompt window:

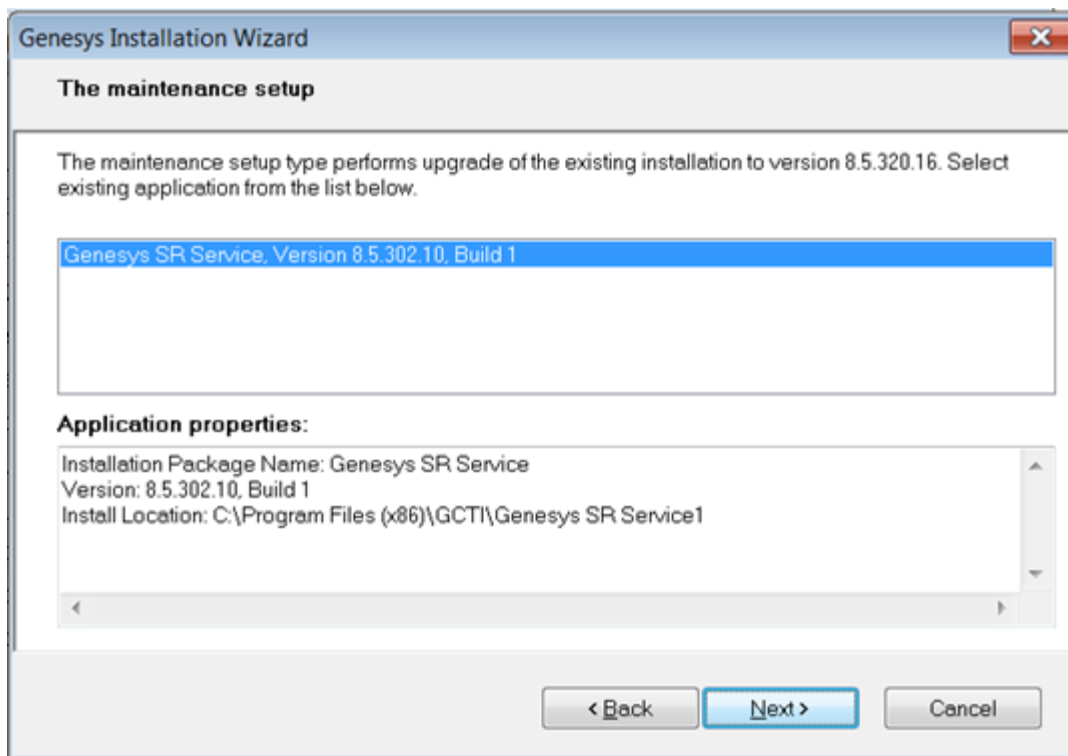
```
setup.exe /s /z"-s '<genesys_silent.ini>' -sl '<setup log file name>' -t '<setup wizard log file name>'"
```
4. Validate the upgrade using the steps in the [Test the Service and Validate the Installation](#) section above.

Manual Upgrade

Important

- The following steps must be performed by a System Administrator.
- Before you upgrade to a newer Screen Recording Service version, check with your Genesys Professional about compatibility with your system.

1. Copy the new SR Service software to a temporary directory.
2. Run the **setup.exe**. As shown in the following image, the setup process automatically detects the existing SR Service installation and selects it for upgradation.



3. Click **Next** and follow the instructions provided in the [Installing the SR Service for the first time with the installation wizard](#) section above.
4. Validate the upgrade using the steps in the [Test the Service and Validate the Installation](#) section.

Automatic Upgrade

Important

When the SR Service Automatic Upgrade attempts to download and install a new version, an Anti-Virus\Firewall may block the SR Service upgrade from downloading and executing the new version files and subsequently prevent automatic upgrades. To prevent this block, it is recommended to add the SR Service processes (SrsProcess.exe, GSRUpdateService.exe and GenesysServiceHandler.exe) to the security software exception / white list.

1. Upload the new SR Service software to the Web Server by copying the content in the IP folder to a location on your Web Server that does not require HTTP authorization. For example, `https://<IP Address>/src/ip/setup.exe`.
2. Configure Genesys Framework to push a new SR Service version to the Agent's Desktop.
 - a. Update the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) Cluster application object in the configuration database. On the **Annex** tab, edit the **[screen-recording-client]** section and add the parameters. **[+] Show the table that describes**

the parameters.

Parameter Name	Description	Required	Example
softwareVersion	The new version of the Screen Recording Service.	Y	8.5.303.02
softwareUrl	The link to the setup file that you copied to the Web Server as part of the IP folder content. If this location is not accessible from the agent desktop that you are intending to upgrade to, the upgrade will not be possible.	Y	https://<IP Address>/src/ip/setup.exe
updateWhen	<p>Determines when to run software update when available. Available options are:</p> <ul style="list-style-type: none"> restart - The safest option in order to not lose any recording. With this option the upgrade will be installed during the next system restart. logout - Runs software update once all agents have logged out. If the agent logs in before the update is complete, they risk losing the screen recording session. In this case, the SR Service will be restarted after the update is complete. immediate - Will shut down SRS and install the new version regardless of the current state (that 	N	restart

Parameter Name	Description	Required	Example
	is, even if a recording is taking place). In this case, SRS will be restarted after upgrade is complete.		

Important

Genesys SR Service on each agent desktop checks for the availability of new software at regular interval, once every 24 hours by default, but can be altered through the configuration parameter **sleepNoNewVersion** on the RWS application object. The new software version is identified and downloaded to the agent desktop when the SoftwareVersion configured on RWS application is greater than the one that is actually installed on the agent's desktop. Later it would start upgrading according to the **updateWhen** parameter.

Upgrading while using HTTPS with an IP address other than 127.0.0.1

When SR Service is upgraded, the self-signed HTTPS certificates are removed and new ones are generated and installed. The newly generated HTTPS certificates will be for the IP address 127.0.0.1. If the IPv4 SRS authenticationHost parameter (see [Advanced Configuration for the Screen Recording Service](#) for more details about the authenticationHost parameter) is configured to something other than 127.0.0.1, then the HTTPS certificates will not work.

To continue using HTTPS with an IP address other than 127.0.0.1, new HTTPS certificates must be generated. Follow the instructions in [Creating Self-Signed Certificates to support Virtual Loopback Addresses](#) to create and install new HTTPS certificates.

Rollback to a previous version

To rollback to a previous version of the Screen Recording Service:

Important

- The SR Service only supports a manual rollback.
- Recordings captured but not uploaded will need to be manually moved to the upload folder of the active SRS directory after the rollback is complete.

1. In the Task Manager, verify that **Genesys SR Service** is stopped. If it has not been stopped, stop it now.
2. Copy the current C:\Program Files (x86)\GCTI\Genesys SR Service directory to a different folder. (For example: C:\Program Files (x86)\GCTI\Genesys SR Service.<date>). This directory contains recordings that have not yet been uploaded; it may be needed for subsequent troubleshooting purposes.
3. Uninstall the existing SR Service installation.
4. Install the previous SR Service version.
5. Restart your computer or start the Genesys SR Service Windows service.
6. Validate the rollback using the steps in the [Verify the Installation](#) section above.

Uninstalling the Screen Recording Service

1. Open the **Start** menu and select **Control Panel**.
2. Click **Programs and Features**.
3. In the **Name** column, select the **Screen Recording Service** entry (for example, Genesys SR Service 8.5.xxx.yy), right click and select **Uninstall**.

The Screen Recording Service is uninstalled.

Deploying the Screen Recording Service - Advanced Configuration

The following sections provide advanced Screen Recording Service installation and configuration steps.

For basic instructions about how to install and configure the Screen Recording Service, see: [Deploying the Screen Recording Service](#).

Enable Screen Recording

Important

Before you can start to capture and play back the screen recordings, you must make sure that you have configured the [Interaction Recording Web Services components](#) (or [Web Services components](#) if you're using version 8.5.210.02 or earlier), and [encryption](#) specifically for screen recording.

To set up recording conditions, using Genesys Administrator Extension, add the **recordingWhen** parameter to the **[screen-recording-client]** section of the Interaction Recording Web Services or Web Services Cluster application object, depending on your deployment (see [Installing RWS](#)).

When this parameter is set in the Cluster object, the recording condition applies to all agents in the environment. You can create the **recordingWhen** parameter in a **[screen-recording-client]** section of each agent object to override the settings at the environment level.

The parameter value is an expression of conditions to enable screen recording for each agent. The format is:

- For Non-voice agents: **recordingWhen** = condition1,condition2,... where condition1, condition2, etc. are a set of conditions that must all be true in order for the screen recording to be taking place.
- For Voice agents: Screen recording starts when the voice recording starts except in cases where **recordingWhen** is explicitly set to off.

Important

For blended agents that are configured to support the handling of both voice and non-voice interactions, GIR will perform screen recording of voice interactions only.

Integrating with Workspace Web Edition

If your agents use Workspace Web Edition (WWE) as their desktop, screen recording must be set up as follows:

Important

- The SR Service does not support single sign-on for WWE.
- If the following Internet Explorer 11 settings are enabled when the SR Service is used together with WWE, you must work with SR Service version 8.5.302.14 or later:
 - Enhanced Protected Mode under the Miscellaneous settings
 - Enable Protected Mode under Security Setting
 - Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) address is added to the Local Intranet sites

Using Genesys Administrator Extension, add the following parameters to the interaction-workspace section of the Web Services Cluster application object.

Important

If you are working with HTTP, the Screen Recording Service default port number is 8080. If you are working with HTTPS, the default port number is 443. In addition, verify that the Workspace Web Edition configuration is set to 8080 or 443.

Parameter Name	Mandatory	Description	Default Value
privilege.screen-recording.can-use	Y	Specifies whether agents can use screen recording. If set to true, the integration module is loaded and sends credentials to the client.	false
screen-recording.client.address	N	Specifies the IP address that the Screen Recording Service listens for credentials on. Valid values: 127.0.0.1, [::1]	127.0.0.1
screen-recording.client.port	N	Specifies the port that the Screen Recording Service listens for credentials on.	443

Parameter Name	Mandatory	Description	Default Value
screen-recording.client.ping-interval	N	Specifies the interval, in milliseconds, between ping requests to the Screen Recording Service.	5
screen-recording.client.max-attempts	N	Specifies the maximum number of attempts to establish communication with the Screen Recording Service. Note: In a Citrix environment, set the value of this parameter to 15.	5
screen-recording.client.secure-connection	N	Specifies if a secure connection will be made to the Screen Recording Service using HTTPS.	true
screen-recording.client.server-url	N	Defines the Interaction Recording Web Services (Web Services) server address that the Screen Recording Service will use for communication.	

Integrating with Workspace Desktop Edition

If your agents use Workspace Desktop Edition as their desktop, screen recording must be set up according to the instructions in the [Workspace Desktop Edition Deployment Guide](#).

Important

The SR Service does not support single sign-on for Workspace Desktop Edition.

Enable Screen Recording for a Contact Center

To enable the screen recording feature for a given Contact Center refer to the [Configuration for Screen Recordings > Configuring the Interaction Recording Web Services Parameters](#) section.

Advanced Installation Procedures

Creating Self-Signed Certificates

During installation the SR Service can create self-signed certificates to be used as local host

connections. To do this, select the **Use HTTPS self-signed certificates** check box in the advance installation. For the SR Service version 8.5.345.24 and later, selecting the **Standard** option installs SR Service in the HTTPS mode and creates a self-signed certificate.

To create self-signed certificates as local host connections, following installation, perform the following:

1. Open a command window as an Administrator.
2. Navigate to the `<install_dir>\Certificates\Self-Signed` directory.
3. Run the **create_certificates.bat** file. This creates a set of unique self-signed certificates.
4. Run the **install_certificates.bat** file. This installs the new self-signed certificates to Windows trusted certificates store.

Important

- If the SR Service is installed with self-signed certificates for the local host server, the certificates are automatically imported into the Firefox certificate database. If Firefox is installed after the SR Service is installed, the certificates must be imported manually. To import the self-signed certificates into the Firefox database, run the following script as an administrator `<install_dir>\Certificates\Firefox\add_certificates.bat`.
- When the SR Service starts, it will attempt to read the certificate files `server.pem` and `serverIp6.pem` in the `<install_dir>\Certificates` directory. If these files are missing, the SR Service will run in HTTP mode instead of HTTPS mode.

Installing Your Own Certificates

If desired, you can use your own certificates as follows:

1. Provide a certificate for the IPv4 host, 127.0.0.1, in the `<install_dir>\Certificates\server.pem` file.
2. Install the .pfx form of this certificate to the local certificates store as a "Trusted Root Certification Authority" file.
3. Provide a certificate for the IPv6 host, `:::1`, in the `<install_dir>\Certificates\serverIp6.pem` file.
4. Install the .pfx form of this certificate to the local certificates store as a "Trusted Root Certification Authority" file.
5. The PEM certificate files should include both the private RSA key and the certificate itself. **[+] Show an example.**

```
-----BEGIN RSA PRIVATE KEY-----
.
.
.
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
.
.
.
-----END CERTIFICATE-----
```

Important

When the .pem certificates must be protected by a password, the password is configured in the `config.json` file using the `certificatePassword` parameter. The default `certificatePassword` is `genesyscreenrecording`. For more details, refer to the [Client Side parameters](#) table in step #2 of the Advanced Configuration for the Screen Recording Service section.

Creating Self-Signed Certificates to support IP Loopback Addresses other than 127.0.0.1

SRS can be configured so that its Authentication Server uses Loopback IP Addresses other than 127.0.0.1. The HTTPS Certificates that are created by default only work if SRS is configured to use the Loopback IP Address 127.0.0.1. To use SRS with Loopback Addresses besides 127.0.0.1 and HTTPS, new HTTPS Certificates must be created specifically for the Loopback IP Address that SRS is using.

To create self-signed certificates with Loopback addresses other than 127.0.0.1, following installation, perform the following:

1. Open a command window as an Administrator.
2. Navigate to the `<install_dir>\Certificates\Self-Signed` directory.
3. Run **`uninstall_certificates.bat`** to remove the existing certificates.
4. Run **`create_certificates.bat`** and pass a value for the **`IPV4_HOST`** parameter. Below is an example to create certificates for 127.1.1.2:

```
create_certificates.bat -IPV4_HOST 127.1.1.2
```
5. Run **`install_certificates.bat`** to install the new certificates. This installs the new self-signed certificates to the Windows trusted certificates store.
6. Configure SRS to use the newly created certificates. Please see the **`authenticationCertificate`** option in [Advanced Configuration for the Screen Recording Service](#) for more details.
7. Restart the Genesys SR Service Windows service.

Advanced Configuration for the Screen Recording Service

Some Screen Recording Service configurations are managed locally on the system (that is, using the `config.json` configuration file). Other configurations are managed centrally. Advanced configuration should be performed using the Interaction Recording Web Services or Web Services Cluster application object, depending on your deployment (see [Install RWS](#)) in Genesys Administrator Extension. All the configuration parameter values should be in JSON notation. More information about how JSON escapes rules can be found here: <https://msdn.microsoft.com/en-us/library/dn921889.aspx>.

Important

Screen Recording Service does not support the use of System Proxies configured via PAC (Proxy Auto-Configuration) files.

Important

The default port number for SRS is 443. If this port is used by another application, you must configure the **authenticationPort** and **authenticationPortIp6** parameters to use a different port. The following parameter for Agent Desktop must also be changed accordingly:

```
[interaction-workspace] screen-recording.client.port
```

1. If your server uses a self-signed certificate, set the **certificate** parameter to the path on the file system where the pem file is stored.
2. Edit the local **config.json** file on the Screen Recording Service machine, and add the client parameters.
Note: The following parameters should ONLY be configured locally and NOT using GAX. Please note that in a multiple user SR Service deployment these settings will take effect for all users using the system.

Important

All parameter names are case sensitive.

Name	Mandatory	Description	Default value
addressType	N	<p>Enables the identification of the SR Service for monitoring and reporting purposes on Interaction Recording Web Services (Web Services). addressType supports the following options:</p> <ul style="list-style-type: none">• fqdn - Use fully qualified domain name.• ip - Use IPv4.• ip6 - Use IPv6. <p>Note: With addressType you</p>	fqdn

Name	Mandatory	Description	Default value
		can also provide a custom name to identify a specific machine (for example, pc-id-1).	
allowedHosts	N	<p>Represents a list of allowed host names to be configured as the SRS Interaction Recording Web Services (Web Services) server address using POST API. The value can be a single address, a list of specific addresses or a wild card.</p> <p>Notes:</p> <ul style="list-style-type: none"> When SRS receives the server configuration parameter from Workspace Web Edition (WWE) after the agent logs in, it will check if the URI matched the allowedHosts configuration parameter. If there is a match, it will establish a connection to the Genesys Web Services with the information provided regardless of whether or not the Server parameter is configured locally. If the parameter does not match, the SR Service will only use the Server parameter if it is configured. If the Server parameter is not configured, the SR Service will not establish a 	https://*.genesyscloud.com

Name	Mandatory	Description	Default value
		<p>connection with Interaction Recording Web Services (Web Services).</p> <ul style="list-style-type: none"> If the server parameter is not configured in the config.json file and it is passed in real time as part of the login POST API, you must update the <code>allowedHosts</code> so that it matches the server address of Interaction Recording Web Services (Web Services) in the actual deployment. You can configure multiple host URLs for <code>allowedHosts</code> in the manner in which JSON presents multiple values (<code>["URL1", "URL2", . . . , "URLN"]</code>). For example, <code>{"name": "allowed Hosts", "value": ["URL1", "URL2"]}</code>. 	
allowedOrigins	N	<p>Specifies the approved CORS Origin headers that Screen Recording Service approves. If it is not provided, the <code>*</code> character is set as the default, which means any request will be approved, with or without origin header.</p> <p>If it is provided, the value can be a single origin, or a list of approved origins, that is used to approve the CORS requests. The defined server parameter is always added to the list of approved</p>	*

Name	Mandatory	Description	Default value
		origins automatically.	
authenticationCertificate	N	<p>The relative or full path to the authentication server's PEM certificate. If a value is available, the authentication server uses it for the HTTPS connection to the agent's desktop.</p> <p>Note: This parameter is not needed in the default Screen Recording Service installation. The Screen Recording Service uses the default self-signed certificate (%install_dir\Certificates\server.pem) automatically.</p>	'%install_dir\Certificates\server.pem'
authenticationCertificateIp6	N	<p>The relative or full path to the authentication server's PEM certificate for IPv6. If a value is available, the authentication server uses it for the HTTPS connection to the agent's desktop.</p> <p>Notes:</p> <ul style="list-style-type: none"> This parameter is not needed in the default Screen Recording Service installation. The Screen Recording Service uses the default self-signed certificate (%install_dir\Certificates\server.pem) automatically. Each host requires a unique certificate. 	%install_dir\Certificates\serverIp6.pem
authenticationHost	N	The IPv4 Address that the Authentication Server will bind to when SRS starts if SRS is configured to use IPv4. The parameter	127.0.0.1

Name	Mandatory	Description	Default value
		value must be an IPv4 address within 127.0.0.0/8. The IP addresses 127.0.0.0 and 127.255.255.255 are not allowed.	
authenticationPort	N	The port used for internal communication with Web Services.	If using HTTP, the port is 8080. If using HTTPS, the port is 443.
authenticationPortIp6	N	The port used for internal communication with Web Services.	If using HTTP, the port is 8080. If using HTTPS, the port is 443.
certificate	N	Indicates how the Screen Recording Service validates the Web Services server TLS certificate. If set to false, the Screen Recording Service will not validate the certificate. If set to true, the client will validate the certificate using the Windows certificate store when the server is using a certificate from the public CA. If set to a file path (for example, C:\Automation\server.pem), the Screen Recording Service will validate the certificate using a self-signed certificate when the server is using a private self-signed certificate.	true
certificatePassword	N	The password for the PEM certificate's private RSA key.	Empty
certificatePasswordIp6	N	The password for the IPv6 PEM certificate's private RSA key.	Empty
credentialsTimeout	N	The timeout duration, in minutes, between the keep alive (GET/Ping) requests from the agent's desktop and the Screen Recording Service. When the	35 Note: This value must be longer than the Web Services session timeout duration for the agent's desktop. By default the Web

Name	Mandatory	Description	Default value
		timeout expires, the agent's credentials are deleted from the Screen Recording Service's cache.	Services session timeout is 30 minutes, and the credentialsTimeout is 35 minutes. The latter must be increased if the Web Services session timeout is increased.
diskCheckInterval	N	The interval, in seconds, between disk space checks.	30
diskFreeSpaceLimit	N	The minimum disk space, in MB, on the client machine. When the disk space drops below this value, the screen recording will stop any active recording sessions.	2000
diskFreeSpaceThreshold	N	The amount of free space available above the defined limit, before recordings can be restored, after dropping below the disk space limit.	500
ip6	N	Indicates whether to support IPv6 in addition to IPv4 for communication with Web Services.	true
isVlcSlowCapture	N	Indicates that VLC has delay in starting the screen recording. If set to true, the Screen Recording Service (SRS) will update the start time of the screen recording with the time the media file is created. The valid values are true and false. Warning: This parameter is deprecated by preLoadVlc .	false
partitionedCookies	N	Enables partitioned cookie to support new changes in the Google Chrome browser related to sharing of third-party cookies. The configuration parameter,	2

Name	Mandatory	Description	Default value
		<p>partitionedCookies, supports the following options:</p> <ul style="list-style-type: none"> • 0 - disabled (never add the Partitioned cookie attribute) • 1 - enabled (always add the Partitioned cookie attribute) • 2 - auto (enable the Partitioned cookie attribute conditionally when Chromium version requires it; this is the default) • If Chrome/Edge version is 118 or higher, partitionedCookies will be enabled. • If Chrome/Edge version is lower than 118, partitionedCookies will be disabled. • For Firefox, partitionedCookies will be enabled for all versions. 	
peer_server	N	<p>The server base url of the backup data center. The default port is 80; to use a different port, use the url:port format.</p> <p>This value will be overridden if supplied by the client application.</p> <p>Note: This parameter is not applicable for single data center deployments.</p>	Empty
postProcessingSavePath	N	<p>The post processing temp location. When used as a UNC path, verify that the computer running SRS</p>	%LOCALAPPDATA%/Genesys/SRS (C:/Users/<user_name>/AppData/Local/Genesys/SRS)

Name	Mandatory	Description	Default value
		(SYSTEM account) has read\write permissions.	
preLoadVlc	N	Decides whether to load VLC process in advance after the agent logs in. Valid values are true and false. Warning: Only configure this parameter if instructed by Genesys.	false
proxyServerHost	N	The proxy server hostname or IP address.	Empty
proxyServerPort	N	The proxy server port.	Empty
proxyServerUsername	N	The username to connect to the proxy server.	Empty
proxyServerPassword	N	The password to connect to the proxy server.	Empty
reEncodingTimeoutSeconds	N	Specifies the number of seconds that Screen Recording Service will wait for VLC to finish processing a screen recording after a call that includes pause and resume operations. Valid values are any integer greater than 0. Warning: Only configure this parameter if instructed by Genesys.	120
rwsRetryBeforeSwitchOver	N	The number of times SRS will attempt to connect to the primary RWS before switching over to RWS in the backup data center and vice versa.	1
sendLogToGWS	N	Disables the sending of an error log to Interaction Recording Web Services (Web Services) from the SR Service.	false
server	N	The server base url.	Empty

Name	Mandatory	Description	Default value
		The default port is 80; to use a different port, use the url:port format.	
sharedSavePath	N	The Shared folder. The location in which recordings are saved to be uploaded. When used as a UNC path, verify that the computer running SRS (SYSTEM account) has read\write permissions.	<Installation_dir>
statusTimeout	N	The timeout duration, in seconds, between the keep alive GET/ Ping requests from the agent's desktop and the Screen Recording Service.	60
systemMetricTimeout	N	The timeout duration, in seconds, for reading the system metrics. On a slow machine, set a higher timeout value to avoid timing out from reading the system metrics.	5
userSavePath	N	The user recordings temp location. The location must be a local folder. If a non-default location is used, verify that the user has read\write permissions.	%LOCALAPPDATA%/Genesys/SRS (C:/Users/<user_name>/AppData/Local/Genesys/SRS)
useSystemProxy	N	If this value is true, the Screen Recording Service uses the Windows System Proxy settings.	false
vlcHttpTimeout	N	The HTTP request timeout, in seconds, for VLC start and stop recording commands.	10
vlcPortBegin	N	The beginning of the port range for the VLC http interface.	4916
vlcPortEnd	N	The end of the port range for the VLC http interface.	65530

Important

Proxy server parameters specified in the config.json file take precedence over the **useSystemProxy** parameter.

3. In the most basic configuration, you will not need to add the following parameters, they are all optional. However, if you intend to use any of the server parameters, use Genesys Administrator Extension, and follow the next steps:
 - a. At the Environment level, locate the Interaction Recording Web Services or Web Services Cluster application object, depending on your deployment (see [Install RWS](#)).
 - b. Edit the application object, and create a new section named **screen-recording-client**. The following table provides an example of the **screen-recording-client** section.

Important

All parameter names are case sensitive.

Name	Mandatory	Description	Default value
CaPath	N	The path for the authority PEM certificate file used for verification of encryption certificates. If not present, verification will not take place.	false
cleanupPolicy	N	<p>Specifies the method for managing failed screen recording files on the Client machine. The available values are:</p> <ul style="list-style-type: none">• delete – Deletes the recording from the local drive, regardless if the upload was successful or not.• keep – Deletes successfully uploaded recordings. Recordings whose upload failed are	keep

Name	Mandatory	Description	Default value
		<p>kept in the Recordings folder and retried until they are successfully uploaded.</p> <ul style="list-style-type: none"> keepForever - All recordings are permanently stored on the local drive. Successfully uploaded recordings are stored in the Uploaded sub-folder. Recordings whose upload failed are kept in the Recordings folder and retried until they are successfully uploaded. <p>Note: This setting is only recommended for debugging, as it can cause disk space to run out quickly.</p>	
clockColor	N	The color of the time stamp clock. Use HTML color codes.	0xffffffff (white)
clockFormat	N	The display format for the time stamp clock. See the table later in this section for the valid values.	%H:%M:%S-%Y-%m-%d %Z (HH:MM:SS-YYYY-MM-DD TZ)
clockOpacity	N	How non-transparent the time stamp clock displays. Valid values: 0-255	150
clockPosition	N	The position for time stamp clock. Valid values: 0=center, 1=left, 2=right, 4=top, 8=bottom. You can also use combinations of these values—for example, 6 = top-right.	8 (bottom-center)

Name	Mandatory	Description	Default value
clockSize	N	The font size for the timestamps written to the screen. Note: This option is available if the <code>timeStamp</code> option is set to <code>true</code> .	40
delayShutdown	N	The time, in seconds, to delay shutting down the SRS and the system if the uploadPolicy parameter is set to <code>immediate</code> . This allows all the screen recording files to upload before the shutdown starts. The maximum value is 125 seconds (limited by Windows).	15
encodingLevel	N	The H.264 encoding level restriction. Valid values: 10,11,12,13,20,21,22,30,31,32,40,41,42,50,51. For more information, see H.264/MPEG-4 AVC Levels .	
encodingProfile	N	The H264 encoding profile. Valid values: baseline, main, high.	high
folder	N	The folder name where the media is uploaded in the WebDAV server.	/
fps	N	Frames per second.	1
grayScale	N	Indicates whether to record the screen in color or gray scale. Set to <code>true</code> to record in gray scale. Set to <code>false</code> to record in color.	false
ignoreCertificateVerificationErrors	N	Ignores errors that occur during certificate verification for screen recording encryption. This option is used only when certificate verification is enabled by configuring the CaPath parameter. Valid values:	true

Name	Mandatory	Description	Default value
		<ul style="list-style-type: none"> • true: The errors that occur during certificate verification will be ignored with a warning message being logged. • false: The errors that occur during certificate verification will not be ignored. 	
isACWEnabled	N	<p>Indicates whether to record the agent when they are in the After Call Work (ACW) state.</p> <p>Note: You must also configure the wrap-up-time parameter under the T-Server or Agent Login object. (The Agent Login object is not supported for deployments using SIP Cluster.) For more information, see Agent Login on the Deploying SIP Server for GIR page.</p>	<p>true</p> <p>Note: If isACWEnabled is set to any value other than false, then the value is true.</p>
keepAspectRatio	N	Indicates whether to keep the original aspect ratio or stretch the video to fill the screen if the screen resolution is large than the maximum resolution, and the screen is down scaled.	true
logsToKeep	N	The number of log files to keep.	10
logLevel	N	The logging level. Set to one of the following: debug, info, warning, error, critical. Only messages with a level set equal to or above the defined level will be logged.	info
maxDurationMinutes	N	The maximum duration, in minutes, before slicing a screen recording file.	<p>According to the selected qualityPreset:</p> <p>low-180 standard-120 high-75</p>

Name	Mandatory	Description	Default value
maxHeight	N	The maximum height resolution in pixels. The client will always use the lower resolution defined by either the maxHeight/maxWidth parameters or the resolutionScale parameter.	1080
maxLogSize	N	The maximum size, in MB, of the log file before a new log file is created. The old log file is named with the .1 extension. Set the value to 0 if you do not want to limit the log file size.	5
maxWidth	N	The maximum width resolution in pixels. The client will always use the lower resolution defined by either the maxHeight/maxWidth parameters or the resolutionScale parameter.	1920
multipleMonitorsEnabled	N	Indicates whether to record on all available monitors. If set to <code>false</code> , the client will record on the primary display monitor only.	true
qualityPreset	N	Defines the desired bitrate, depending on the agent's screen resolution. Valid Values: <ul style="list-style-type: none"> low—Emphasis is on storage capacity, and text is readable 90% of the time. For example, 120 kbit/s for 1920 x 1080 resolution with color. standard—Text should be readable 100% of the time 	standard

Name	Mandatory	Description	Default value
		<p>with normal use. For example, 150 kbit/s for 1920 x 1080 resolution with color.</p> <ul style="list-style-type: none"> high—Emphasis is on quality, and text should be readable 100% even on a high movement environment. For example, 190 kbit/s for 1920 x 1080 resolution with color. <p>See the table later in this section for the full list of preset examples.</p>	
recordingWhen	N	<p>An expression from configuration states when screen recording should be taking place for a particular recording client. The format is:</p> <p>recordingWhen= <code>condition1,condition2,...</code> where <code>condition1,condition2,...</code> are a set of conditions that must all be true in order for the screen recording to take place. In other words: Screen Recording Active = condition1 && condition2 && ...</p> <p>Note: If the state of any of the conditions is unknown (occurs only before first determining agent state, so limited to initial state), then the state of screen recording is unknown. See the table later in this section for the full list of conditions.</p>	random_voice(100)
resolutionScale	N	Used to scale the screen size. Setting resolutionScale to 0.5 will resize the screen resolution in half. Setting it to 1 will do nothing. The client	1

Name	Mandatory	Description	Default value
		will always use the lower resolution defined by either the maxHeigh/maxWidth parameters or the resolutionScale parameter.	
rwsFailedRecordingRetrySleep	N	The time, in minutes, to sleep before retrying recordings that failed to upload.	15
rwsRetryBeforeSwitchOver	N	The number of times SRS will attempt to connect to the primary RWS before switching over to RWS in the backup data center and vice versa.	1
sleepNoConnection	N	The maximum time, in minutes, that the client will sleep if there is no connection with the server before attempting to reconnect.	1
sleepNoNewVersion	N	The time, in minutes, that the client updater thread will sleep if a new version is not available.	1440(24H)
slowMachine	N	Indicates whether the Screen Recording Service is installed on a slow machine, so that the extra time is available to save the video files before closing the client. Note: slowMachine has been deprecated by vlcCloseTimeout as of 8.5.302.14.	false
softwareChecksum	N	The SHA512 checksum of the latest software setup file.	Empty
softwareUrl	N	The URI used to fetch the latest Screen Recording Service software installation package.	None
softwareVersion	N	The latest Screen	Empty

Name	Mandatory	Description	Default value
		Recording Service software version number.	
systemMetricTimeout	N	The timeout duration, in seconds, for reading the system metrics. On a slow machine, set a higher timeout value to avoid timing out from reading the system metrics.	5
timeout	N	The timeout duration, in seconds, for HTTP requests. This value must be bigger than the cometD Server request timeout.	60
timeStamp	N	Indicates whether a time stamp will be water marked on the video.	false
updateWhen	N	<p>Determines when to run software update when available. Available options are:</p> <ul style="list-style-type: none"> restart - The safest option in order to not to loose any recording. With this option the upgrade will be installed during the next system restart. logout - Runs software update once all agents have logged out. If the agent logs in before the update is complete, they risk losing the screen recording session. In this case, the SR Service (SRS) will be restarted after the update is complete. immediate - Will shut down SRS and install the new 	restart

Name	Mandatory	Description	Default value
		<p>version regardless of the current state (that is, even if a recording is taking place). In this case, SRS will be restarted after upgrade is complete.</p> <p>Note:</p> <ul style="list-style-type: none">• To receive a new version of the SR Service, you must first log into Workspace Web Edition (WWE). Use the immediate option with caution. Since the SR Service is restarted immediately, this may cause screen recordings to be lost and may require the agent to logout and login again to restore the screen recording operation. If possible always use the logout or restart option.• If updateWhen is set to restart and the system is restarted before all the SRS installation files are downloaded, the download process continues after the system is restarted. However, the software is updated only during the next system restart.	
uploadPolicy	N	Specifies the screen recording upload policy. If set to window, the screen recording files are uploaded to	immediate

Name	Mandatory	Description	Default value
		<p>storage during the times specified by the windowStartTime and windowEndTime parameters. If set to immediate, the files are uploaded immediately; however, after the agent's last call, the Screen Recording Service needs some time to upload the recording to the server before the Agent's desktop shuts down. The amount of time needed depends on the duration of the last call and network speed. Genesys recommends to estimate one minute for every minute of screen recording on a network with 150 kbit/s per second and upload speed approximate to 20 KB per second. For example, if the last screen recording lasted 10 minutes, and the network speed is 300 kbit/s (~40KB/s), five minutes is required.</p> <p>Note: If the Agent's PC is shutdown before the upload is completed, the recording will be uploaded on next PC start up.</p>	
videoBitrate	N	Encoding bitrate. Use this parameter to override the default bitrate that is calculated based on the resolution and the selected qualityPreset value.	150 kbit/s for 1920 x 1080 resolution (standard preset, color recording)
vlcCloseTimeout	N	Sets the amount of time the SR Service will wait, after stopping a screen recording, before closing VLC. This time is required to	2

Name	Mandatory	Description	Default value
		ensure VLC completes writing the file correctly. This time should not be changed unless the SR Service is running on a very slow machine, and the screen recording file is invalid but without an error in the log. If such a scenario occurs, increase the time the SR Service must wait before closing VLC.	
windowEndTime	N	Specifies the upload end time, in the local time. The format is hh:mm—for example, 23:00. This parameter is mandatory for the Window upload policy.	Empty
windowStartTime	N	Specifies the upload start time, in the local time. The format is hh:mm—for example, 23:00. This parameter is mandatory for the Window upload policy.	Empty

Video File Size/Compression Optimization Estimate

The following table provides file size estimates according to the selected **Quality Preset**, FPS and color scheme, given a specific resolution.

Preset	Color	Resolution	Frame Rate	Encoding Level	Average File Size MB/Minute
Low	Color	1920x1080	1	High 4.1	0.864
Standard	Color	1920x1080	1	High 4.1	1.055
High	Color	1920x1080	1	High 4.1	1.37
Low	Grayscale	1920x1080	1	High 4.1	0.608
Standard	Grayscale	1920x1080	1	High 4.1	0.732
High	Grayscale	1920x1080	1	High 4.1	0.886

Recording Conditions

The following table describes the recording conditions for the **recordingWhen** parameter:

Condition	Description
off	A special case. Cannot appear with other conditions. When specified as such, screen recording never occurs for the agent.
loggedin	When the agent is logged in
DNDoff	When agent sets DND (do not disturb) to off
ready(any)	True when any media type is set to ready, or <code>list(ready media).count != 0</code>
ready(abc)	True when the abc media type is set to ready
ready(abc,...xyz)	A list of media types that are set to ready. Note that <code>ready(abc,...xyz) = ready(abc) ... ready(xyz)</code> .
random_voice(%)	Records the agent's screens based on a percentage of the total voice call volume for that agent.

Important

Each individual setting's key/value can be overwritten at the agent level by setting the Person object with the Annex of the same section name (**screen-recording-client**).

Clock Format Directives

The follow table lists and describes the values that are available for the **clockFormat** parameter.

Directive	Meaning
%a	Locale's abbreviated weekday name.
%A	Locale's full weekday name.
%b	Locale's abbreviated month name.
%B	Locale's full month name.
%c	Locale's appropriate date and time representation.
%d	Day of the month as a decimal number [01,31].
%H	Hour (24-hour clock) as a decimal number [00,23].
%I	Hour (12-hour clock) as a decimal number [01,12].
%j	Day of the year as a decimal number [001,366].
%m	Month as a decimal number [01,12].
%M	Minute as a decimal number [00,59].
%p	Locale's equivalent of either AM or PM.

Directive	Meaning
%S	Second as a decimal number [00,61].
%U	Week number of the year (Sunday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Sunday are considered to be in week 0.
%w	Weekday as a decimal number [0(Sunday),6].
%W	Week number of the year (Monday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Monday are considered to be in week 0.
%x	Locale's appropriate date representation.
%X	Locale's appropriate time representation.
%y	Year without century as a decimal number [00,99].
%Y	Year with century as a decimal number.
%Z	Time zone name (no characters if no time zone exists).
%%	A literal '%' character.

Deploying Recording Muxer Script

Genesys Interaction Recording (GIR) needs the Recording Muxer Script to combine the call and screen recordings for a seamless playback.

Recording Muxer Script (Python 3)

Prerequisites

Before installing and configuring the Recording Muxer Script, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage or S3 Premise where the recordings are stored.
- For Recording Muxer Script 8.5.500.10 (or higher), Recording Processor Script must be upgraded to 8.5.500.13 (or higher) if using Recording Processor Script.

Installing Recording Muxer Script

Installing on Windows

1. Install 64-bit Python 3.11.5 from the [Python](#) website. To make Python 3 to work with OpenSSL 3.0.13, follow the below steps:
 - Download `libcrypto-3.dll` and `libssl-3.dll` from the [Python Binary repository](#).
 - In `[python-source-folder]\DLLs`, replace with the above downloaded DLL files.
2. Install **Recording Muxer Script IP** with the installer.

Note: Install the following third-party libraries in the order they appear and untar the files in Administrator mode.

3. Untar the `<Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz` file.
4. Run `py -m pip install .` from the `<Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1` directory.
5. Untar the `<Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz` file.

-
6. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
 7. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
 8. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
 9. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
 10. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
 11. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
 12. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
 13. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
 14. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
 15. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
 16. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
 17. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
 18. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
 19. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.
 20. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0 directory.
 21. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
 22. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
 23. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
 24. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
 25. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
 26. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
 27. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
 28. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
 29. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
 30. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
 31. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
-

32. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
33. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
34. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
35. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
36. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
37. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
38. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
39. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
40. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
41. Unzip the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-win64-static-gpl3.0.zip.
42. Unzip the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-3.0.13-win64.zip. This OpenSSL library is used to encrypt the resulting muxed recording file when required.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).
2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `libffi-devel` (`yum install libffi-devel`).
4. Install OpenSSL.
 - For 8.5.500.09 or lower versions, install OpenSSL version 1.1.1.
 - For 8.5.500.10 or higher versions, install OpenSSL 3.0.13. Download OpenSSL 3.0.13 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-3.0.13 --openssldir=/usr/home/openssl-3.0.13 --libdir=lib no-shared`
5. Install 64 bit Python 3.11.5.
 - For 8.5.500.09 or lower versions, compile with OpenSSL 1.1.1 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
 - For 8.5.500.10 or higher versions, compile with OpenSSL 3.0.13 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-3.0.13 --enable-optimizations`
6. Install the **Recording Muxer Script IP** with the installer provided.

Note: Install the following third-party libraries in the order they appear.

-
7. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz file.
 8. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1 directory.
 9. Untar the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz file.
 10. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
 11. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
 12. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
 13. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
 14. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
 15. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
 16. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
 17. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
 18. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
 19. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
 20. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
 21. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
 22. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
 23. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.
 24. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0 directory.
 25. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
 26. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
 27. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
 28. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
 29. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
 30. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
 31. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
 32. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
-

33. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
34. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
35. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
36. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
37. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
38. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
39. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
40. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
41. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
42. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
43. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
44. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
45. Perform one of the following steps depending on the Muxer version.
 - For 8.5.500.03, untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-centos7-x86_64-static-gpl3.0.tar.bz2.
 - For 8.5.500.09 or higher versions, untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-rhel8-x86_64-static-gpl3.0.tar.bz2.
46. Execute `chmod a+x ffmpeg` and `chmod a+x ffprobe`.
47. Perform one of the following steps depending on the Muxer version. The OpenSSL library is used to encrypt the resulting muxed recording file when required.
 - For 8.5.500.03, untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-linux-x86_64.tar.bz2.
 - For 8.5.500.09, untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-rhel8-x86_64.tar.bz2.
 - For 8.5.500.10 or higher, untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-3.0.13-rhel8-x86_64.tar.bz2.
48. Execute `chmod a+x openssl`.

Important

- GIR does not support direct upgrade of Muxer from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip`

install command as mentioned above.

- Run `sudo dnf install libnsl` if you encounter the following error while executing Muxer installation script (`install.sh`):
./Perl: error while loading shared libraries: libnsl.so.1: cannot open shared object file: No such file or directory.

Configuring Recording Muxer Script

This section describes how to configure the Recording Muxer Script for your environment.

Configure Passwords (Optional)

Important

In a Linux or Windows environment, Muxer supports the use of environment variables instead of parameters in the configuration file for certain parameters. When both are available, the environment variable take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

- **HTCC_PASSWORD**—maps to the existing configuration parameter under the `htcc` section, password value.
- **RCS_PASSWORD**— maps to the existing configuration parameter under the `rsc` section, password value.
- **WEBDAV_PASSWORD**—maps to the existing configuration parameter under the `webdav` section, password value.

In a Windows only environment, Recording Muxer Script supports storing all passwords in a secure keystore instead of storing in plain-text in the **`muxer.cfg`** file.

1. From the **`muxer`** directory folder in the Recording Muxer installation folder (for example, **<Recording Muxer Install Directory>\muxer**), execute the following command:
`py encryptPassword.py`
The command will prompt for the appropriate values to be entered for the password/key configuration parameters. See the [Genesys Interaction Recording Options Reference](#) for the descriptions of the parameters.
2. Configure the **`muxer.cfg`** file leaving the following parameter values empty:

```
[webdav]
password =
```

```
[htcc]
password=

[rcs]
password =
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Muxer **muxer.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri		Specifies the host and port of the Interaction Recording Web Services (Web Services) server—for example, https://<web services host>:<web services port>/.
contact_center_id		Specifies the unique identifier of the contact center.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note:</p> <ul style="list-style-type: none">If the "Configuring the Secure Password Storage" step was performed, leave this value empty.The password can be overridden by the HTCC_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is

Parameter Name	Default Value	Description
		corrupted. For more information, see Configuring TLS connection to Interaction Recording Web Services on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.
rws_timeout	30	<p>Specifies the timeout duration, in seconds, for Recording Muxer Script while sending a request to Interaction Recording Web Services.</p> <p>Note: The timeout value must be greater than or equal to 30.</p>

Configuring the Connection to Recording Crypto Server

To configure the connection to the Recording Crypto Server, set the following parameters in the **[rcs]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
base_uri	Empty	Specifies the host and port of the Recording Crypto Server: https://<Recording Crypto Server host>:<Recording Crypto Server port>
username	Empty	<p>Specifies the contact center admin username used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section.</p> <p>Note: The user must have the media decrypt permission.</p>
password	Empty	<p>Specifies the contact center admin password used to access the Recording Crypto Server account belonging to the contact center specified by the the contact_center_id option in the [htcc] section.</p> <p>Note:</p> <ul style="list-style-type: none">If the Configuring the Secure Password Storage step was performed, leave this value

Parameter Name	Default Value	Description
		empty. <ul style="list-style-type: none"> The password can be overridden by the <code>RCS_PASSWORD</code> environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Recording Crypto Server on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Processing Commands

- The Recording Muxer uses libraries for analyzing and handling multimedia data. To configure these commands, set the following parameters in the **muxer.cfg** file, the **[processing]** section:
 - ffmpeg** = The path to the ffmpeg executable file.

Important

The ffmpeg executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- ffprobe** = The path to the ffprobe executable file.

Important

The ffprobe executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- To enable Muxer to read multiple screen recordings metadata with one request, configure the following parameters using the **muxer.cfg** configuration file (optional):

- **batch_read_screen_recording_metadata:** Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.
Valid Values: Using Bulk API = 1 / Using previous algorithm the integer ≤ 1
Default Value: 1
- **query_slice_size:** Defines the maximum number of call recording records whose screen recordings should be queried.
Valid Values: all integers > 0
Default Value: 100

3. Configure the **openssl** parameter to set the path to the openssl executable.

Important

- The openssl executable is located under the directory where the thirdparty openssl package was unzipped/untarred.
- On Linux, specifying the absolute path to the openssl executable path is recommended to ensure that the default installed openssl (for example, `/usr/bin/openssl`) is not executed instead.

4. Configure the **window_past** and **window_past_older_than** parameters to set the time in the past to search for the call recordings to multiplex with the screen recordings. See the "Configure HA" section for the recommended values for these parameters.

5. Configure the **clean_temp_folder_timeout** parameter in the **[processing]** section to determine how often the recording files are cleaned up in the **temp folder**. **clean_temp_folder** should only be configured when **auto_clean_temp_folder** is set to 1. By default the **clean_temp_folder** value is 43200 (that is, cleanup occurs every 12 hours). If this value is set to -1, Muxer will attempt to perform a cleanup when it is idle.

For more information about the **[processing]** section parameters, see the [Genesys Interaction Recording Options Reference](#).

Configuring Sharding (Optional)

Sharding can be used to increase the capacity of the Recording Muxer Script solution. When configured, the muxing workload is divided among multiple active instances. By default, Sharding is disabled and `muxer_id = -1`.

When Sharding is in use, a Muxer instance can be configured to run in primary or in backup mode:

- In primary mode, the Muxer should be configured to query for call records from the last `n` minutes (`window_past_older_than=0`, `window_past=n` minutes), based on configuration in the `muxer.cfg` file for that instance.

- In backup mode, the Muxer should be configured to query for call records that are older than the last `n` minutes but newer than `m` minutes (`window_past_older_than= n, window_past= m` minutes), based on configuration in the `muxer.cfg` file for that instance.

Sharding is configured based on the following command line or configuration file parameters within the `[processing]` section:

- **muxer_id:** A unique Muxer ID.
Valid values: A non-negative integer starting with 0 (the Muxer ID should be incremented by 1 for each additional instance).
If you are not using Sharding, the value should be empty or -1.
- **total_muxers:** The total number of primary Muxer instances deployed (excluding the backup).
Valid Values: `max(muxer_id) + 1`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.
- **muxer_type:** indicates if the Muxer is operating in primary mode or backup mode.
Valid Values: `primary, backup`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.

To specify Sharding parameters using the command line, the following arguments are used:

- `muxer-type`
- `muxer-id`
- `total-muxers`

Note: The Sharding parameter values passed in the command line overrides the corresponding values specified within the configuration file. The following is the supported command line:
`python.exe muxer_process.py --config-file=CONFIG_FILE --muxer-type=MUXER_TYPE --muxer-id=MUXER_ID --total-muxers=TOTAL_MUXERS`

For example: When using the following values, the system will have two instances of Muxer running:

- `muxer_type=primary`
- `muxer_id=0` (for the first instance)
- `muxer_id=1` (for the second instance)
- `total_muxers=2`

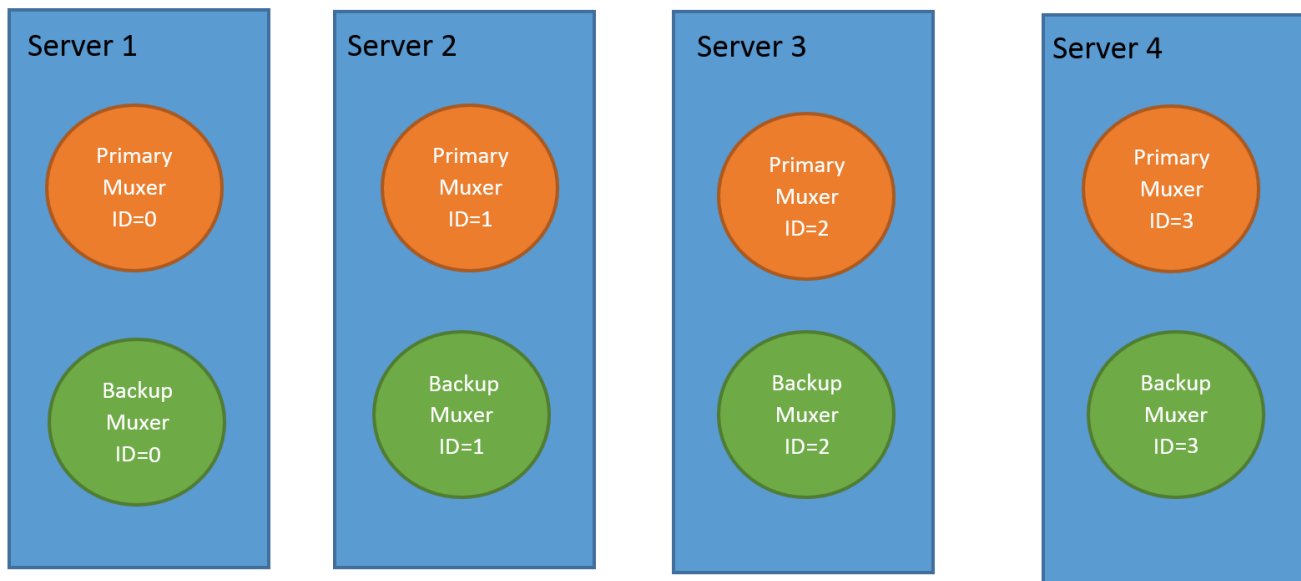
The following is the command line example for running the first instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=0 --total-muxers=2`

The following is the command line example for running the second instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=1 --total-muxers=2`

Note: When there are multiple instances of Muxers deployed on the same machine, then, a different **temp_dir** value for each instance of the Muxer must be configured in the `[processing]` section of the `muxer.cfg` file, and each Muxer instance must use a separate Muxer configuration file. This avoids the issues of one Muxer deleting the temporary files for the other instances.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that each active primary instance be run on a separate machine. For a high availability deployment, a primary instance and a backup instance can be run on the same machine; however, in this case the instances should be configured so that the node IDs overlap (so that a single machine does not provide primary and backup coverage for the same muxer_id).



When running in backup mode, the Muxer will automatically calculate the muxer_id to be used to support this deployment mechanism, based on the specified muxer_id. The configured muxer_id used for the backup instance should match the muxer_id that is configured for the primary instance on the same machine, if both primary mode and backup mode instances are deployed together. For example, if muxer_id=2 and total_muxers=4 in the Muxer configuration file:

- When muxer_type=primary, the muxer_id used will be 2.
- When muxer_type=backup, the muxer_id used will be 3.

Important

If a Muxer instance is added or removed:

- The `total_muxers` value must be changed for each existing Muxer instance.
- All muxer instances must be restarted.
- Before starting the Muxer application, create and configure the **temp_dir** and **logfile_path** folders for both the Primary Muxer instance and the Backup Muxer instances running on the same machine.

Configuring High Availability (HA)

Important

The content in the Configure HA tab only applies if the Sharding configuration is not in use (see: Configure Sharding (Optional) tab). If Sharding is in use, refer to the high availability configuration described in the Configure Sharding (Optional) tab.

Recording Muxer Cluster

The Recording Muxer Script provides High Availability support using multiple instances of the Recording Muxer Script (all active). HA supports:

- Active/active pairs with the aim to load balance equally between the Recording Muxer nodes by splitting and configuring the time window on each node, so that it is close to equal the number of recordings found on each time window.
- When one of the node dies, recordings are still multiplexed.

Limitations:

- If the node with time window, now - $N/2$, dies, multiplexing will still occur, but a slower rate since the second node's time window is from $N/2$ to N .
- If the node with time window, $N/2 - N$, dies, screen recordings that are uploaded with the delay more than $N/2$ might not be multiplexed.
- Nodes should be configured so that the time windows are exclusive of each other, otherwise it may result in two multiplexed files being uploaded.

To configure HA:

1. In each Recording Muxer's **muxer.cfg** configuration file, in the **[processing]** section, set the following values for each node. For example,
 - On first node:
 - **window_past**= 720
 - **window_past_older_than** = 5

2. On second node:

- **window_past** = 1440
- **window_past_older_than** = 725

The above will multiplex all recordings that were recorded within the last 1 day.

3. As a general rule, if the screen recording upload occurs with a delay of N , the configuration on each node can be set to:

- On first node:
 - **window_past** = $N / 2$
 - **window_past_older_than** =
- **min-poll_interval** = $N/200$

4. On second node:

- **window_past** = N
- **window_past_older_than** = $N / 2$
- **min-poll_interval** = $N/200$

Ensure that all Recording Muxer instances have the same configuration other than the above.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that multiple Recording Muxer instances be deployed on different hosts to provide better HA and also not to have machine resource contentions.
- If the recording upload is delayed by more than the time window configured for the Recording Muxer Script, it is possible that the recording will be outside of the processing window and not be multiplexed. For such cases, the Recording Muxer Script can be run as a migration tool to batch process the records matching any desired criteria. For more information see the **call_recording_query_string** parameter under **Configuring the Advanced Options** in the **Advanced Configuration** tab.
- If the screen recording upload is delayed longer than 24 hours, configure a separate Muxer instance or Muxer sharding group for every 12 hours. When the Screen Recording Service is provisioned to upload files during non-business hours, the actual delay can be a couple of days if the agent workstation is shut down when the agent signs off from the Agent Desktop.

Configuring the Connection to WebDAV

To configure the connection to WebDAV, set the following parameters in the **[webdav]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
username	Empty	Specifies the username to allow read/write access to the WebDAV storage server.
password	Empty	<p>Specifies the password to allow read/write access to the WebDAV storage server.</p> <p>Note:</p> <ul style="list-style-type: none">• If multiple WebDAV storage are used for same contact center region, make sure to use the same username and password.• If the "Configuring the Secure Password Storage" step was performed, leave the password value empty.• A password can be overridden by the WEBDAV_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to WebDAV. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to WebDAV on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Advanced Options

The following advanced options can be configured in the **[advanced]** section of the **muxer.cfg** file:

- **worker_threads** = The number of parallel processing threads.
- **pagination** = The maximum number of records returned with each Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) query.

- **max_overlap_allowed** = The overlap time before truncating.
- **video_padding_slice_length_ms** = If the video starts later or ends earlier than the audio, set the duration needed to prepend or append a padded video slice. Genesys recommends to set it to 5000.
- **mark_screen_recording_label** = Whether to apply the label "screenRecording" to the associated call recording metadata after muxing. This configuration is optional. The default value is 1.
- **call_recording_extra_query_string** = Used to specify parameter value pairs other than startTime, endTime, and limit.
If left empty, the **call_recording_extra_query_string** value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if the RWS version is `>= 8.5.201.14`, otherwise, it remains an "" (empty string).
Specify "disable" (without quotes) to force it to be an empty string without checking the RWS version. When the final value of this configuration is not empty, the Recording Muxer Script will continually poll for records that match the searching criteria according to the final value of the configuration that should be processed.
Genesys recommends that this parameter be left empty. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`.
The following table describes values (query parameters) that are available (except startTime and endTime).
- **call_recording_query_string** = When not empty, [call_recording_query_string] queries Interaction Recording Web Services (Web Services) with the given string for records to process. Instead of continually polling for records to process, the Recording Muxer script will exit once the returned records are processed. Genesys recommends that this parameter be left empty unless the Muxer script is to be used for batch migrating the old recordings. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`. The following table describes values (query parameters) that are available:

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified ANI attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard which can substitute any number of any symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified DNIS attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard - which can substitute any number of any symbols in request. Search is case-sensitive.
startTime	Retrieves all recordings that started <code>>=</code> the specified time.
endTime	Retrieves all recordings that ended <code><=</code> the specified time.
userName	Retrieves all recordings in eventHistory- >contacts of which the passed userName/ firstName/Lastname is present. User can use wildcards to specify only part of the username/ firstname/lastname. If more than 1 word is used (divided by spaces) -the records containing any of provided terms as username, firstname or lastname will be included. If user wants to

Parameter Name	Description
	retrieve records containing ALL terms - the AND keyword should be used. Sample: ?userName=Alice AND Amber - will seek for recording with events->contact-> username/firstName/ lastName containing Alice and Amber (possible - in different users). Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap. These matches are supported: <ul style="list-style-type: none"> Exact match - match the entire value (for example, "tom" will find "tom"). Wildcarded value (for example, "tom*" will find a record with "tomas"). Combination of matches - If the query terms are separated by spaces (for example, "tom jerry" will look for recordings that contain "tom" or "jerry").

Configuring the Recording Muxer Using Genesys Administrator Extension (Optional)

The Recording Muxer uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Muxer as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Muxer as a "third party server" application in Genesys Administrator Extension. For more information, see the *Using the Management Layer* section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring Recording Muxer Script to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest Recording Muxer script.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Muxer script of type Third Party Server.
4. Create a new application (for example, myRecordingMuxer) in Genesys Administrator Extension using this new application template.
5. On Windows:
 - a. Set the Command Line parameter to the python executable (for example, C:\Python311\python.exe).
 - b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>\muxer directory. For example, C:\Program Files\GCTI\Recording Muxer Script\muxer.
 - d. Set the Command Line Arguments parameter to the python arguments: muxer_process.py -- config-file=muxer.cfg.
6. On Linux:

- a. Set the Command Line parameter to env.
- b. Set the Host parameter in the application's server info to the correct Host object.
- c. Set the Working Directory parameter to the <Recording Muxer Install Directory>/muxer directory. For example, /opt/genesys/Recording_Muxer_Script_8.5/muxer/.
- d. Set the Command Line Arguments parameter. The LD_LIBRARY_PATH must be set to include the openssl binary directory before muxer script execution. For example, LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:<untarred openssl directory> /opt/python311/python muxer_process.py --config-file=muxer.cfg.

Important

The Recording Muxer does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

Configuring Transport Layer Security (TLS) Connections (Optional)

Python provides the OpenSSL library that is used to establish TLS connections. The OpenSSL library that Python uses is not related to the OpenSSL library installed during installation of third-party libraries, which are used to encrypt muxed recording files.

Configuring TLS connection to Interaction Recording Web Services

1. Set up TLS on Interaction Recording Web Services (RWS). For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the [htcc] section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use https.
3. In the [htcc] section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to true.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, then set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject

alternative name.

Configuring TLS connection to Recording Crypto Server

1. Set up TLS on Recording Crypto Server. For more information, see [Configuring an HTTP Port](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[rcs]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use the secure port.
3. In the **[rcs]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to WebDAV

1. Set up TLS on WebDAV. For more information, see [Configuring TLS for the WebDAV Server](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[webdav]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the

last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

For more information about the Recording Muxer Script parameters, see the [Genesys Interaction Recording Options Reference](#).

Starting the Recording Muxer Script

Important

For **muxer.cfg**, if **temp_dir** is configured, verify that the path exists and is writable by the muxer process.

To launch the Recording Muxer script, run the following command from the <Recording Muxer Install Directory> (where x = 6):

On Windows:

```
<python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

On Linux:

```
env LD_LIBRARY_PATH=<untarred openssl directory>:$LD_LIBRARY_PATH <python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

By default the Recording Muxer's log file is stored in the working directory. This can be changed by specifying a preexisting folder in the **logfile_path** parameter in the **[logfile]** section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingMuxer
```

Recording Muxer Script (Python 3) RHEL 7

Prerequisites

Before installing and configuring the Recording Muxer Script, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.

- A **Recording Crypto Server** 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage or S3 Premise where the recordings are stored.

Installing Recording Muxer Script

Installing on Windows

1. Install 64-bit Python 3.11.5 from the **Python** website.
2. Install **Recording Muxer Script IP** with the installer.

Note: Install the following third-party libraries in the order they appear and untar the files in Administrator mode.

3. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz file.
4. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1 directory.
5. Untar the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz file.
6. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
7. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
8. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
9. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
10. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
11. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
12. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
13. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
14. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
15. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
16. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
17. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
18. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
19. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.
20. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/

pyasn1-0.5.0 directory.

21. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
22. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
23. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
24. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
25. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
26. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
27. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
28. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
29. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
30. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
31. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
32. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
33. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
34. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
35. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
36. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
37. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
38. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
39. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
40. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
41. Unzip the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-win64-static-gpl3.0.zip.
42. Unzip the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-win64.zip. This OpenSSL library is used to encrypt the resulting muxed recording file when required.

Installing on Linux (RHEL)

1. Install zlib-devel (yum install zlib-devel).
2. Install sqlite devel (yum install sqlite-devel.x86_64).

3. Install libffi devel (yum install libffi-devel).
4. Install OpenSSL 1.1.1.
 - For RHEL 7:
 1. Download OpenSSL 1.1.1 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-1.1.1 --openssldir=/usr/home/openssl-1.1.1`
 2. Add OpenSSL lib path in LD_LIBRARY_PATH. Example command - `export LD_LIBRARY_PATH=/usr/home/openssl-1.1.1/lib:$LD_LIBRARY_PATH`
5. Install 64 bit Python 3.11.5 compiled with OpenSSL 1.1.1 from the [Python](#) website.
 - While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
6. Install the **Recording Muxer Script IP** with the installer provided.

Note: Install the following third-party libraries in the order they appear.

7. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz file.
8. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1 directory.
9. Untar the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz file.
10. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
11. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
12. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
13. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
14. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
15. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
16. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
17. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
18. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
19. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
20. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
21. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
22. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
23. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.

-
24. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0 directory.
 25. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
 26. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
 27. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
 28. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
 29. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
 30. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
 31. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
 32. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
 33. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
 34. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
 35. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
 36. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
 37. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
 38. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
 39. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
 40. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
 41. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
 42. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
 43. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
 44. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
 45. Untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-rhel7-x86_64-static-gpl3.0.tar.bz2.
 46. Execute `chmod a+x ffmpeg` and `chmod a+x ffprobe`.
 47. Untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-rhel7-x86_64.tar.bz2.
 48. Execute `chmod a+x openssl`.
-

Important

- GIR does not support direct upgrade of Muxer from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip install` command as mentioned above.
- Run `sudo dnf install libnsl` if you encounter the following error while executing Muxer installation script (`install.sh`):
./Perl: error while loading shared libraries: libnsl.so.1: cannot open shared object file: No such file or directory.

Configuring Recording Muxer Script

This section describes how to configure the Recording Muxer Script for your environment.

Configure Passwords (Optional)

Important

In a Linux or Windows environment, Muxer supports the use of environment variables instead of parameters in the configuration file for certain parameters. When both are available, the environment variable take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

- **HTCC_PASSWORD**—maps to the existing configuration parameter under the `htcc` section, password value.
- **RCS_PASSWORD**— maps to the existing configuration parameter under the `rccs` section, password value.
- **WEBDAV_PASSWORD**—maps to the existing configuration parameter under the `webdav` section, password value.

In a Windows only environment, Recording Muxer Script supports storing all passwords in a secure keystore instead of storing in plain-text in the **muxer.cfg** file.

1. From the **muxer** directory folder in the Recording Muxer installation folder (for example, **<Recording Muxer Install Directory>\(muxer)**), execute the following command:
`py encryptPassword.py`
The command will prompt for the appropriate values to be entered for the password/key configuration parameters. See the [Genesys Interaction Recording Options Reference](#) for the descriptions of the parameters.

2. Configure the **muxer.cfg** file leaving the following parameter values empty:

```
[webdav]
password =

[htcc]
password=

[rsc]
password =
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Muxer **muxer.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri		Specifies the host and port of the Interaction Recording Web Services (Web Services) server—for example, https://<web services host>:<web services port>/.
contact_center_id		Specifies the unique identifier of the contact center.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note:</p> <ul style="list-style-type: none">If the "Configuring the Secure Password Storage" step was performed, leave this value empty.The password can be overridden by the HTCC_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in

Parameter Name	Default Value	Description
		PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web Services on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.
rws_timeout	30	Specifies the timeout duration, in seconds, for Recording Muxer Script while sending a request to Interaction Recording Web Services. Note: The timeout value must be greater than or equal to 30.

Configuring the Connection to Recording Crypto Server

To configure the connection to the Recording Crypto Server, set the following parameters in the **[rcs]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
base_uri	Empty	Specifies the host and port of the Recording Crypto Server: https://<Recording Crypto Server host>:<Recording Crypto Server port>
username	Empty	Specifies the contact center admin username used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section. Note: The user must have the media decrypt permission.
password	Empty	Specifies the contact center admin password used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section.

Parameter Name	Default Value	Description
		Note: <ul style="list-style-type: none">• If the Configuring the Secure Password Storage step was performed, leave this value empty.• The password can be overridden by the RCS_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Recording Crypto Server on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Processing Commands

1. The Recording Muxer uses libraries for analyzing and handling multimedia data. To configure these commands, set the following parameters in the **muxer.cfg** file, the **[processing]** section:
 - **ffmpeg** = The path to the ffmpeg executable file.

Important

The ffmpeg executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- **ffprobe** = The path to the ffprobe executable file.

Important

The ffprobe executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

untarred.

2. To enable Muxer to read multiple screen recordings metadata with one request, configure the following parameters using the **muxer.cfg** configuration file (optional):
 - **batch_read_screen_recording_metadata**: Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.
Valid Values: Using Bulk API = 1 / Using previous algorithm the integer <>1
Default Value: 1
 - **query_slice_size**: Defines the maximum number of call recording records whose screen recordings should be queried.
Valid Values: all integers > 0
Default Value: 100
3. Configure the **openssl** parameter to set the path to the openssl executable.

Important

- The openssl executable is located under the directory where the thirdparty openssl package was unzipped/untarred.
- On Linux, specifying the absolute path to the openssl executable path is recommended to ensure that the default installed openssl (for example, /usr/bin/openssl) is not executed instead.

4. Configure the **window_past** and **window_past_older_than** parameters to set the time in the past to search for the call recordings to multiplex with the screen recordings. See the "Configure HA" section for the recommended values for these parameters.
5. Configure the **clean_temp_folder_timeout** parameter in the **[processing]** section to determine how often the recording files are cleaned up in the **temp folder**. **clean_temp_folder** should only be configured when **auto_clean_temp_folder** is set to 1. By default the **clean_temp_folder** value is 43200 (that is, cleanup occurs every 12 hours). If this value is set to -1, Muxer will attempt to perform a cleanup when it is idle.

For more information about the **[processing]** section parameters, see the [Genesys Interaction Recording Options Reference](#).

Configuring Sharding (Optional)

Sharding can be used to increase the capacity of the Recording Muxer Script solution. When configured, the muxing workload is divided among multiple active instances. By default, Sharding is

disabled and `muxer_id = -1`.

When Sharding is in use, a Muxer instance can be configured to run in primary or in backup mode:

- In primary mode, the Muxer should be configured to query for call records from the last `n` minutes (`window_past_older_than=0`, `window_past=n` minutes), based on configuration in the `muxer.cfg` file for that instance.
- In backup mode, the Muxer should be configured to query for call records that are older than the last `n` minutes but newer than `m` minutes (`window_past_older_than= n`, `window_past= m` minutes), based on configuration in the `muxer.cfg` file for that instance.

Sharding is configured based on the following command line or configuration file parameters within the `[processing]` section:

- **muxer_id:** A unique Muxer ID.
Valid values: A non-negative integer starting with 0 (the Muxer ID should be incremented by 1 for each additional instance).
If you are not using Sharding, the value should be empty or -1.
- **total_muxers:** The total number of primary Muxer instances deployed (excluding the backup).
Valid Values: `max(muxer_id) + 1`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.
- **muxer_type:** indicates if the Muxer is operating in primary mode or backup mode.
Valid Values: `primary`, `backup`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.

To specify Sharding parameters using the command line, the following arguments are used:

- `muxer-type`
- `muxer-id`
- `total-muxers`

Note: The Sharding parameter values passed in the command line overrides the corresponding values specified within the configuration file. The following is the supported command line:
`python.exe muxer_process.py --config-file=CONFIG_FILE --muxer-type=MUXER_TYPE --muxer-id=MUXER_ID --total-muxers=TOTAL_MUXERS`

For example: When using the following values, the system will have two instances of Muxer running:

- `muxer_type=primary`
- `muxer_id=0` (for the first instance)
- `muxer_id=1` (for the second instance)
- `total_muxers=2`

The following is the command line example for running the first instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=0 --total-muxers=2`

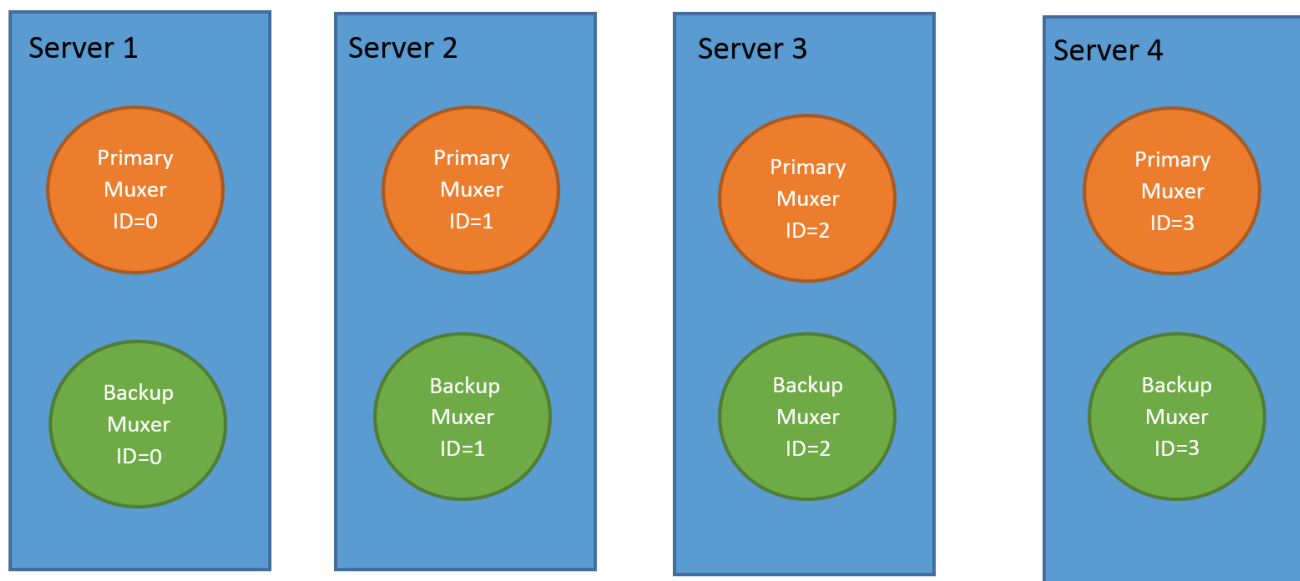
The following is the command line example for running the second instance: `python.exe ../muxer/`

```
muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=1 --total-muxers=2
```

Note: When there are multiple instances of Muxers deployed on the same machine, then, a different **temp_dir** value for each instance of the Muxer must be configured in the [processing] section of the `muxer.cfg` file, and each Muxer instance must use a separate Muxer configuration file. This avoids the issues of one Muxer deleting the temporary files for the other instances.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that each active primary instance be run on a separate machine. For a high availability deployment, a primary instance and a backup instance can be run on the same machine; however, in this case the instances should be configured so that the node IDs overlap (so that a single machine does not provide primary and backup coverage for the same muxer_id).



When running in backup mode, the Muxer will automatically calculate the `muxer_id` to be used to support this deployment mechanism, based on the specified `muxer_id`. The configured `muxer_id` used for the backup instance should match the `muxer_id` that is configured for the primary instance on the same machine, if both primary mode and backup mode instances are deployed together. For example, if `muxer_id=2` and `total_muxers=4` in the Muxer configuration file:

- When `muxer_type=primary`, the `muxer_id` used will be 2.

- When `muxer_type=backup`, the `muxer_id` used will be 3.

Important

If a Muxer instance is added or removed:

- The `total_muxers` value must be changed for each existing Muxer instance.
- All muxer instances must be restarted.
- Before starting the Muxer application, create and configure the **`temp_dir`** and **`logfile_path`** folders for both the Primary Muxer instance and the Backup Muxer instances running on the same machine.

Configuring High Availability (HA)

Important

The content in the Configure HA tab only applies if the Sharding configuration is not in use (see: Configure Sharding (Optional) tab). If Sharding is in use, refer to the high availability configuration described in the Configure Sharding (Optional) tab.

Recording Muxer Cluster

The Recording Muxer Script provides High Availability support using multiple instances of the Recording Muxer Script (all active). HA supports:

- Active/active pairs with the aim to load balance equally between the Recording Muxer nodes by splitting and configuring the time window on each node, so that it is close to equal the number of recordings found on each time window.
- When one of the node dies, recordings are still multiplexed.

Limitations:

- If the node with time window, now - $N/2$, dies, multiplexing will still occur, but a slower rate since the second node's time window is from $N/2$ to N .
- If the node with time window, $N/2 - N$, dies, screen recordings that are uploaded with the delay more than $N/2$ might not be multiplexed.
- Nodes should be configured so that the time windows are exclusive of each other, otherwise it may result in two multiplexed files being uploaded.

To configure HA:

1. In each Recording Muxer's **`muxer.cfg`** configuration file, in the **`[processing]`** section, set the following values for each node. For example,

- On first node:
 - **window_past**= 720
 - **window_past_older_than** = 5
- 2. On second node:
 - **window_past** = 1440
 - **window_past_older_than** = 725

The above will multiplex all recordings that were recorded within the last 1 day.

3. As a general rule, if the screen recording upload occurs with a delay of N , the configuration on each node can be set to:
 - On first node:
 - **window_past** = $N / 2$
 - **window_past_older_than** =
 - **min-poll_interval** = $N/200$
4. On second node:
 - **window_past**= N
 - **window_past_older_than** = $N / 2$
 - **min-poll_interval** = $N/200$

Ensure that all Recording Muxer instances have the same configuration other than the above.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that multiple Recording Muxer instances be deployed on different hosts to provide better HA and also not to have machine resource contentions.
- If the recording upload is delayed by more than the time window configured for the Recording Muxer Script, it is possible that the recording will be outside of the processing window and not be multiplexed. For such cases, the Recording Muxer Script can be run as a migration tool to batch process the records matching any desired criteria. For more information see the **call_recording_query_string** parameter under **Configuring the Advanced Options** in the **Advanced Configuration** tab.
- If the screen recording upload is delayed longer than 24 hours, configure a separate

Muxer instance or Muxer sharding group for every 12 hours. When the Screen Recording Service is provisioned to upload files during non-business hours, the actual delay can be a couple of days if the agent workstation is shut down when the agent signs off from the Agent Desktop.

Configuring the Connection to WebDAV

To configure the connection to WebDAV, set the following parameters in the **[webdav]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
username	Empty	Specifies the username to allow read/write access to the WebDAV storage server.
password	Empty	<p>Specifies the password to allow read/write access to the WebDAV storage server.</p> <p>Note:</p> <ul style="list-style-type: none">• If multiple WebDAV storage are used for same contact center region, make sure to use the same username and password.• If the "Configuring the Secure Password Storage" step was performed, leave the password value empty.• A password can be overridden by the WEBDAV_PASSWORD environment variable.
trusted-ca	false	<p>Configures TLS certificate validation when making a secure outbound connection to WebDAV. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to WebDAV on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.</p>

Configuring the Advanced Options

The following advanced options can be configured in the **[advanced]** section of the **muxer.cfg** file:

- **worker_threads** = The number of parallel processing threads.
- **pagination** = The maximum number of records returned with each Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) query.
- **max_overlap_allowed** = The overlap time before truncating.
- **video_padding_slice_length_ms** = If the video starts later or ends earlier than the audio, set the duration needed to prepend or append a padded video slice. Genesys recommends to set it to 5000.
- **mark_screen_recording_label** = Whether to apply the label "screenRecording" to the associated call recording metadata after muxing. This configuration is optional. The default value is 1.
- **call_recording_extra_query_string** = Used to specify parameter value pairs other than startTime, endTime, and limit.
If left empty, the **call_recording_extra_query_string** value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if the RWS version is `>= 8.5.201.14`, otherwise, it remains an "" (empty string).
Specify "disable" (without quotes) to force it to be an empty string without checking the RWS version. When the final value of this configuration is not empty, the Recording Muxer Script will continually poll for records that match the searching criteria according to the final value of the configuration that should be processed.
Genesys recommends that this parameter be left empty. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`.
The following table describes values (query parameters) that are available (except startTime and endTime).
- **call_recording_query_string** = When not empty, [call_recording_query_string] queries Interaction Recording Web Services (Web Services) with the given string for records to process. Instead of continually polling for records to process, the Recording Muxer script will exit once the returned records are processed. Genesys recommends that this parameter be left empty unless the Muxer script is to be used for batch migrating the old recordings. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`. The following table describes values (query parameters) that are available:

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified ANI attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard which can substitute any number of any symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified DNIS attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard - which can substitute any number of any symbols in request. Search is case-sensitive.
startTime	Retrieves all recordings that started <code>>=</code> the specified time.
endTime	Retrieves all recordings that ended <code><=</code> the

Parameter Name	Description
	specified time.
userName	Retrieves all recordings in eventHistory->contacts of which the passed userName/ firstName/Lastname is present. User can use wildcards to specify only part of the username/ firstname/lastname. If more than 1 word is used (divided by spaces) -the records containing any of provided terms as username, firstname or lastname will be included. If user wants to retrieve records containing ALL terms - the AND keyword should be used. Sample: ?userName=Alice AND Amber - will seek for recording with events->contact-> username/firstName/ lastName containing Alice and Amber (possible - in different users). Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap. These matches are supported: <ul style="list-style-type: none">• Exact match - match the entire value (for example, "tom" will find "tom").• Wildcarded value (for example, "tom*" will find a record with "tomas").• Combination of matches - If the query terms are separated by spaces (for example, "tom jerry" will look for recordings that contain "tom" or "jerry").

Configuring the Recording Muxer Using Genesys Administrator Extension (Optional)

The Recording Muxer uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Muxer as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Muxer as a "third party server" application in Genesys Administrator Extension. For more information, see the *Using the Management Layer* section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring Recording Muxer Script to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest Recording Muxer script.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Muxer script of type Third Party Server.
4. Create a new application (for example, myRecordingMuxer) in Genesys Administrator Extension using this new application template.
5. On Windows:
 - a. Set the Command Line parameter to the python executable (for example, C:\Python311\python.exe).

- b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>\muxer directory. For example, C:\Program Files\GCTI\Recording Muxer Script\muxer.
 - d. Set the Command Line Arguments parameter to the python arguments: muxer_process.py --config-file=muxer.cfg.
6. On Linux:
- a. Set the Command Line parameter to env.
 - b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>/muxer directory. For example, /opt/genesys/Recording_Muxer_Script_8.5/muxer/.
 - d. Set the Command Line Arguments parameter. The LD_LIBRARY_PATH must be set to include the openssl binary directory before muxer script execution. For example, LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:<untarred openssl directory> /opt/python311/python muxer_process.py --config-file=muxer.cfg.

Important

The Recording Muxer does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

Configuring Transport Layer Security (TLS) Connections (Optional)

Python provides the OpenSSL library that is used to establish TLS connections. The OpenSSL library that Python uses is not related to the OpenSSL library installed during installation of third-party libraries, which are used to encrypt muxed recording files.

Configuring TLS connection to Interaction Recording Web Services

1. Set up TLS on Interaction Recording Web Services (RWS). For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[htcc]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use https.
3. In the **[htcc]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to true.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the

certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, then set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to Recording Crypto Server

1. Set up TLS on Recording Crypto Server. For more information, see [Configuring an HTTP Port](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[rcs]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use the secure port.
3. In the **[rcs]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to WebDAV

1. Set up TLS on WebDAV. For more information, see [Configuring TLS for the WebDAV Server](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[webdav]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set

trusted_ca to true.

- If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

For more information about the Recording Muxer Script parameters, see the [Genesys Interaction Recording Options Reference](#).

Starting the Recording Muxer Script

Important

For **muxer.cfg**, if **temp_dir** is configured, verify that the path exists and is writable by the muxer process.

To launch the Recording Muxer script, run the following command from the <Recording Muxer Install Directory> (where x = 6):

On Windows:

```
<python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

On Linux:

```
env LD_LIBRARY_PATH=<untarred openssl directory>:$LD_LIBRARY_PATH <python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

By default the Recording Muxer's log file is stored in the working directory. This can be changed by specifying a preexisting folder in the **logfile_path** parameter in the **[logfile]** section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingMuxer
```

Recording Muxer Script Legacy (Python 2) Deprecated

Important

Recording Muxer Script Legacy (based on Python 2) has been discontinued as of March 31, 2024.

Prerequisites

Before installing and configuring the Recording Muxer Script, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) (or [Web Services](#) if you're using version 8.5.210.02 or earlier) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.

Installing Recording Muxer Script

Installing on Windows

1. Install 32 bit Python 2.7.x from the [Python](#) website.
2. Install the Recording Muxer Script IP.

Note: Install the following third party libraries in the order they appear.

3. Untar the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2.tar.gz file.
4. From the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2 directory, run `python setup.py install`.
5. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1.tar.gz file.
6. From the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1 directory, run `python setup.py install`.
7. Untar the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1.tar.gz file.
8. From the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1 directory, run `python setup.py install`.
9. Untar the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0.tar.gz file.
10. From the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0 directory, run `python setup.py install`.
11. Untar the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5.tar.gz file.
12. From the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5 directory, run

```
python setup.py install.
```

13. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7.tar.gz file.
14. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7 directory, run `python setup.py install`.
15. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5.tar.gz file.
16. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5 directory, run `python setup.py install`.
17. Unzip the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-2.4.3-win64-static-gpl3.0.zip.
18. Unzip the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.0.2j-win64.zip. This OpenSSL library is used to encrypt the resulting muxed recording file when required.

Important

The following steps are only applicable for Muxer 8.5.265.66 or higher.

19. Untar the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1.tar.gz file.
20. From the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1 directory, run `python setup.py install`.
21. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.10.0.tar.gz file.
22. From the <Recording Muxer Install Directory>/thirdparty/six-1.10.0 directory, run `python setup.py install`.
23. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0.tar.gz file.
24. From the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0 directory, run `python setup.py install`.
25. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1.tar.gz file.
26. From the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1 directory, run `python setup.py install`.
27. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57.tar.gz file.
28. From the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57 directory, run `python setup.py install`.
29. Untar the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5.tar.gz file.
30. From the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5 directory, run `python setup.py install`.
31. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10.tar.gz file.
32. From the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10 directory, run `python setup.py install`.
33. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0.tar.gz file.
34. From the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0 directory, run `python setup.py install`.

Installing on Linux (RHEL)

1. Install Python 2.7.6 or later:

- Download the software from the [Python](#) website.
It is recommend that newer versions of Python are installed separately from an existing versions (do not update).

2. Install the Recording Muxer Script IP.

Note: Install the following third party libraries in the order they appear.

3. Untar the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2.tar.gz file.
4. From the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2 directory, run `python setup.py install`.
5. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1.tar.gz file.
6. From the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1 directory, run `python setup.py install`.
7. Untar the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1.tar.gz file.
8. From the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1 directory, run `python setup.py install`.
9. Untar the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0.tar.gz file.
10. From the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0 directory, run `python setup.py install`.
11. Untar the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5.tar.gz file.
12. From the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5 directory, run `python setup.py install`.
13. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7.tar.gz file.
14. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7 directory, run `python setup.py install`.
15. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5.tar.gz file.
16. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5 directory, run `python setup.py install`.
17. Untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/
ffmpeg-2.4.3-centos5-x86_64-static-gpl3.0.tar.bz2.
18. Execute `chmod a+x ffmpeg` and `chmod a+x ffmpegprobe`.
19. Untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.0.2j-centos5-x86_64.tar.bz2. This OpenSSL library is used to encrypt the resulting muxed recording file when required.
20. Execute `chmod a+x openssl`.

Important

The following steps are only applicable for Muxer 8.5.265.66 or higher.

21. Untar the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1.tar.gz file.
22. From the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1 directory, run `python setup.py install`.
23. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.10.0.tar.gz file.
24. From the <Recording Muxer Install Directory>/thirdparty/six-1.10.0 directory, run `python setup.py install`.
25. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0.tar.gz file.
26. From the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0 directory, run `python setup.py install`.
27. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1.tar.gz file.
28. From the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1 directory, run `python setup.py install`.
29. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57.tar.gz file.
30. From the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57 directory, run `python setup.py install`.
31. Untar the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5.tar.gz file.
32. From the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5 directory, run `python setup.py install`.
33. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10.tar.gz file.
34. From the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10 directory, run `python setup.py install`.
35. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0.tar.gz file.
36. From the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0 directory, run `python setup.py install`.

Upgrading Recording Muxer Script

1. Backup the Recording Muxer Script installation directory including logs and configuration file.
2. Uninstall the Recording Muxer Script component.
3. Install the new Recording Muxer Script component.
4. Update the Recording Muxer Script configuration according to the standard installation procedures.

Important

Uninstalling the previous Recording Muxer Script is optional.

Configuring Recording Muxer Script

This section describes how to configure the Recording Muxer Script for your environment.

Configure Passwords (Optional)

Important

In a Linux or Windows environment, Muxer supports the use of environment variables instead of parameters in the configuration file for certain parameters. When both are available, the environment variable take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

- **HTCC_PASSWORD**—maps to the existing configuration parameter under the htcc section, password value.
- **RCS_PASSWORD**— maps to the existing configuration parameter under the rcs section, password value.
- **WEBDAV_PASSWORD**—maps to the existing configuration parameter under the webdav section, password value.

In a Windows only environment, Recording Muxer Script supports storing all passwords in a secure keystore instead of storing in plain-text in the **muxer.cfg** file.

1. From the **muxer** directory folder in the Recording Muxer installation folder (for example, **<Recording Muxer Install Directory>\(muxer)**, execute the following command:
`python encryptPassword.py`
The command will prompt for the appropriate values to be entered for the password/key configuration parameters. See the [Genesys Interaction Recording Options Reference](#) for the descriptions of the parameters.
2. Configure the **muxer.cfg** file leaving the following parameter values empty:

```
[webdav]
password =

[htcc]
password=

[rcs]
password =
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Muxer **muxer.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri		Specifies the host and port of the Interaction Recording Web Services (Web Services) server—for example, <code>https://<web_services_host>:<web_services_port>/</code> .
contact_center_id		Specifies the unique identifier of the contact center.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note:</p> <ul style="list-style-type: none">• If the "Configuring the Secure Password Storage" step was performed, leave this value empty.• The password can be overridden by the <code>HTCC_PASSWORD</code> environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are <code>true</code> , <code>false</code> , and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web Services on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Parameter Name	Default Value	Description
rws_timeout	30	Specifies the timeout duration, in seconds, for Recording Muxer Script while sending a request to Interaction Recording Web Services. Note: The timeout value must be greater than or equal to 30.

Configuring the Connection to Recording Crypto Server

To configure the connection to the Recording Crypto Server, set the following parameters in the **[rcs]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
base_uri	Empty	Specifies the host and port of the Recording Crypto Server: https://<Recording Crypto Server host>:<Recording Crypto Server port>
username	Empty	Specifies the contact center admin username used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section. Note: The user must have the media decrypt permission.
password	Empty	Specifies the contact center admin password used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section. Note: <ul style="list-style-type: none">• If the Configuring the Secure Password Storage step was performed, leave this value empty.• The password can be overridden by the RCS_PASSWORD environment variable.

Parameter Name	Default Value	Description
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Recording Crypto Server on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Processing Commands

1. The Recording Muxer uses libraries for analyzing and handling multimedia data. To configure these commands, set the following parameters in the **muxer.cfg** file, the **[processing]** section:

- **ffmpeg** = The path to the ffmpeg executable file.

Important

The ffmpeg executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- **ffprobe** = The path to the ffprobe executable file.

Important

The ffprobe executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

2. To enable Muxer to read multiple screen recordings metadata with one request, configure the following parameters using the **muxer.cfg** configuration file (optional):
 - **batch_read_screen_recording_metadata**: Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.
Valid Values: Using Bulk API = 1 / Using previous algorithm the integer <>1
Default Value: 1
 - **query_slice_size**: Defines the maximum number of call recording records whose screen recordings should be queried.

Valid Values: all integers > 0
Default Value: 100

3. Configure the **openssl** parameter to set the path to the openssl executable.

Important

- The openssl executable is located under the directory where the thirdparty openssl package was unzipped/untarred.
- On Linux, specifying the absolute path to the openssl executable path is recommended to ensure that the default installed openssl (for example, /usr/bin/openssl) is not executed instead.

4. Configure the **window_past** and **window_past_older_than** parameters to set the time in the past to search for the call recordings to multiplex with the screen recordings. See the "Configure HA" section for the recommended values for these parameters.
5. Configure the **clean_temp_folder_timeout** parameter in the **[processing]** section to determine how often the recording files are cleaned up in the **temp folder**. **clean_temp_folder** should only be configured when **auto_clean_temp_folder** is set to 1. By default the **clean_temp_folder** value is 43200 (that is, cleanup occurs every 12 hours). If this value is set to -1, Muxer will attempt to perform a cleanup when it is idle.

For more information about the **[processing]** section parameters, see the [Genesys Interaction Recording Options Reference](#).

Configuring Sharding (Optional)

Sharding can be used to increase the capacity of the Recording Muxer Script solution. When configured, the muxing workload is divided among multiple active instances. By default, Sharding is disabled and `muxer_id = -1`.

When Sharding is in use, a Muxer instance can be configured to run in primary or in backup mode:

- In primary mode, the Muxer should be configured to query for call records from the last n minutes (`window_past_older_than=0, window_past=n minutes`), based on configuration in the `muxer.cfg` file for that instance.
- In backup mode, the Muxer should be configured to query for call records that are older than the last n minutes but newer than m minutes (`window_past_older_than= n, window_past= m minutes`), based on configuration in the `muxer.cfg` file for that instance.

Sharding is configured based on the following command line or configuration file parameters within the **[processing]** section:

- **muxer_id:** A unique Muxer ID.
Valid values: A non-negative integer starting with 0 (the Muxer ID should be incremented by 1 for each additional instance).
If you are not using Sharding, the value should be empty or -1.
- **total_muxers:** The total number of primary Muxer instances deployed (excluding the backup).
Valid Values: $\max(\text{muxer_id}) + 1$
If you are not using Sharding, (indicated by muxer_id not being set, or being set to -1), the Muxer ignores this value.
- **muxer_type:** indicates if the Muxer is operating in primary mode or backup mode.
Valid Values: primary, backup
If you are not using Sharding, (indicated by muxer_id not being set, or being set to -1), the Muxer ignores this value.

To specify Sharding parameters using the command line, the following arguments are used:

- muxer-type
- muxer-id
- total-muxers

Note: The Sharding parameter values passed in the command line overrides the corresponding values specified within the configuration file. The following is the supported command line:
`python.exe muxer_process.py --config-file=CONFIG_FILE --muxer-type=MUXER_TYPE --muxer-id=MUXER_ID --total-muxers=TOTAL_MUXERS`

For example: When using the following values, the system will have two instances of Muxer running:

- muxer_type=primary
- muxer_id=0 (for the first instance)
- muxer_id=1 (for the second instance)
- total_muxers=2

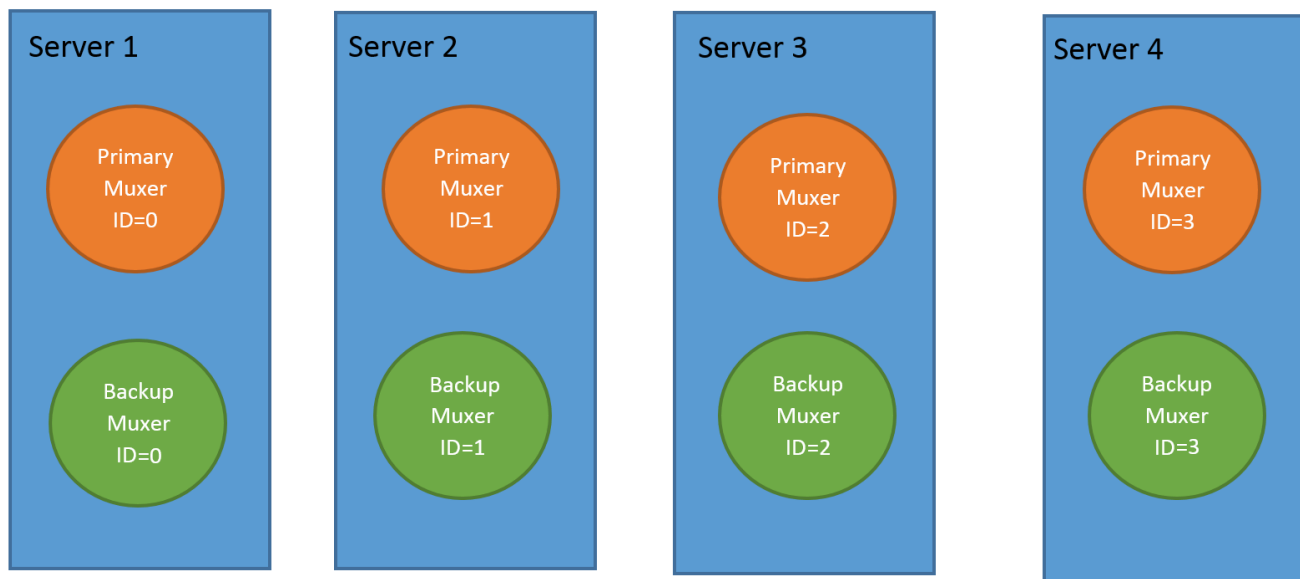
The following is the command line example for running the first instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=0 --total-muxers=2`

The following is the command line example for running the second instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=1 --total-muxers=2`

Note: When there are multiple instances of Muxers deployed on the same machine, then, a different **temp_dir** value for each instance of the Muxer must be configured in the [processing] section of the muxer.cfg file, and each Muxer instance must use a separate Muxer configuration file. This avoids the issues of one Muxer deleting the temporary files for the other instances.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that each active primary instance be run on a separate machine. For a high availability deployment, a primary instance and a backup instance can be run on the same machine; however, in this case the instances should be configured so that the node IDs overlap (so that a single machine does not provide primary and backup coverage for the same muxer_id).



When running in backup mode, the Muxer will automatically calculate the muxer_id to be used to support this deployment mechanism, based on the specified muxer_id. The configured muxer_id used for the backup instance should match the muxer_id that is configured for the primary instance on the same machine, if both primary mode and backup mode instances are deployed together. For example, if muxer_id=2 and total_muxers=4 in the Muxer configuration file:

- When muxer_type=primary, the muxer_id used will be 2.
- When muxer_type=backup, the muxer_id used will be 3.

Important

If a Muxer instance is added or removed:

- The `total_muxers` value must be changed for each existing Muxer instance.
- All muxer instances must be restarted.
- Before starting the Muxer application, create and configure the **temp_dir** and **logfile_path** folders for both the Primary Muxer instance and the Backup Muxer instances running on the same machine.

Configuring High Availability (HA)

Important

The content in the Configure HA tab only applies if the Sharding configuration is not in use (see: Configure Sharding (Optional) tab). If Sharding is in use, refer to the high availability configuration described in the Configure Sharding (Optional) tab.

Recording Muxer Cluster

The Recording Muxer Script provides High Availability support using multiple instances of the Recording Muxer Script (all active). HA supports:

- Active/active pairs with the aim to load balance equally between the Recording Muxer nodes by splitting and configuring the time window on each node, so that it is close to equal the number of recordings found on each time window.
- When one of the node dies, recordings are still multiplexed.

Limitations:

- If the node with time window, now - $N/2$, dies, multiplexing will still occur, but a slower rate since the second node's time window is from $N/2$ to N .
- If the node with time window, $N/2 - N$, dies, screen recordings that are uploaded with the delay more than $N/2$ might not be multiplexed.
- Nodes should be configured so that the time windows are exclusive of each other, otherwise it may result in two multiplexed files being uploaded.

To configure HA:

1. In each Recording Muxer's **muxer.cfg** configuration file, in the **[processing]** section, set the following values for each node. For example,
 - On first node:
 - **window_past**= 720
 - **window_past_older_than** = 5

2. On second node:

- **window_past** = 1440
- **window_past_older_than** = 725

The above will multiplex all recordings that were recorded within the last 1 day.

3. As a general rule, if the screen recording upload occurs with a delay of N , the configuration on each node can be set to:

- On first node:
 - **window_past** = $N / 2$
 - **window_past_older_than** =
- **min-poll_interval** = $N/200$

4. On second node:

- **window_past** = N
- **window_past_older_than** = $N / 2$
- **min-poll_interval** = $N/200$

Ensure that all Recording Muxer instances have the same configuration other than the above.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that multiple Recording Muxer instances be deployed on different hosts to provide better HA and also not to have machine resource contentions.
- If the recording upload is delayed by more than the time window configured for the Recording Muxer Script, it is possible that the recording will be outside of the processing window and not be multiplexed. For such cases, the Recording Muxer Script can be run as a migration tool to batch process the records matching any desired criteria. For more information see the **call_recording_query_string** parameter under **Configuring the Advanced Options** in the **Advanced Configuration** tab.
- If the screen recording upload is delayed longer than 24 hours, configure a separate Muxer instance or Muxer sharding group for every 12 hours. When the Screen Recording Service is provisioned to upload files during non-business hours, the actual delay can be a couple of days if the agent workstation is shut down when the agent signs off from the Agent Desktop.

Configuring the Connection to WebDAV

To configure the connection to WebDAV, set the following parameters in the **[webdav]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
username	Empty	Specifies the username to allow read/write access to the WebDAV storage server.
password	Empty	<p>Specifies the password to allow read/write access to the WebDAV storage server.</p> <p>Note:</p> <ul style="list-style-type: none">• If multiple WebDAV storage are used for same contact center region, make sure to use the same username and password.• If the "Configuring the Secure Password Storage" step was performed, leave the password value empty.• A password can be overridden by the WEBDAV_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to WebDAV. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to WebDAV on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Advanced Options

The following advanced options can be configured in the **[advanced]** section of the **muxer.cfg** file:

- **worker_threads** = The number of parallel processing threads.
- **pagination** = The maximum number of records returned with each Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) query.

- **max_overlap_allowed** = The overlap time before truncating.
- **video_padding_slice_length_ms** = If the video starts later or ends earlier than the audio, set the duration needed to prepend or append a padded video slice. Genesys recommends to set it to 5000.
- **mark_screen_recording_label** = Whether to apply the label "screenRecording" to the associated call recording metadata after muxing. This configuration is optional. The default value is 1.
- **call_recording_extra_query_string** = Used to specify parameter value pairs other than startTime, endTime, and limit.
If left empty, the **call_recording_extra_query_string** value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if the RWS version is `>= 8.5.201.14`, otherwise, it remains an "" (empty string).
Specify "disable" (without quotes) to force it to be an empty string without checking the RWS version. When the final value of this configuration is not empty, the Recording Muxer Script will continually poll for records that match the searching criteria according to the final value of the configuration that should be processed.
Genesys recommends that this parameter be left empty. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`.
The following table describes values (query parameters) that are available (except startTime and endTime).
- **call_recording_query_string** = When not empty, [call_recording_query_string] queries Interaction Recording Web Services (Web Services) with the given string for records to process. Instead of continually polling for records to process, the Recording Muxer script will exit once the returned records are processed. Genesys recommends that this parameter be left empty unless the Muxer script is to be used for batch migrating the old recordings. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`. The following table describes values (query parameters) that are available:

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified ANI attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard which can substitute any number of any symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified DNIS attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard - which can substitute any number of any symbols in request. Search is case-sensitive.
startTime	Retrieves all recordings that started <code>>=</code> the specified time.
endTime	Retrieves all recordings that ended <code><=</code> the specified time.
userName	Retrieves all recordings in eventHistory- >contacts of which the passed userName/ firstName/Lastname is present. User can use wildcards to specify only part of the username/ firstname/lastname. If more than 1 word is used (divided by spaces) -the records containing any of provided terms as username, firstname or lastname will be included. If user wants to

Parameter Name	Description
	retrieve records containing ALL terms - the AND keyword should be used. Sample: ?userName=Alice AND Amber - will seek for recording with events->contact-> username/firstName/ lastName containing Alice and Amber (possible - in different users). Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap. These matches are supported: <ul style="list-style-type: none">• Exact match - match the entire value (for example, "tom" will find "tom").• Wildcarded value (for example, "tom*" will find a record with "tomas").• Combination of matches - If the query terms are separated by spaces (for example, "tom jerry" will look for recordings that contain "tom" or "jerry").

Configuring the Recording Muxer Using Genesys Administrator Extension (Optional)

The Recording Muxer uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Muxer as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Muxer as a "third party server" application in Genesys Administrator Extension. For more information, see the *Using the Management Layer* section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring Recording Muxer Script to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest Recording Muxer script.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Muxer script of type Third Party Server.
4. Create a new application (for example, myRecordingMuxer) in Genesys Administrator Extension using this new application template.
5. On Windows:
 - a. Set the Command Line parameter to the python executable (for example, C:\Python27\python.exe).
 - b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>\muxer directory. For example, C:\Program Files\GCTI\Recording Muxer Script\muxer.
 - d. Set the Command Line Arguments parameter to the python arguments: muxer_process.py -- config-file=muxer.cfg.
6. On Linux:

- a. Set the Command Line parameter to env.
- b. Set the Host parameter in the application's server info to the correct Host object.
- c. Set the Working Directory parameter to the <Recording Muxer Install Directory>/muxer directory. For example, /opt/genesys/Recording_Muxer_Script_8.5/muxer/.
- d. Set the Command Line Arguments parameter. The LD_LIBRARY_PATH must be set to include the openssl binary directory before muxer script execution. For example,
LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:<untarred openssl directory> /opt/python27/python muxer_process.py --config-file=muxer.cfg.

Important

The Recording Muxer does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

Configuring Transport Layer Security (TLS) Connections (Optional)

Python provides the OpenSSL library that is used to establish TLS connections. To use a newer version of OpenSSL, upgrade the version of Python being used (within the 2.7.x family). The OpenSSL library that Python uses is not related to the OpenSSL library installed during installation of third-party libraries, which are used to encrypt muxed recording files.

Configuring TLS connection to Interaction Recording Web Services

1. Set up TLS on Interaction Recording Web Services (RWS). For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the [htcc] section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use https.
3. In the [htcc] section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to true.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, then set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject

alternative name.

Configuring TLS connection to Recording Crypto Server

1. Set up TLS on Recording Crypto Server. For more information, see [Configuring an HTTP Port](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[rcs]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use the secure port.
3. In the **[rcs]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to WebDAV

1. Set up TLS on WebDAV. For more information, see [Configuring TLS for the WebDAV Server](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[webdav]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the

last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

For more information about the Recording Muxer Script parameters, see the [Genesys Interaction Recording Options Reference](#).

Starting the Recording Muxer Script

Important

For **muxer.cfg**, if **temp_dir** is configured, verify that the path exists and is writable by the muxer process.

To launch the Recording Muxer script, run the following command from the <Recording Muxer Install Directory> (where x = 6):

On Windows:

```
<python2.7.x executable> muxer_process.py --config-file=muxer.cfg
```

On Linux:

```
env LD_LIBRARY_PATH=<untarred openssl directory>:$LD_LIBRARY_PATH <python2.7.x executable> muxer_process.py --config-file=muxer.cfg
```

By default the Recording Muxer's log file is stored in the working directory. This can be changed by specifying a preexisting folder in the **logfile_path** parameter in the **[logfile]** section of the configuration file. For example:

```
logfile_path = C:\logs\recordingMuxer
```


Deploying SpeechMiner for GIR

GIR uses the SpeechMiner application to play back call and screen recordings that are stored in the GIR system. Note that call or screen recordings that have been backed up and then purged from the GIR system cannot be played back through SpeechMiner. These should be played with your own media player.

Installing SpeechMiner

To install and configure SpeechMiner, follow the instructions in the [Deploying SpeechMiner](#) topic of the *SpeechMiner Administration Guide*. Pay attention to the instructions specific to Analytics and Recording UI, or Recording UI Only modes.

Important

The UConnector service is not required for GIR.

Important

To access this content:

- **Customers:** Log in to [My Support](#) and select *Documentation*.
- **Partners:** Log in to [Partner Portal](#) and select *Genesys Technical Docs*.
- **Employees:** Go to the [internal access point](#).

Upgrading SpeechMiner

For information about upgrading your SpeechMiner components, see the [SpeechMiner 8.5.x Upgrade Guide](#).

The SpeechMiner documents are restricted and require specific credentials to access the content. If you have a login, you can access the docs here: [Login](#).

If you are a Genesys employee:

- To access the online documents and if you don't have an account on this site, click the **Employee Login / Restricted Content** link in the lower-right corner of the page. Create your login and send your username to Tech Pubs Admins so that we can grant you access.

If you are a Genesys Partner or customer:

- Contact the product manager to access to these documents.

If you are upgrading SpeechMiner, you must upload and deploy the Solution Definition (SPD) files in order to gain access to any new Genesys Administrator Extension roles or privileges.

For more information about uploading the SPD file, see the [Upgrading the Plug-in](#).

Configuring SpeechMiner users

In addition to the configuration that is described in the *SpeechMiner Administration Guide*, the SpeechMiner database and application must be configured specifically for Genesys Interaction Recording (GIR).

1. Use Genesys Administrator Extension to create four new Application Templates:
 - a. Import the following templates from the SpeechMiner CD:
 - Speechminer_ClientApplications.apd
 - Speechminer_InteractionReceiver.apd
 - Speechminer_Platform.apd
 - Speechminer_Web.apd
 - b. Verify that each template has Genesys Generic Server in the **Type** field.
 - c. Verify that the log parameters are defined.
2. Create four new Application objects using the newly created templates.
 - a. Enter the names in the **Name** field as they are indicated in the template. For example, name SpeechMiner Platform application, Speechminer_Platform, where SpeechMiner is the default and can be changed in the SMConfig Recording panel. The name used and the name in the SMConfig Recording panel should match.
 - b. Set the hosts.
 - c. The Start info Working Directory, Command Line must not be empty. Set it to ". ".

Important

The SpeechMiner components do not integrate with LCA.

3. Use Genesys Administrator Extension to create an additional new Application Template:

- a. Import the following template from the SpeechMiner CD:
 - `Speechminer_node.apd`
 - b. Verify that the template has Genesys Generic Client in the **Type** field.
4. Create three new Application objects.
- a. In the **Name** field, enter the name of each application object. The three new Application objects should be named as follows: (where SpeechMiner is the default and can be changed in the SMConfig Recording panel. The name used and the name in the SMConfig Recording panel should match):
 - `Speechminer_Platform_Node`
 - `Speechminer_InteractionReceiver_Node`
 - `Speechminer_Web_Node`
 - b. Create a connection for each Application object to the Server application with the similar name. For example, for *Speechminer_Web_Node* use the name *Speechminer_Web*.
5. In **Genesys Administrator Extension**, navigate to **Configuration > Environment > Tenants** and in the **Options** tab of each Tenant (including the Environment tab), in the **[recording.archive]** section configure the following parameters:
- user
 - password

Important

The user and password value must be the same as the username and password configured in both of the following sections:

- **Configuring SpeechMiner Interaction Receiver Authorization Header** in the **Recording Destinations** section of **IVR profile configuration**.
- **Configuring SpeechMiner settings** in RWS.

6. In the SMConfig-Login screen (SpeechMiner Configuration), login with your Configuration Server credentials (for example, default/password) and set the Configuration Server host and port.

Important

When you are logging in for the first time, you must go to the license tab, and add your recording-only license, and login to SM Config again. You will see the **Recording** tab in the UI only after you have added the license .

7. In the Sites and Machines/Machines and Tasks screen:
- Configure the Interaction Receiver tasks.

- Click **Select Languages** and select **English USA**.

8. Perform the following:

- Configure the **User Application Name** so that it refers to the application you want to use for authenticating users. By default, it is **default** and should only be changed in a multi-tenant environment, for security reasons. For details, refer to the [Configuring SpeechMiner users](#) section in the Permissions page.
- Enable the following tasks in the SpeechMiner configuration (Recording Only mode):
 - web server
 - interaction receiver
 - indexer
 - uplatform

9. In the Media panel of SMConfig, set the following to avoid creating unnecessary audio files and storing them for too long (Analytics Only mode):

Parameter	Value
Recognition Audio Format	WAV_PCM
Create compressed audio file	Do not Generate
WAV_PCM Retention Period	0

10. In the Recording tab, set the parameters.

[+] Show the table that describes the parameters.

Section	Parameter	Description	Example
Configuration	Tenant	Specifies the tenant as configured in Genesys Administrator Extension. Note: For single-tenant contact centers, the Tenant should match the tenant used in the configuration server.	Resources
	User and Password	The Configuration Server user and password that SpeechMiner applications should use when connecting to the Configuration Server. Verify that the specific user was given read and execute permissions for the tenant object in the configuration server and all its objects in the tenant object	

Section	Parameter	Description	Example
		hierarchy.	
	Application Name	Specifies the prefix of the SpeechMiner application objects as configured in Genesys Administrator Extension.	SpeechMiner
	User Application Name	The name of the Configuration Manager application object that will be used to validate user credentials.	
	Update Agents Every	Specifies how often to update the agent tree in the user interface.	24 hours
Interaction Receiver	Default Program	Specifies the Program ID to be used if the Recording Processor Script does not assign a Program ID to the call.	default
	Extension Speaker Type	Specifies the type of speaker to be used for the extension side of the call.	agent
	Trunk Speaker Type	Specifies the type of speaker to be used for the trunk side of the call.	customer
RP Authorization Note: The values of RP Authorization User and Password must match the values that are configured in the SpeechMiner Interaction Receiver Authorization Header in the IVR Profile.	User	Specifies the username used by the Recording Processor Script when posting metadata to the SpeechMiner Interaction Receiver.	<rp_username>
	Password	Specifies the password used by the Recording Processor Script when posting metadata to the SpeechMiner Interaction Receiver.	<rp_password>
MCP Authorization Note: The values of MCP Authorization User and Password must match the values that are configured in the SpeechMiner HTTP Authorization Header in the IVR Profile. Leave these parameters empty unless you have purchased and enabled speech analytics	User	Specifies the username used by the MCP when posting files to the SpeechMiner Interaction Receiver.	<username>
	Password	Specifies the password used by the MCP when posting files to the SpeechMiner Interaction Receiver.	<password>

Section	Parameter	Description	Example
mode on SpeechMiner.			
Playback	RCS URI	Specifies the URI that the SpeechMiner Web Service component uses to communicate with the Recording Crypto Server as a server-to-server connection. This parameter is used for playback of call recordings.	http://<Recording Crypto Server Host>:<port>/rcs or https://<Recording Crypto Server Host>:<port>/rcs
	RWS URI	Specifies the URI that the SpeechMiner Web Service component uses to communicate with Interaction Recording Web Services as a server-to-server connection. This parameter must be set to use the tagging or deletion protection functionality. Note: You must disable CSRF protection functionality in RWS if you are using tagging or deletion protection.	http://<Interaction Recording Web Services host>:<port> or https://<Interaction Recording Web Services host>:<port>
	External RCS URI	Specifies the URI that the SpeechMiner browser application uses to access the Recording Crypto Server. Note: This parameter is required to playback encrypted screen recordings, unless you have configured the local decrypt URI prefix (refer to Local Decrypt URI Prefix for Call Recording and Screen Recording).	http://<Recording Crypto Server Host>:<port>/rcs or https://<Recording Crypto Server Host>:<port>/rcs
	External RWS URI	Specifies the URI that the SpeechMiner browser application uses to access Interaction Recording Web Services. Note: This parameter is required for screen recording playback only. If you do not	http://<Interaction Recording Web Services host>:<port> or https://<Interaction Recording Web Services host>:<port>

Section	Parameter	Description	Example
		want to use screen recording, leave this value blank.	

11. Depending on the license you are using and only after you have verified that the license matches the installation mode, perform one of the following:
- In **Recording and Analytics** mode, create and apply a program in SMART for every Program ID that RP may send.

Important

Both of these actions can be skipped, if the default in the database is satisfactory, or if the Recording Processor Script includes the program ID in the metadata.

On a per-call basis, the attached data key `GRECORD_PROGRAM` can be set to define the program external ID to be used for this call. For example, attached data can be set in a routing strategy.

12. Configure the roles and permissions for the **SpeechMiner Users**.

Deploying Workspace Desktop Edition for GIR

You can use the Workspace Desktop Edition (formerly known as Interaction Workspace) for GIR as the agent's desktop.

Installing Workspace Desktop Edition

To install and configure the Workspace Desktop Edition, see the [Workspace Desktop Edition 8.5 Deployment Guide](#). You can learn more about the desktop [here](#).

Configuring Workspace Desktop Edition

In addition to the configuration described in the deployment guide, you must configure the Workspace Desktop Edition as follows:

1. Set the MSML recording parameters:

Parameter Name	Value
<code>active-recording.voice.recorder-uri</code>	Leave empty. The file recording destination is configured through the GVP IVR Profile.
<code>active-recording.voice.recording-type</code>	MSML

2. Configure the desktop to attach data:

- In the `interaction-workspace` section, set `interaction.case-data.format-business-attribute=CaseData` where `CaseData` is the name of the Business Attributes object that contains a list of attributes that are the attached data keys.

To allow integration with the Screen Recording Client, see:

[Integrating with Workspace Desktop Edition](#)

Configuring permissions, access control, and privacy

The following sections describe, and provide examples of how to configure access control for Genesys Interaction Recording Users.

For more information about controlling the access for voice recording users, see [Access Control for Voice Recording Users](#).

Configuring SpeechMiner roles and permissions

Configuring SpeechMiner users

All SpeechMiner users must be assigned to the Users Access Group. If agent hierarchy and partition features are not used, assign all the SpeechMiner users to the / (slash) Access Group. If agent hierarchy or partition features are used, the users must be granted to the specific Access Groups in order to be able to access recordings for the various agent hierarchy and partitions.

Important

- To restrict log-in to the SpeechMiner UI, a new Configuration Manager application object must be created. Backup the default Configuration Manager object, since this object is accessible by all users from all tenants. The new Configuration Manager application object should be configured to allow Environment administrators, Environment users and Super administrators access to it.
- To see members in the User Access Group (by default, SpeechMiner Users) in the Speechminer UI, Log On As the account of Speechminer_WEB application should have Read rights to User Access Group.

You must configure Genesys Interaction Recording to enable the SpeechMiner UI search option to display a list of agent names:

1. In the Agent's **Person** object, create a **[recording]** section in the **Annex** (if it doesn't already exist).
2. Add the **agent_hierarchy** option in the **[recording]** section, and set the value to slash: "/" or what is appropriate for access control.
3. Repeat these steps for any additional agents that might be searched for in the SpeechMiner UI.
4. This configuration will not take effect until the SpeechMiner cache is updated:
 - In the **SMConfig > Recording** tab, update the **Update Agents Every** parameter to the number of hours between the SpeechMiner person object updates. SpeechMiner will check the Configuration

Server according to this option to retrieve the list of person objects under the **Recording folder** access group. The names of these agents are then available when searching for call recordings or screen recordings.

- To force the list of agents to update sooner, update the **NextAgentsUpdate** column in the **configServer** table of the SpeechMiner database to a date in the near future.

Important

- The Access Group / (forward slash) grants access to all recordings.

The following is a screen shot showing the assignment of Access Group members to /Anthony/Paul in Genesys Administrator Extension:

The screenshot displays the Genesys Administrator Extension interface. The top navigation bar includes tabs for GAX, System Dashboard, Agents, Configuration, Routing Parameters, and Administration. The main content area shows the 'Persons' table with columns for Name, First Name, Last Name, and Agent. A red arrow points to the 'Agent' column, and another red arrow points to the 'Save' button.

	Name	First Name	Last Name	Agent
<input checked="" type="checkbox"/>	agent1000	Firstname1000	lastname	<input checked="" type="checkbox"/>

The Recording Plug-in for GAX includes a **Solution Definition (SPD) file** that can be used to configure roles and access groups.

Configuring roles

For information about configuring roles for Genesys Interaction Recording users, see [Role Privileges](#) in the Genesys Administrator Extension Deployment Guide.

Configuring permissions for recording labels

A label definition defines a label, which can then be applied to a recording. For example, a label definition could be created to mark a recording for further review.

Permissions are required to perform these operations. You can configure the label permissions using Genesys Administrator Extension (GAX), in the IRWS_Cluster (or WS_Cluster, where applicable) application object or the Person object.

To configure label permissions, do this:

1. If you're configuring this at the application level: add a new recording settings group to the Annex/ Application options group for the IRWS_Cluster (or WS_Cluster where applicable) application object, or update the existing recording group. For details, refer to [Genesys Administrator Extension Help: Configuration Manager](#) and [Installing Interaction Recording Web Services](#).

Important

You are not required to do it this way; you can also set this at the Person object level.

2. Configure one or all of the following options in the recording settings as follows:

```
[recording]
RECORDING_PERMISSION_ADD_LABEL_DEFINITION = true
RECORDING_PERMISSION_DELETE_LABEL_DEFINITION = true
RECORDING_PERMISSION_ADD_LABEL = true
RECORDING_PERMISSION_DELETE_LABEL = true
```

The system applies permissions in the following order:

1. Default settings that are defined in the application code, which are overridden by:
2. Settings that are specified in the Application, which are overridden by:
3. Settings that are specified in the Person object that corresponds to the agent.

The following permissions are required to allow users with the role supervisor or agent to be able to access and use the different label operations.

Permission	Description	Applies to	Checks against
RECORDING_PERMISSION_ADD_LABEL_DEFINITION	Permission to create a label definition	<ul style="list-style-type: none"> Creating a label definition Updating a label definition 	<ul style="list-style-type: none"> Supervisor Agent
RECORDING_PERMISSION_DELETE_LABEL_DEFINITION	Permission to delete a label definition	<ul style="list-style-type: none"> Deleting a label definition 	<ul style="list-style-type: none"> Supervisor Agent
RECORDING_PERMISSION_ADD_LABEL	Permission to add/Update label(s) on a recording	<ul style="list-style-type: none"> Adding a label to a recording Updating a label on a recording Adding a label to multiple recordings 	<ul style="list-style-type: none"> Supervisor Agent
RECORDING_PERMISSION_DELETE_LABEL	Permission to delete label(s) from a recording	<ul style="list-style-type: none"> Deleting a label from a recording 	<ul style="list-style-type: none"> Supervisor Agent

Configuring Permissions for Recording Non-Deletion

You can protect recordings from deletion using SpeechMiner, or using the [Recording Non-Deletion API](#), if you have the appropriate permissions that are required.

You can configure the non-deletion permissions using Genesys Administrator Extension (GAX), in the Configuration Manager view, the IRWS_Cluster (or WS_Cluster where applicable) application object or the Person object. Contact center administrators have full access by default.

To configure non-deletion permissions, do this:

1. If you're configuring this at the application level: add a new recording settings group to the Annex/ Application options group for the IRWS_Cluster (or WS_Cluster, where applicable) application object, or update the existing recording group. For details, refer to [Genesys Administrator Extension Help: Configuration Manager](#) and [Installing Interaction Recording Web Services](#).

Important

You are not required to do it this way; you can also set this at the Person object level.

2. Configure one or both of the following options in the recording settings as follows:

```
[recording]
RECORDING_PERMISSION_APPLY_NON_DELETE = true
RECORDING_PERMISSION_UNAPPLY_NON_DELETE = true
```

The system applies permissions in the following order:

1. Default settings that are defined in the application code, which are overridden by:
2. Settings that are specified in the Application, which are overridden by:
3. Settings that are specified in the Person object that corresponds to the agent.

The following permissions are required to allow users with the role of supervisor or agent to be able to access and use the different non-deletion operations.

Permission	Description	Applies to	Checks against
RECORDING_PERMISSION_APPLY_NON_DELETE	Permission to protect a recording from being deleted	Apply Non-Deletion to a Recording	<ul style="list-style-type: none">• Supervisor• Agent
RECORDING_PERMISSION_UNAPPLY_NON_DELETE	Permission to remove deletion protection from a recording	Remove Non-Deletion from a Recording	<ul style="list-style-type: none">• Supervisor• Agent

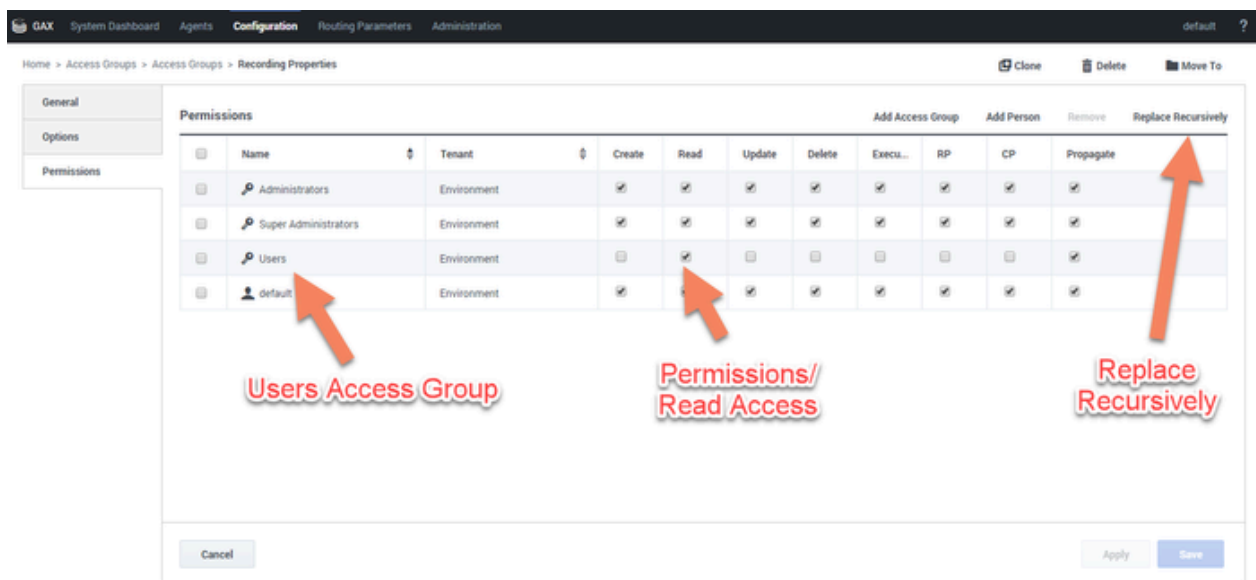
Configuring access control and agent hierarchy

Configuring access groups

By default, the Configuration Server has an **Access Group** called Users stored in the configuration database.

Install the Solution Deployment SPD file "Creation of base access groups" option to perform the following steps:

1. Create an **Access Group**, and set the permission to grant the users in the Access Group with Read access.
2. Add a new folder within Access Groups, called **Recording**, and set the permission to add the Users Access Group with Read access. Make sure the Replace Permissions Recursively action is set as shown in the following diagram:



3. Create the / (forward slash) Access Group within the **Recording** folder.

Important

If this **User Access Group** exists in more than one tenant, use unique naming conventions; otherwise, the users will not appear in the SpeechMiner UI.

Configuring partitions

For each partition used in the contact, create an **Access Group** object with the name of the partition within the **Recording** folder. For example, if there are three partitions— /sales, /support, and /marketing, create three **Access Group** objects named /sales, /support, and /marketing, respectively.

Important

Access Group names for partitions and agent hierarchy must be unique for each tenant.

Configuring agent hierarchy

Agent hierarchy and partitions are not required to record calls or access recordings; however, all agents must be assigned to the Users Access Group.

If agent hierarchy is required, assign the agent's hierarchy by configuring the `agent_hierarchy` option in the recording section of the Person object's Annex tab. For each hierarchy name, create a corresponding **Access Group** object within the **Recording** folder.

For the example above, create the following **Access Groups**:

- /
- /Anthony
- /Anthony/John
- /Anthony/Paul

Important

The `agent_hierarchy` field for a user should not include that user's name. For example, David's `agent_hierarchy` can be:

- /Genesys/Tel Aviv/
Not:
- /Genesys/Tel Aviv/David

Every user can only be part of one hierarchy (a single path) in the entire hierarchy tree. For example:

- If the hierarchy for David is /Genesys/Toronto, then John's hierarchy cannot be /Genesys/Tel Aviv/David. That is, David cannot be a part of two different hierarchies.
- /Genesys/Tel Aviv/BE and /Genesys/Toronto/BE should not exist in the same hierarchy tree. But, /Genesys/BE/Tel Aviv and /Genesys/BE/Toronto can exist in the same hierarchy tree.

Configuring user access control

Agents and users can be seen by a logged in user based on the logged in user's read permissions to the agents and users Person objects in the Configuration Server. Additionally, access to items within SpeechMiner (for example, Forms, Evaluations, Reports and so on), is also limited based on read permissions to the creator of those items.

To limit which agents and users can be seen by a logged in user you must set **AccessControlEnabled** to **1** (true) in the **ConfigServer** table in the SpeechMiner Database (that is, the database selected during the SpeechMiner installation).

Important

If **AccessControlEnabled** is not set to true, all users can see and access all agents and users items within SpeechMiner.

Configuring sensitive data privileges

Sensitive information (for example, credit card numbers, telephone numbers, home addresses and so on) can be hidden from agents when stored in the system.

To configure sensitive data privileges:

1. Add a new Recording settings group to the Annex/Application options group for the GIR cluster application object. For details, refer to ["Genesys Administrator Extension User Guide > Configuration Manager"](#)
2. Configure one or both of the following options in the Recording settings group created in step #1:
 - **metadata.privacy.agent_fields:** Add a comma-separated value of all the metadata fields that must be hidden if the user does not have permission to view the agent metadata fields. For example, callerPhoneNumber, dialedPhoneNumber, dnis, ani, agentId, username, phoneNumber, username, firstName, lastName, GSIP_RECORD, and so on.
 - **metadata.privacy.customer_fields:** Add a comma-separated value of all the metadata fields that must be hidden if the user does not have permission to view the customer metadata fields. For example, firstName, lastName, and so on.

Important

Metadata fields with angle brackets or backslashes are not supported.

With the following privileges you can view recording metadata fields that are usually masked from unauthorized users:

- **Customer Sensitive Data:** This privilege enables the user to display customer-sensitive data in the SpeechMiner GUI. When this privilege is enabled, the data is visible.
- **Agent Sensitive Data:** This privilege enables the user to display agent-sensitive data in the SpeechMiner GUI. When this privilege is enabled, the data is visible.

For more information on how to configure the above privileges, refer to [Configuring Roles and Privileges in GAX](#).

Important

- Both the Customer Sensitive Data privilege and the Agent Sensitive Data privilege will not affect report results. That is, sensitive data will be included in reports. If you do not want sensitive data to be included in reports you must disable the relevant report.

For more information about configuring Access Controls in Genesys Administrator Extension, see the [Genesys Administrator Extension User Guide](#).

Secure Transport Configuration

This section describes how to configure Transport Layer Security (TLS) for the Genesys Interaction Recording solution.

Server-Side Configuration

The following components must configure secure transports for HTTP.

Interaction Recording Web Services

Configuring TLS for Interaction Recording Web Services

See [Configuring TLS on the Server Side for Interaction Recording Web Services](#).

Configuring TLS for the Recording Processor Script

1. Configure HTTPS on the primary recording server. For more information, see the "Configure SSL" section of [Configuring Recording Processor Script](#).
 - a. For Windows, make sure the pyOpenSSL is installed. pyOpenSSL is already be installed on RHEL6.
 - b. Create a self-signed certificate and private key for the Recording Processor host. For example, on Ubuntu run:

```
openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem
```
 - c. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—Point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228.pem`.
 - d. Send the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section of the [GVP 8.5 User's Guide](#).
 - e. Restart Recording Processor.
2. Configure HTTPS on the backup recording server by following the same instructions as above using a new certificate and private key.

Configuring TLS for the Voice Processor

See [Voice Processor Service Level Configuration](#).

Configuring TLS for the Recording Crypto Server

See [Configure HTTP Port](#) tab in the [Configuring Recording Crypto Server](#) section.

Configuring TLS for the WebDAV Server

See [Configuring TLS for the WebDAV Server](#).

Configuring TLS for the Interaction Receiver and SpeechMiner UI Server

See [Enabling HTTPS for SpeechMiner](#).

Configuring TLS for the HTTP Load Balancer

See [Configuring TLS for the HTTP Load Balancer](#) in a single-tenant environment.
See [Configuring TLS for the HTTP Load Balancer](#) in a multi-tenant environment.

Client-Side Configuration

Configuring TLS for the Media Control Platform (MCP)

To add a Certificate Authority (CA):

1. Place the CA file on the MCP.
2. Using Genesys Administrator or Genesys Administrator Extension, in the **[fm]** section set the **ssl_ca_info** option to the location of the CA file.
3. Restart MCP.

To add client-side authentication:

1. Place the certificate file (PEM format) on the MCP.
2. Using Genesys Administrator or Genesys Administrator Extension, in the **[fm]** section set the **ssl_cert** option to the location of the certification file.
3. Restart MCP.

For more information about the MCP options, see the [Voice Platform Media Control Platform Configuration Options](#).

Configuring TLS for the IVR Profile

Using Genesys Administrator Extension, navigate to the Recording tab of the IVR Profile. Update the following addresses with the HTTPS locations:

- Storage Destination
- Recording Processor URI

- SpeechMiner Interaction Receiver
- SpeechMiner Destination for Analytics only

Configuring TLS for the Recording Processor Script

The Recording Processor Script creates three client connections, to:

- Interaction Recording Web Services (Web Services)
- SpeechMiner Interaction Receiver
- Backup Recording Processor Script

For details on configuring each connection, refer to the appropriate section at the [Configure SSL](#) link on the page [Deploying Recording Processor Script](#).

Configuring TLS for the Voice Processor

The Voice Processor creates three client connections, to:

- Interaction Recording Web Services (Web Services)
- SpeechMiner Interaction Receiver
- Genesys Info Mart

For details on configuring these connections, see [Configuring Voice Processor](#).

Configuring TLS for Interaction Recording Web Services

Interaction Recording Web Services (RWS) may be configured to use secure connections to the following components:

- Configuration Server
- SIP Server
- Interaction Server
- WebDAV
- Recording Crypto Server
- SpeechMiner Interaction Receiver
- Cassandra

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Security](#).

Configuring TLS for the Recording Muxer Script

The Recording Muxer Script creates client connections to the following:

- Interaction Recording Web Services
- Recording Crypto Server (if the recordings are encrypted)
- WebDAV

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Transport Layer Security \(TLS\) Connections](#).

Configuring TLS for the Recording Crypto Server

The Recording Crypto Server creates client connections to the following:

- Interaction Recording Web Services
- SpeechMiner Interaction Receiver
- Message Server
- Configuration Server

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Transport Layer Security \(TLS\) Connections](#).

Configuring TLS for the Recording Plug-in for GAX

See [Configuring Transport Layer Security](#).

Configuring Media Lifecycle Management

When it is time to purge old recording files, Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) requires additional configuration to allow the records to purge and/or backup successfully. For instance, when it is time to purge old recording files, Interaction Recording Web Services (Web Services) sends a purge request to the SpeechMiner database indicating which records to delete.

Interaction Recording Web Services (Web Services)

To enable Genesys Interaction Recording to purge and back up recording files, configure the Interaction Recording Web Services (Web Services) node as follows:

In the **backgroundScheduledMediaOperationsSettings** section of the **serverSettings** section within the **application.yaml** file:

- Set **enableBackgroundScheduledMediaOperations** to true
- Set **defaultBackupExportURI** to a backup folder—for example, `file:///tmp/archLocDefault` is the default backup folder.
- Set **enableScanAndScroll** to true to enable the Scan and Scroll feature of Elasticsearch. The default value is false.

Important

Verify that the SIP Server is running before accessing the MLM configuration. If the SIP Server is not running, a **User not authorized** message will appear when you try to access MLM in GAX.

For more information about these options, see the [Advanced Settings for the MLM API](#).

In the **recordingSettings** section of the **serverSettings** section within the **application.yaml** file, set **auditLogDeletedFiles** to true if you want to log all deleted recordings in the audit log when they are purged.

For more information about these options, see [Configuring the Call Recording Audit Log](#).

SpeechMiner

To enable Interaction Recording Web Services (Web Services) to contact Interaction Receiver and purge the requested recordings, use a text editor to add the following to the location based setting group in the `json.txt` file:

```
{
  "name": "interaction-receiver",
  "location": "/US/CA",
  "value": {
    "uri-prefix": "http://speechminer-server/interactionreceiver",
    "userName": "interaction receiver user name",
    "password": "interaction receiver password"
  }
}
```

Important

- <http://speechminer-server/interactionreceiver> is the Load Balancer URL that points to the Interaction Receiver.
- The Interaction Receiver user name and password must be the same as the **user** and **password** property values found within the **[recording.archive]** section of the tenant Annex in the configuration, and are set when configuring Recording Crypto Server. If these values are not found there, they should be added.

Execute the following command:

```
curl -u <user:password> -X POST -d @json.txt --header
"Content-Type: application/json" http://<Web Services-cluster-address>/api/v2/settings/
speechminer
```

Important

The username and password provided in the above command line must be associated with a user defined in the Genesys configuration environment.

For more information on the properties of this settings group, see [Interaction Recording Web Services Settings Groups](#).

For more information about the location based setting group for encryption, see [Encrypting and Provisioning Certificates](#).

Creating Rules and Schedules

Use Genesys Administrator Extension to create rules and schedules. For step-by-step instructions, see [Recording Lifecycle Scheduler](#).

Consider the following when creating backup and purge tasks:

- Do not schedule backup tasks to run concurrently on the same Interaction Recording Web Services (Web Services) node if these tasks back up overlapping records.

- Do not schedule backup and purge tasks to run concurrently if they act on overlapping records.
- Ensure that all the Interaction Recording Web Services (Web Services) nodes have accurate clocks.
- Genesys Administrator Extension's time is based on UTC.

Important

- When using the MLM Purge feature, if you specify a rule for voice recordings, the corresponding muxed screen recordings will not be purged unless you also select the **Include Screen Recordings** checkbox.
- Recordings that are protected from deletion (using the Non-Deletion API or SpeechMiner) will not be deleted by Media Lifecycle Management purge tasks.
- Do not schedule a purge task to run independently in its own rule unless you are willing to lose the associated data. Even if a backup has been scheduled, it is not guaranteed to complete successfully before the purge task is executed.
- Review the [Recording Lifecycle Scheduler Parameters](#) page in the Genesys Interaction Recording Help. It contains important details about the rule parameters, such as any limitations you should be aware of.

Warning

When you are scheduling rules containing purge tasks, adhere to the following guidelines to avoid an unexpected failure of Purge or Backup tasks:

- Run only *one* Purge task in a rule.
- Run the Purge task *last* in a rule.
- Do not run two rules with overlapping minAge/maxAge time intervals too close together (less than 5 seconds) if the first rule contains a Purge task. Note that the interval is the time between the rules that are running (that is, the completion of one rule and the start of the next) and not between the scheduled start time of rules.

You can look at the `recording.log` file to determine when a rule has finished. Look for the following message:

```
... [] [] [] Scheduled rule [<rule name>] at location [<node path>] completed
```

The `<rule name>` and `<node path>` depend on the customer configuration. Note that the amount of time to run a rule depends on many factors, including call volume. The interval should be much greater than that suggested above to make allowances for day to day variations.

Configuring For Multiple Regions

The following sections describe how to configure MLM for multiple regions.

Need For An MLM Node In Each Region Requiring Backup and/or Purge

By design, an MLM node will only backup and/or purge call and screen recordings for which the metadata region property exactly matches the `crRegion` (call recording region) property found in the node's Interaction Recording Web Services (Web Services) `application.yaml` configuration file (if you are using Web Services and Application version 8.5.201.09 or earlier it is found in the `server-settings.yaml` file). This design prevents these nodes from "pulling" media between data centers.

For example, if there are two data centers defining regions "east" and "west", and the client Interaction Recording Web Services (Web Services) nodes with `nodePaths` (in the `application.yaml` file or in the `server-settings.yaml` file if you are using Web Services and Application version 8.5.201.09 or earlier) `/US/EAST/10.2.0.1` through `/US/EAST/10.2.0.10` are in region "east", and client Interaction Recording Web Services (Web Services) nodes with `nodePaths` `/US/WEST/10.2.1.1` through `/US/WEST/10.2.1.10` are in region "west", and there is a requirement for deleting all call recordings after 90 days, then there will need to be at least one MLM node in each region (possibly with `nodePaths` `/US/EAST/10.2.0.20` and `/US/WEST/10.2.1.20`) each with a 90-day purge rule.

Configuring SpeechMiner Purge API

If a deployment supports call recording and SpeechMiner, a deployment will need to have the SpeechMiner Purge API configured (see [SpeechMiner](#) for more information).

For a multi-region deployment that has only one SpeechMiner, the SpeechMiner Purge API should be configured with a location property value that is the nearest common ancestor of the `nodePaths` of all the MLM nodes. For instance, using the example above, the nearest common ancestor of `nodePaths` `/US/EAST/10.2.0.20` and `/US/WEST/10.2.1.20` is `/US`.

For a multi-region deployment that has one SpeechMiner per region, the SpeechMiner Purge API should be configured for the SpeechMiner of each region, using a location property value for each that is the nearest common ancestor of all the `nodePaths` of the region's Interaction Recording Web Services (Web Services) nodes. For instance, using the example above, the nearest common ancestor of `nodePaths` `/US/EAST/10.2.0.1` through `/US/EAST/10.2.0.10` and `/US/EAST/10.2.0.20` is `/US/EAST`, and the nearest common ancestor of `nodePaths` `/US/WEST/10.2.1.1` through `/US/WEST/10.2.1.10` and `/US/WEST/10.2.1.20` is `/US/WEST`.

Configuring Pre-Recording

You can configure MLM to keep the entire audio and the screen of calls that might need review of a Contact Center supervisor or manager. Use the following steps to set up Pre-recording:

1. Using Genesys Administrator Extension (GAX), select **Business Attributes** and create a new **Custom** business attribute object. Name the object **Recording**.
2. In GAX, select **Business Attribute Values** and select the **Recording** object you created in step #1 above.

3. Select **Attribute Values**, and create a new attribute value named **Keep Recording**.
4. In the **Interaction-Workspace** section, create the following parameters:
 - display-type=enum
 - enum.business-attribute=enumkeep_recording
 - enum.default-value=no
 - ready-only=false
5. In GAX, select **Business Attributes** and create a **Customer** new business attribute object. Name the new object **enumkeep_recording**.
6. In GAX, select **Business Attribute Values** and select the **enumkeep_recording** object you created in the step above.
7. Select **Attribute Values** and create the following attribute values:
 - no (set to default)
 - yes
8. In the **Workspace Web Edition Cluster** object (WWEWS_Cluster) or from the **Workspace Desktop Edition** application object, select the **Application Options** tab.
9. In the **[interaction-workspace]** section, set the **interaction.case-data.format-business-attribute** option to **Recording**.
10. From **Routing Strategy**, attach the following user data to the interaction: keep_recording="no" .. For additional information, refer to the [Universal Routing 8.1 Reference Manual](#).

Selective Recording

If your business retention policy is to keep a random percentage, say 20% of calls, then the routing strategy would call a function to determine whether to keep the call. If the call should not be kept, set the value to **keep_recording="no"**. If the call should be kept based on the rule, set the value to **keep_recording="yes"**. The agent does not need to mark the call to be kept for review.

Use the following steps to setup Pre-recording for selective recording:

1. In the routing strategy, attach the following user data to the call:
 - keep_recording="no"
2. Add the same user data on the Agent's desktop, so that the agent can change the value from no to yes if the agent wants to keep the recording based on what the caller said.

For more information about how to create rules and schedules, see [Recording Lifecycle Scheduler](#).

Upgrading the GIR Components

When upgrading from version 8.5.205.01 or earlier to version 8.5.206.01 or later, the GIR components can be upgraded in any order (Web Services, Recording Plug-in, Recording Processor

Script or Voice Processor, etc), but `callType` should not be specified in MLM tasks within the upgraded Recording plug-in until all Web Services nodes have been upgraded.

Rolling Back the GIR Components

When rolling back the components from version 8.5.206.01 or later to version 8.5.205.01 or earlier, the GIR components can be rolled back in any order (Web Services, Recording Plug-in, Recording Processor Script or Voice Processor, etc), as long as no MLM tasks specify `callType` in the filter. If a `callType` is specified as a filter of a task, the task must be removed before rolling back Web Services to a previous version. Disabling the task is not sufficient.

Advanced Settings for the MLM API

The following table describes the parameters that are in the **backgroundScheduledMediaOperationsSettings** section of the `serverSettings` section within the **application.yaml** file.

Parameter Name	Description	Default Value
<code>enableBackgroundScheduledMediaOperations</code>	Specifies whether to turn on the MLM feature.	false
<code>schedulerThreads</code>	Specifies the maximum number of enabled MLM scheduled rules that can run concurrently (that is, they start and run at the same time everyday). Valid range of values: 1 - 64. Note: If you set this value too low, the scheduled rules will run sequentially as necessary to save the number of threads opened, resulting in scheduled rules running some time after the time they are scheduled.	4
<code>schedulePollingInterval</code>	Specifies, in seconds, the time to poll for gir-scheduler settings and synchronize the rule schedule.	60
<code>speechMinerMaxConnection</code>	Specifies the maximum number of concurrent TCP connections when issuing API request to the SpeechMiner.	20
<code>speechMinerMaxTotalConnection</code>	Specifies the size of the connection pool for issuing API requests to the SpeechMiner. If set to a value less than 1, the value is automatically set to speechMinerMaxConnection *.	-1

Parameter Name	Description	Default Value
	10.	
speechMinerSocketTimeout	Specifies the time, in milliseconds, to wait for the SpeechMiner API response before timing out.	15000
defaultBackupExportURI	Specifies the file path (file:// URI) to the backup the export directory to if gir-scheduler rule setting does not specify the backup location.	Empty
enableScanAndScroll	Specifies whether to turn on the feature where MLM uses Elasticsearch scan and scroll queries to determine the recording IDs on which to act.	false
scanIntervalsPerDay	<p>When MLM is configured to use Elasticsearch scan and scroll queries to determine the recording IDs on which to act, this parameter determines the number of scan intervals used in a day of recordings. Reduce this value to reduce the number of Elasticsearch scan queries performed by an MLM Task for its work, assuming that all other things remain equal. Reducing this value also increases the lifetime of the search context created by each Elasticsearch scan query, which in turn increases the number of open file descriptors in use by Elasticsearch.</p> <p>Note: When configuring, ensure that the number of minutes in a day (i.e. 24 * 60) is exactly divisible by the configured value. If this condition is not met, RWS replaces the value with the default value if the specified value is less than or equal to zero, the next lowest integer from the specified value that is valid if the specified value is greater than zero, or 1440 if the specified value is greater than 1440.</p>	24

Creating Folder Hierarchy for Recording Storage

WebDAV performance degrades over time and the file system becomes inoperable if all your recording files are saved in a single directory. If you are using Apache HTTP Server for your WebDAV server, Genesys recommends using the following example procedures to create a folder hierarchy.

Important

Genesys recommends that you create the storage directories before your contact center goes into production. Keep in mind, that if you use the following procedure to create the directories, you will need to extend the directories at the end of 2023.

Prerequisites

Before you can create recording storage with folder hierarchy, you must have the following items in place:

- The SIP Server's **recording-filename** option must be set to \$UUID\$_\$DATE\$_\$TIME\$
- The IVR Profile Recording Storage must be set to `http://<WebDAV server>/recordings`

Important

These instructions assume that `/recordings` is the base path of the WebDAV URI. If you use something else as the base path, change it accordingly in the rewrite rule.

- The IVR Profile **Recording Filename Template** option must be set to `id`

Important

The filename must be prefixed with `id`, to allow for additional parameters after this.

- For screen recording, these instructions assume that the storage path has a base path of screens. If you use something else as the base path, change it accordingly in the rewrite rule.

Create and Use New Hierarchy

For Linux

1. Create a file and copy and paste the content below into the file (for example, "createfolders.sh"). This will create many directories in yyyy/mm/dd/hh format. The example provided creates directories to the end of 2023.

```
#!/bin/sh
for year in 2019 2020 2021 2022 2023
do
for month in 01 02 03 04 05 06 07 08 09 10 11 12
do
for day in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31
do
for hour in 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
do
echo "mkdir -p $year/$month/$day/$hour"
mkdir -p $year/$month/$day/$hour
done
done
done
done
```

After saving the file, ensure the script has read and execute privileges and then run the script from /mnt/recordings.

2. Make sure the following Apache modules are loaded in the /etc/httpd/conf/httpd.conf file.
 - LoadModule rewrite_module modules/mod_rewrite.so
 - LoadModule proxy_module modules/mod_proxy.so
3. Depending on the Apache version, add the following rewrite rules in the /etc/httpd/conf/httpd.conf file to parse the file name into /recordings/{yyyy}/{mm}/{dd}/{hh}/{filename}:
 - Apache version 2.2 and lower

```
RewriteLog "logs/rewrite.log"
RewriteLogLevel 1
RewriteEngine on
RewriteRule ^/recordings/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*)
/recordings/$2/$3/$4/$5/$1 [P,L]
RewriteRule ^/screens/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*) /screens/$2/$3/$4/$5/$1
[P,L]
RewriteRule ^/screens/([A-Za-z0-9]*_(\d{4})_(\d\d)_(\d\d)_(\d\d).*)
/screens/$2/$3/$4/$5/$1 [P,L]
```

Tip

Test this rewrite rule in your environment to ensure that all the files are placed in the correct sub-directories.

- Apache version 2.4 and higher

```
LogLevel debug rewrite:trace1
RewriteEngine on
```

```
RewriteRule ^/recordings/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*)  
/recordings/$2/$3/$4/$5/$1 [P,L]  
RewriteRule ^/screens/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*) /screens/$2/$3/$4/$5/$1  
[P,L]  
RewriteRule ^/screens/([A-Za-z0-9]*_(\d{4})_(\d\d)_(\d\d)_(\d\d).*)  
/screens/$2/$3/$4/$5/$1 [P,L]
```

Important

For Apache version 2.4 and higher, rewrite logs are not written to a separate file (as in Apache 2.2) but to error_log file. To get the mod_rewrite specific log messages, you can use the following command:

```
tail -f error_log | fgrep '[rewrite:]'
```

Setting up the Load Balancer in a Single-Tenant Environment

See Also: [Setting Up the Load Balancer in a Multi-Tenant Environment](#)

Important

- The load balancer used for RWS must be configured with sufficient capacity to accommodate one persistent connection from each logged in agent with SR Service in addition to other RWS requests.
- Currently, Genesys does not provide instructions on how to set up load balancer for the Voice Processor. You can configure your own load balancing solution for multiple Voice Processor instances, if required.
- The architecture for load balancer cluster is supported on Red Hat Enterprise Linux 6 for HTTPD 2.2 only.

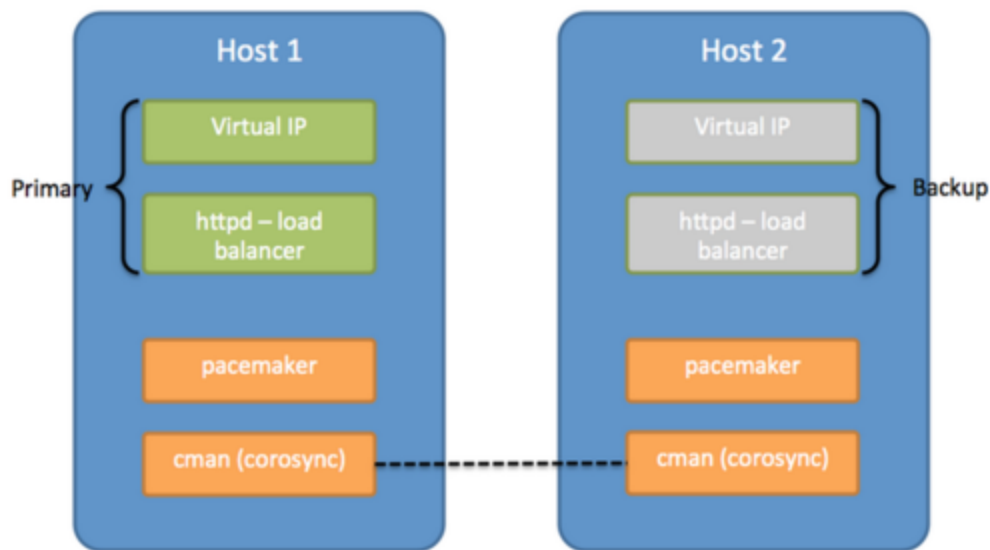
Red Hat Enterprise Linux 6 for HTTPD 2.2

Overview and Architecture

The solution uses a common Linux HA framework from <http://clusterlabs.org>. There are two components involved in this solution:

- Cman uses corosync internally to provide a platform for membership, messaging, and quorum among the hosts.
- Pacemaker is a cluster resource manager that controls where resources (processes) are executed. Pacemaker works with the processes like Apache httpd using resource agents to provide controls of the process such as start/stop/status.

The following diagram shows a primary/backup design to associate a single virtual IP address with httpd. Whenever the primary host fails, the virtual IP address and the httpd process can be automatically fail over to the backup host.



As a simple two host primary/backup solution, the hosts must be deployed on the same subnet that allows UDP multicast. This solution provides the same reliability as a network that hosts the two machines handling the virtual IP address.

Deploying the Load Balancer

Important

For load balancers used for Recording Processors, warm standby functionality must be disabled.

Prerequisites

- Red Hat Enterprise Linux 6 with the High Availability Add-On, for HTTPD 2.2

Tip

Network Manager can be enabled as part of the OS installation. To disable Network Manager, see [Red Hat documentation](#).

Installing the OS

Install the required software using the following command:


```
yum -y install httpd pacemaker cman pcs ccs resource-agents
```

Setting up the HTTP Load Balancer

Setting up the HTTP Load Balancer (when working with 8.5.210.02 or earlier)

On both servers, create the following files:

- Create `/etc/httpd/conf.d/serverstatus.conf`, and add the following text:

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

- Create `/etc/httpd/conf.d/loadbalancer.conf`, and add the following text:

Important

If your existing configuration already includes the load balancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the _ (underscore) characters. For example, if the application name is RCS 1, set the **route** value to RCS_1.

```
# Web Services
<Proxy balancer://htcc>
BalancerMember http://htcc1:8080 route=HTCC1
BalancerMember http://htcc2:8080 route=HTCC2
BalancerMember http://htcc3:8080 route=HTCC3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/" env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /api balancer://htcc/api
ProxyPass /internal-api balancer://htcc/internal-api
ProxyPass /ui balancer://htcc/ui

# RP
<Proxy balancer://rp>
BalancerMember http://rp1:8889
BalancerMember http://rp2:8889
</Proxy>
ProxyPass /rp/api balancer://rp/api

# RCS
<Proxy balancer://rcs>
BalancerMember http://rcs1:8008 disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
```

```

BalancerMember http://rcs2:8008 disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickySession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcs/rcs

# Interaction Receiver
<Proxy balancer://sm>
BalancerMember http://ir1
BalancerMember http://ir2
</Proxy>
ProxyPass /interactionreceiver balancer://sm/interactionreceiver

# WebDAV
<Proxy balancer://webdav>
BalancerMember http://webdav1
BalancerMember http://webdav2 status=H
</Proxy>
ProxyPass /recordings balancer://webdav/recordings
ProxyPass /dest2 balancer://webdav/dest2

```

Setting up the HTTP Load Balancer (when working with a version later than 8.5.210.02 with Workspace Desktop Edition)

On both servers, create the following files:

- Create `/etc/httpd/conf.d/serverstatus.conf` and add the following text:

```

<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>

```

- Create `/etc/httpd/conf.d/loadbalancer.conf`, and add the following text:

Important

If your existing configuration already includes the loadbalancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the _ (underscore) characters. For example, if the application name is RCS 1, set the **route** value to RCS_1.

```

# Interaction Recording Web Services
<Proxy balancer://rws>
BalancerMember http://rws1:8080 route=RWS1
BalancerMember http://rws2:8080 route=RWS2

```

```

BalancerMember http://rws3:8080 route=RWS3
Header add Set-Cookie "ROUTEID=.{BALANCER_WORKER_ROUTE}e; path=/" env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /api balancer://rws/api
ProxyPass /internal-api balancer://rws/internal-api

# RP
<Proxy balancer://rp>
BalancerMember http://rp1:8889
BalancerMember http://rp2:8889
</Proxy>
ProxyPass /rp/api balancer://rp/api

# RCS
<Proxy balancer://rcs>
BalancerMember http://rcs1:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://rcs2:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcs/rcs

# Interaction Receiver
<Proxy balancer://sm>
BalancerMember http://ir1
BalancerMember http://ir2
</Proxy>
ProxyPass /interactionreceiver balancer://sm/interactionreceiver

# WebDAV
<Proxy balancer://webdav>
BalancerMember http://webdav1
BalancerMember http://webdav2 status=H
</Proxy>
ProxyPass /recordings balancer://webdav/recordings
ProxyPass /dest2 balancer://webdav/dest2

```

Setting up the HTTP Load Balancer (when working with a version later than 8.5.210.02 with Workspace Web Edition)

On both servers, create the following files:

- Create `/etc/httpd/conf.d/serverstatus.conf`, and add the following text:

```

<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>

```

- Create `/etc/httpd/conf.d/loadbalancer.conf`, and add the following text:

Important

If your existing configuration already includes the load balancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Web Services, Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the **_** (underscore) characters. For example, if the application name is RCS 1, set the **route** value to **RCS_1**.

```
# Web Services
<Proxy balancer://htcc>
BalancerMember http://htcc1:8080 route=HTCC1
BalancerMember http://htcc2:8080 route=HTCC2
BalancerMember http://htcc3:8080 route=HTCC3
Header add Set-Cookie "GWSROUTEID=%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=GWSROUTEID
</Proxy>
ProxyPass /api balancer://htcc/api
ProxyPass /internal-api balancer://htcc/internal-api
ProxyPass /ui balancer://htcc/ui

# Interaction Recording Web Services
<Proxy balancer://rws>
BalancerMember http://rws1:8080 route=RWS1
BalancerMember http://rws2:8080 route=RWS2
BalancerMember http://rws3:8080 route=RWS3
Header add Set-Cookie "RWSROUTEID=%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=RWSROUTEID
</Proxy>
ProxyPass /gir/api balancer://rws/api
ProxyPass /gir/internal-api balancer://rws/internal-api

# RP
<Proxy balancer://rp>
BalancerMember http://rp1:8889
BalancerMember http://rp2:8889
</Proxy>
ProxyPass /rp/api balancer://rp/api

# RCS
<Proxy balancer://rcs>
BalancerMember http://rcs1:8008 disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://rcs2:8008 disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcs/rcs

# Interaction Receiver
<Proxy balancer://sm>
BalancerMember http://ir1
BalancerMember http://ir2
</Proxy>
```

```
ProxyPass /interactionreceiver balancer://sm/interactionreceiver
```

```
# WebDAV
<Proxy balancer://webdav>
BalancerMember http://webdav1
BalancerMember http://webdav2 status=H
</Proxy>
ProxyPass /recordings balancer://webdav/recordings
ProxyPass /dest2 balancer://webdav/dest2
```

Interaction Recording Web Services

1. Set the following properties in the **application.yaml** file.

```
sessionCookieName: GIRJSESSIONID
externalApiUrlV2: [ToBeChanged: public URL including protocol, address and port,
<PUBLIC_SCHEMA_BASE_URL>]/gir/api/v2
  internalApiUrlV2: [ToBeChanged: internal URL including protocol, address and port,
<INTERNAL_SCHEMA_BASE_URL>]/gir/internal-api
  undocumentedExternalApiUrl: [ToBeChanged: public URL including protocol, address and
port, <PUBLIC_SCHEMA_BASE_URL>]/gir/internal-api
```

2. Verify that the following URLs are routed to **<loadbalancer>/gir**:

```
externalApiUrlV2
internalApiUrlV2
undocumentedExternalApiUrl
```

Screen Recording Service

- Verify that the **config.json** file on the agent desktop does not include the **server** entry.
- Verify that **<load balancer> address/hostname** appears in the **allowedHosts** entry.
- Using Genesys Administrator Extension, add the following parameter to the **[interaction-workspace]** section of the Web Services Cluster object:

```
screen-recording.client.server-url: https://<load balancer>:443/gir
```

Recording Processor

Configure the **base_uri** parameter in the **[htcc]** section of the **rpconfig.cfg** configuration file for each Recording Processor instance to point to **<load balancer>/gir**.

Recording Crypto Server

Use Genesys Administrator Extension to configure the **baseurl** parameter in the **[htcc]** section of the Recording Crypto Server application to point to **<load balancer>/gir**.

Recording GAX plug-in

Use Genesys Administrator Extension to:

- Configure the **baseurl** parameter in RCS (see above).
 - In the GAX application object, override the **htcc_base_url** option in the **[rcs]** section.
-

Recording Muxer Script

Use Genesys Administrator Extension to configure the **base_uri** parameter in the **[htcc]** section of the Recording Muxer application, to point to **<load balancer>/gir**.

Important

The password parameter in the **[htcc]** section should be updated with a strong password during premise deployment.

For additional details, refer to [Interaction Recording Options Reference](#).

SpeechMiner

In the **SpeechMiner Configuration Tool**, in the **Recording** page, enter **<load balancer>/gir** in the **HTCC URL** field. For additional details, refer to [Configuring SpeechMiner](#).

Configuring TLS for the HTTP Load Balancer

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: /etc/pki/tls/certs/localhost.crt
- Key: /etc/pki/tls/private/localhost.key

2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **/etc/httpd/conf.d/ssl.conf** file and modify the following lines:

- SSLCertificateFile /etc/pki/tls/certs/localhost.crt
- SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

3. To enable https for the proxy, edit the **/etc/httpd/conf.d/ssl.conf** file and add the following option:
SSLProxyEngine on

4. Direct the load balancer to the proper https locations. For example:

```
<Proxy balancer://rws>
BalancerMember https://rws1:8080 route=RWS1
BalancerMember https://rws2:8080 route=RWS2
BalancerMember https://rws3:8080 route=RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /api balancer://rws/api
ProxyPass /internal-api balancer://rws/internal-api
```

Setting Up Pacemaker and Cman

Important

Perform the following commands using the **root** user.

Disable Autostart for Httpd

Pacemaker manages the startup of httpd. Disable httpd from chkconfig services using the following command:

```
chkconfig httpd off
```

Setting Up the Hosts File

Make sure there is a hostname for both servers and that the hostname is resolvable on both hosts, either using DNS or /etc/hosts file. ip1 and ip2 are used as the hostnames thereafter.

```
# /etc/hosts
# ... keep the existing lines, and only append new lines below
192.168.33.18 ip1
192.168.33.19 ip2
```

Setting Up the Cluster

Run the following command on each host to create the cluster configuration:

```
ccs -f /etc/cluster/cluster.conf --createcluster webcluster
ccs -f /etc/cluster/cluster.conf --addnode ip1
ccs -f /etc/cluster/cluster.conf --addnode ip2
ccs -f /etc/cluster/cluster.conf --addfencedev pcmk agent=fence_pcmk
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect ip1
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect ip2
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk ip1 pcmk-redirect port=ip1
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk ip2 pcmk-redirect port=ip2
ccs -f /etc/cluster/cluster.conf --setcman two_node=1 expected_votes=1
echo "CMAN_QUORUM_TIMEOUT=0" >> /etc/sysconfig/cman
```

Start the Service

Start the cman and pacemaker services on each host using the following command:

```
service cman start
service pacemaker start
chkconfig --level 345 cman on
chkconfig --level 345 pacemaker on
```

(Optional) Setting Up UDP Unicast

This solution relies on UDP multicast to work, but can also work with UDP unicast. Edit the **/etc/**

cluster/cluster.conf file and insert an attribute to the <cman> tag as follows:

```
...  
<cman transport="udpu" two_node="1" expected_votes="1"/>  
...
```

Restart both servers for the changes to take effect.

Setting Cluster Defaults

Run the following on one of the servers.

```
pcs property set stonith-enabled=false  
pcs property set no-quorum-policy=ignore  
pcs resource defaults migration-threshold=1
```

Configure the Virtual IP Address and Apache httpd

Run the following on one of the servers.

For the first command below, `nic=eth0` refers to the network interface that brings up the virtual IP address. Change `eth0` to the active network interface your environment uses.

Change <Virtual IP> in the first command below to your virtual IP assigned to this load balancer pair.

```
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=<Virtual IP> nic=eth0 cidr_netmask=32  
op monitor interval=30s  
pcs resource create webserver ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf  
statusurl="http://localhost/server-status" op monitor interval=30s  
pcs resource meta webserver migration-threshold=10  
pcs constraint colocation add webserver virtual_ip INFINITY  
pcs constraint order virtual_ip then webserver
```

Maintaining Pacemaker

The following commands help you with the maintenance operations for pacemaker.

To check the status of the cluster:

```
pcs status
```

To clear resource errors (for example, because of incorrect configuration):

```
pcs resource cleanup <resourcename>
```

A resource name is either `virtual_ip` or web server (for example, `pcs resource cleanup webserver`).

To check the status of the resources in the cluster:

```
crm_mon -o -1
```


Red Hat Enterprise Linux 8 for HTTPD 2.4

Deploying the Load Balancer

Important

For load balancers used for Recording Processors, warm standby functionality must be disabled.

Prerequisites

- Red Hat Enterprise Linux 8 with the High Availability Add-On, for HTTPD 2.4

Tip

Network Manager can be enabled as part of the OS installation. To disable Network Manager, see [Red Hat documentation](#).

Installing the OS

Install the required software using the following command:

```
yum -y install httpd
```

Setting up the HTTP Load Balancer

Setting up the HTTP Load Balancer (when working with 8.5.210.02 or earlier)

On both servers, create the following files:

- Create `/etc/httpd/conf.d/serverstatus.conf`, and add the following text:

```
<Location /server-status>  
  SetHandler server-status  
  Order deny,allow  
  Deny from all  
  Allow from 127.0.0.1  
</Location>
```

- Create `/etc/httpd/conf.d/loadbalancer.conf`, and add the following text:

Important

If your existing configuration already includes the load balancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the _ (underscore) characters. For example, if the application name is RCS 1, set the **route** value to RCS_1.

```
# Web Services
<Proxy balancer://htcc>
BalancerMember http://htcc1:8080 route=HTCC1
BalancerMember http://htcc2:8080 route=HTCC2
BalancerMember http://htcc3:8080 route=HTCC3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/" env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /api balancer://htcc/api
ProxyPass /internal-api balancer://htcc/internal-api
ProxyPass /ui balancer://htcc/ui

# RP
<Proxy balancer://rp>
BalancerMember http://rp1:8889
BalancerMember http://rp2:8889
</Proxy>
ProxyPass /rp/api balancer://rp/api

# RCS
<Proxy balancer://rcs>
BalancerMember http://rcs1:8008 disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://rcs2:8008 disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcs/rcs

# Interaction Receiver
<Proxy balancer://sm>
BalancerMember http://ir1
BalancerMember http://ir2
</Proxy>
ProxyPass /interactionreceiver balancer://sm/interactionreceiver

# WebDAV
<Proxy balancer://webdav>
BalancerMember http://webdav1
BalancerMember http://webdav2 status=H
</Proxy>
ProxyPass /recordings balancer://webdav/recordings
ProxyPass /dest2 balancer://webdav/dest2
```

Setting up the HTTP Load Balancer (when working with a version later than 8.5.210.02 with

Workspace Desktop Edition)

On both servers, create the following files:

- Create `/etc/httpd/conf.d/serverstatus.conf` and add the following text:

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

- Create `/etc/httpd/conf.d/loadbalancer.conf`, and add the following text:

Important

If your existing configuration already includes the loadbalancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the _ (underscore) characters. For example, if the application name is RCS 1, set the **route** value to RCS_1.

```
# Interaction Recording Web Services
<Proxy balancer://rws>
BalancerMember http://rws1:8080 route=RWS1
BalancerMember http://rws2:8080 route=RWS2
BalancerMember http://rws3:8080 route=RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/" env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /api balancer://rws/api
ProxyPass /internal-api balancer://rws/internal-api

# RP
<Proxy balancer://rp>
BalancerMember http://rp1:8889
BalancerMember http://rp2:8889
</Proxy>
ProxyPass /rp/api balancer://rp/api

# RCS
<Proxy balancer://rws>
BalancerMember http://rws1:8080 disablereuse=0n connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://rws2:8080 disablereuse=0n connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
```

```

ProxyPass /rcs balancer://rcs/rcs

# Interaction Receiver
<Proxy balancer://sm>
BalancerMember http://ir1
BalancerMember http://ir2
</Proxy>
ProxyPass /interactionreceiver balancer://sm/interactionreceiver

# WebDAV
<Proxy balancer://webdav>
BalancerMember http://webdav1
BalancerMember http://webdav2 status=H
</Proxy>
ProxyPass /recordings balancer://webdav/recordings
ProxyPass /dest2 balancer://webdav/dest2

```

Setting up the HTTP Load Balancer (when working with a version later than 8.5.210.02 with Workspace Web Edition)

On both servers, create the following files:

- Create `/etc/httpd/conf.d/serverstatus.conf`, and add the following text:

```

<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>

```

- Create `/etc/httpd/conf.d/loadbalancer.conf`, and add the following text:

Important

If your existing configuration already includes the load balancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Web Services, Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the _ (underscore) characters. For example, if the application name is RCS 1, set the **route** value to RCS_1.

```

# Web Services
<Proxy balancer://htcc>
BalancerMember http://htcc1:8080 route=HTCC1
BalancerMember http://htcc2:8080 route=HTCC2
BalancerMember http://htcc3:8080 route=HTCC3
Header add Set-Cookie "GWSROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
ProxySet stickySession=GWSROUTEID
</Proxy>

```

```

ProxyPass /api balancer://htcc/api
ProxyPass /internal-api balancer://htcc/internal-api
ProxyPass /ui balancer://htcc/ui

# Interaction Recording Web Services
<Proxy balancer://rws>
BalancerMember http://rws1:8080 route=RWS1
BalancerMember http://rws2:8080 route=RWS2
BalancerMember http://rws3:8080 route=RWS3
Header add Set-Cookie "RWSROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
ProxySet stickySession=RWSROUTEID
</Proxy>
ProxyPass /gir/api balancer://rws/api
ProxyPass /gir/internal-api balancer://rws/internal-api

# RP
<Proxy balancer://rp>
BalancerMember http://rp1:8889
BalancerMember http://rp2:8889
</Proxy>
ProxyPass /rp/api balancer://rp/api

# RCS
<Proxy balancer://rcs>
BalancerMember http://rcs1:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://rcs2:8008 disablereuse=0n connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickySession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcs/rcs

# Interaction Receiver
<Proxy balancer://sm>
BalancerMember http://ir1
BalancerMember http://ir2
</Proxy>
ProxyPass /interactionreceiver balancer://sm/interactionreceiver

# WebDAV
<Proxy balancer://webdav>
BalancerMember http://webdav1
BalancerMember http://webdav2 status=H
</Proxy>
ProxyPass /recordings balancer://webdav/recordings
ProxyPass /dest2 balancer://webdav/dest2

```

Interaction Recording Web Services

1. Set the following properties in the **application.yaml** file.

```

sessionCookieName: GIRJSESSIONID
externalApiUrlV2: [ToBeChanged: public URL including protocol, address and port,
<PUBLIC_SCHEMA_BASE_URL>]/gir/api/v2
  internalApiUrlV2: [ToBeChanged: internal URL including protocol, address and port,
<INTERNAL_SCHEMA_BASE_URL>]/gir/internal-api
  undocumentedExternalApiUrl: [ToBeChanged: public URL including protocol, address and
port, <PUBLIC_SCHEMA_BASE_URL>]/gir/internal-api

```

2. Verify that the following URLs are routed to **<loadbalancer>/gir**:

```
externalApiUrlV2  
internalApiUrlV2  
undocumentedExternalApiUrl
```

Screen Recording Service

- Verify that the **config.json** file on the agent desktop does not include the **server** entry.
- Verify that **<load balancer> address/hostname** appears in the **allowedHosts** entry.
- Using Genesys Administrator Extension, add the following parameter to the **[interaction-workspace]** section of the Web Services Cluster object:

```
screen-recording.client.server-url: https://<load balancer>:443/gir
```

Recording Processor

Configure the **base_uri** parameter in the **[htcc]** section of the **rpconfig.cfg** configuration file for each Recording Processor instance to point to **<load balancer>/gir**.

Recording Crypto Server

Use Genesys Administrator Extension to configure the **baseurl** parameter in the **[htcc]** section of the Recording Crypto Server application to point to **<load balancer>/gir**.

Recording GAX plug-in

Use Genesys Administrator Extension to:

- Configure the **baseurl** parameter in RCS (see above).
- In the GAX application object, override the **htcc_base_url** option in the **[rcs]** section.

Recording Muxer Script

Use Genesys Administrator Extension to configure the **base_uri** parameter in the **[htcc]** section of the Recording Muxer application, to point to **<load balancer>/gir**.

Important

The password parameter in the **[htcc]** section should be updated with a strong password during premise deployment.

For additional details, refer to [Interaction Recording Options Reference](#).

SpeechMiner

In the **SpeechMiner Configuration Tool**, in the **Recording** page, enter **<load balancer>/gir** in the **HTCC URL** field. For additional details, refer to [Configuring SpeechMiner](#).

Configuring TLS for the HTTP Load Balancer

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: /etc/pki/tls/certs/localhost.crt
- Key: /etc/pki/tls/private/localhost.key

2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **/etc/httpd/conf.d/ssl.conf** file and modify the following lines:

- SSLCertificateFile /etc/pki/tls/certs/localhost.crt
- SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

3. To enable https for the proxy, edit the **/etc/httpd/conf.d/ssl.conf** file and add the following option:
SSLProxyEngine on

4. Direct the load balancer to the proper https locations. For example:

```
<Proxy balancer://rws>
BalancerMember https://rws1:8080 route=RWS1
BalancerMember https://rws2:8080 route=RWS2
BalancerMember https://rws3:8080 route=RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /api balancer://rws/api
ProxyPass /internal-api balancer://rws/internal-api
```

Additional Feature Configuration

Audio Tones

The following section outlines the general configuration for audio tones.

Media Server

The following table describes the options required for audio tones when using Media Server:

Section Name	Parameter Name	Description
Conference	record_recorddnhearstone	Specifies whether the RecordDN (Party A) hears the repeating tone.
Conference	record_otherdnhearstone	Specifies whether the OtherDN (Party B) hears the repeating tone.

Media Server allows you to configure whether the recording also gets the audio tone. When the audio tone is injected into the call, Media Server distinguishes between what the participant hears and what the participant says. The above two configuration parameters affect what the participant hears.

Section Name	Parameter Name	Description
Conference	record_chan2source	<p>Specifies the recorded media that represents the first participant (Record DN) in the recording session.</p> <ul style="list-style-type: none">recorddnsaysotherdnhears <p>If the Other DN is configured to receive consent and you want the consent to be recorded, set the value to otherdnhears.</p>
Conference	record_otherdnhearstone	<p>Specifies the recorded media that represents the second participant (Other DN) in the recording session.</p> <ul style="list-style-type: none">otherdnsays

Section Name	Parameter Name	Description
		<ul style="list-style-type: none">recorddnhears <p>If the Record DN is configured to receive consent and you want the consent to be recorded, set the value to recorddnhears.</p>

Enable Call Recording

Call recording can be enabled through three methods:

1. **Full-time recording or Total recording**—A specific DN is configured to enable recording for all calls for the specific DN.
2. **Selective Recording**—Record a party in the call is determined at a route point and the recording starts as soon as the call is established.
3. **Dynamic Recording**—Start/stop/pause/resume a recording call can be requested by an agent at any time after the call is established using Interaction Workspace.

Once a recording has started, there are two conditions where the recording stops:

1. When the party being recorded leaves the call, or when the customer drops the call. For example, when the recording applies to the agent in the call and the call is transferred to a second agent. The recording is stopped when the agent leaves the call. Note that the second agent can have recording enabled and the same call gets recorded with a second call recording segment.
2. When dynamic recording control requests the recording to be stopped.

Important

If using Workspace Desktop Edition for the agent desktop, the agent can hide the status of the recording. This functionality can be enabled through Workspace role configuration. For more information, see the [Setting Up Agents on the System](#) in the Workspace Desktop Edition documentation.

Genesys Interaction Recording Options Reference

Recording Processor Script

The following describes the options for the Recording Processor Script application.

[+] config_server Section

application_name

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the application name of the Recording Processor Script.

hostname

Default Value: <ip address>

Valid Values: Any string

Changes Take Effect: After restart

Specifies the IP address of the Configuration Server host.

port

Default Value: 2020

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the port of the Configuration Server host.

username

Default Value: default

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to connect to the Configuration Server.

password

Default Value: password

Valid Values: Any string
Changes Take Effect: After restart

Specifies the password used to connect to the Configuration Server.

`backup_host`

Default Value: Empty
Valid Values: Any string
Changes Take Effect: After restart

Specifies the IP address of the backup Configuration Server.

`backup_port`

Default Value: Empty
Valid Values: Any integer
Changes Take Effect: After restart

Specifies the port for the backup Configuration Server.

`config_server_encoding`

Default Value: Empty
Valid Values: UTF-8
Changes Take Effect: After restart

Specifies whether to enable decoding of the multibyte usernames from Configuration Server. This parameter is optional. If left empty, RPS might post the multibyte usernames incorrectly, resulting in two usernames being displayed for calls in SpeechMiner. You need to enable this option in the following circumstance:

- Configuration Server is configured for multiple languages (UTF-8).
- Microsoft SQL Server is used for the Genesys Interaction Recording (GIR) ICON database.
- The agent username contains characters encoded as multiple bytes in UTF-8.

[+] processing Section

`check_agent_prev_state`

Default Value: 0
Valid Values: 0 (no), 1 (yes)
Changes Take Effect: After restart

Specifies whether or not Recording Processor Script will wait for the agent to complete After Call Work (ACW) and to no longer be in the ACW state before retrieving the ACW data from the ICON Database.

Note: Enabling this setting can cause delays in processing calls with ACW data.

post_acw_delay

Default Value: 0

Valid Values: 0, or any positive integer

Changes Take Effect: After restart

The minimum amount of time (in seconds) that Recording Processor Script will wait for After Call Work (ACW) data to be processed before retrieving ACW data from the ICON database. If this setting is enabled, it causes RPS to wait (one or more times) for a duration that is approximately equal to **process_wait_seconds** until **post_acw_delay** seconds have passed before collecting ACW data. The purpose of this setting is to ensure all ACW data is collected for all calls.

encoding

Default Value: utf_8

Valid Values: Value defined in [Codec registry and base classes](#)

Changes Take Effect: After restart

Expected encoding type for strings queried from the ICON database. Refer to [Codec registry and base classes](#) to obtain the value that is applicable to your ICON database.

failed_folder_path

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the folder to save failed recordings to.

mode

Default Value: active

Valid Values: active, backup

Changes Take Effect: After restart

Specifies the High Availability Mode of the Recording Processor instance.

post_uri

Default Value: http://<active_rp_ip>:<port>/api/contact-centers/%s/recordings/

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the URL to be used by a backup node to send metadata to an active node (for HA).

get_from_httpc_before_posting

Default Value: 1

Valid Values: 0 (no), 1 (yes)

Changes Take Effect: After restart

Specifies whether Recording Processor is to fetch from data from Interaction Recording Web Services

(or Web Services if you're using version 8.5.210.02 or earlier) before sending a POST.

`acw_threshold_minutes`

Default Value: 5

Valid Values: 0 or any positive integer

Changes Take Effect: After restart

Specifies the maximum time, in minutes, that Recording Processor Script will wait in After Call Work (ACW) mode. If this time is exceeded, Recording Processor Script will skip collecting the ACW customized data from ICON. If set to 0, Recording Processor Script will not collect ACW customized data from ICON.

`enable_acw`

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to enable After Call Work data. If set to 0, collecting ACW data is disabled.

`backup_failed_metadata`

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to back up the messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner to the `<recording processor dir>\failed` folder.

`include_unknown_agent`

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

When this option is set to true (default), and if the agent's username is missing, then the username field will be populated as "UNKNOWN" in the eventHistory metadata. If this option is set to false, and if the agent's username is missing, then the username field will not be updated in the eventHistory metadata. This parameter is not available in the default configuration file. If you want to set this parameter to false, add it manually.

[+] client Section

`certs`

Default Value: tests/cacerts.txt

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the path to the Certificate Authority certificates used to validate the SSL connections to Workspace Web Edition and/or SpeechMiner.

`http_timeout`

Default Value: 20

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the HTTP timeout duration, in seconds, for the Recording Processor Script when sending messages to Workspace Web Edition and/or SpeechMiner.

[+] wait_times Section

`message_wait_seconds`

Default Value: 20

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the internal delay, in seconds, to wait before checking message queue.

`process_wait_seconds`

Default Value: 10

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the internal delay, in seconds, to wait between message processing attempts.

[+] metadata Section

`region`

Default Value: `region1`

Valid Values: Any string

Changes Take Effect: After restart

Specifies the region for the metadata.

`call_end_threshold_minutes`

Default Value: 30

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the length of time, in minutes, after a call ends before Recording Processor Script sends the call's metadata to SpeechMiner.

[+] rp_server Section

port

Default Value: 8889

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the port of the Recording Processor Script (RPS) REST server.

addr

Default Value: 0.0.0.0

Valid Values: Any valid IP address or 0.0.0.0

Changes Take Effect: After restart

Specifies the IP address of the Recording Processor Script (RPS) REST server.

ssl_certificate

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the path to the file that contains the SSL certificate required to support the HTTPS server.

ssl_private_key

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the path to the PEM file that contains the SSL private key required to support the HTTPS server.

Important

This file should **NOT** be password protected.

[+] htcc Section

base_uri

Default Value: http://<htcc_ip>:<port>

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the Base URI for accessing the Interaction Recording Web Services (or Web Services if

you're using version 8.5.210.02 or earlier) API.

get_uri

Default Value: /api/v2/ops/contact-centers/%s/recordings

Valid Values: Any valid URI reference

Changes Take Effect: After restart

Specifies the URI suffix used to retrieve metadata from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). This is appended to the base_uri. %s is replaced internally by the Contact Center ID. Normally it is sufficient to change the base_uri and this option does not need to be changed.

post_uri

Default Value: /internal-api/contact-centers/%s/recordings

Valid Values: Any valid URI reference

Changes Take Effect: After restart

Specifies the URI suffix used to send metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). This is appended to the base_uri. %s is replaced internally by the Contact Center ID. Normally it is sufficient to change the base_uri and this option does not need to be changed.

username

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

password

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

csrfp

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to enable the Cross Site Request Forgery (CSRF) protection mode. This option must be enabled (set to 1) if Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled.

[+] speechminer Section

post_uri

Default Value: Empty

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the URL used to send metadata to the SpeechMiner Interaction Receiver.

Note: This should not typically be specified; instead the information configured in the IVR Profile is used for SpeechMiner connectivity. Refer to the Recording Destinations section on the Recording Tab, described in the Step #4 table in the [IVR Profile](#) section.

username

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the SpeechMiner operating account.

Note: This should not typically be specified; instead the information configured in the IVR Profile is used for SpeechMiner connectivity. Refer to the Recording Destinations section on the Recording Tab, described in the Step #4 table in the [IVR Profile](#) section.

password

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the SpeechMiner operating account.

Note: This should not typically be specified; instead the information configured in the IVR Profile is used for SpeechMiner connectivity. Refer to the Recording Destinations section on the Recording Tab, described in the Step #4 table in the [IVR Profile](#) section.

disable_ssl_certificate_validation

Default Value: 1

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to disable validation of the SpeechMiner certificate when establishing a TLS connection to the SpeechMiner Interaction Receiver.

[+] persistence Section

table_name

Default Value: No default value

Valid Values: Any string

Changes Take Effect: After restart

Specifies the table name that the Recording Processor Script uses to store data.

`db_filename`

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the filename (full path or relative path) used for the sqlite3 database file.

[+] logfile Section

`max_log_file_size_mb`

Default Value: 21

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the maximum size, in MB, of the Recording Processor Script log file.

`logfile_path`

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the filename (full path or relative path) of the Recording Processor Script log file.

`level`

Default Value: INFO

Valid Values: DEBUG, INFO, WARNING, ERROR, CRITICAL

Changes Take Effect: After restart

Specifies the level or severity of the events to track in the log file. For more information about these logging levels, see the [Python documentation](#).

`log_backup_count`

Default Value: 100

Valid Values: Any integer (a non-zero integer is recommended)

Changes Take Effect: After restart

Specifies the number of log files to back up. When **log_backup_count** is a non-zero integer and the current log file reaches **max_log_file_size_mb**, the system closes the current log file and creates a new log file with the current time appended to the filename. For example, when **log_backup_count** is set as 5, the system will have backup for a maximum of 5 log files after the current log file size reaches **max_log_file_size_mb**.

Recording Crypto Server

Application-Level Options

The following describes the options for the Recording Crypto Server application.

[+] general Section

archive.enable

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Specifies whether to enable archiving for recordings.

samesite.enable

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Specifies whether the **SameSite=None** and **Secure** cookie attributes are set during screen recording playback from the SpeechMiner browser application.

Important

Before enabling this option, ensure that the connection between the SpeechMiner browser application and Recording Crypto Server is configured to use HTTPS. If you set the value of this option to `true` and are using HTTP, the cookie will not be returned by the browser.

gwsAuthUri

Default Value: Empty string

Valid Values: Any URL path

Changes Take Effect: After restart

This parameter is not supported. Do not change the default value.

[+] htcc Section

baseurl

Default Value: `https://htcchost:8080`

Valid Values: Any URL path

Changes Take Effect: After restart

Specifies the base URL for the connection to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). This parameter is dependent on the Interaction Recording Web Services (Web Services) server protocol (http, or https), port, and URL suffix.

internalUrlPrefix

Default Value: /api/v2

Valid Values: Any string

Changes Take Effect: After restart

Controls the prefix added to requests sent to Interaction Recording Web Services to retrieve recording files. By default, or if a value other than **disable** is specified, RCS will concatenate the **baseurl**, **internalUrlPrefix**, and the **mediaPath** returned by RWS as the request URL. If the **internalUrlPrefix** value is set to **disable**, RCS will use the **mediaUri** from the metadata instead when fetching the recordings from RWS.

domain

Default Value: Empty string

Valid Values: Any string

Changes Take Effect: After restart

Specifies the domain of the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) contact center. This is the domain ID set for the contact center within Interaction Recording Web Services (Web Services).

user

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the name of the user for the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection.

password

Default Value: opspassword

Valid Values: Any positive integer

Changes Take Effect: After restart

Specifies the password of the user for the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection.

max-sr-playback-connections

Default Value: 50

Valid Values: Any string

Changes Take Effect: After restart

Specifies the maximum number of HTTP connections between Recording Crypto Server and Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for

screen recording playback.

contactcenterid

Default Value: Empty string

Valid Values: Any string

Changes Take Effect: After restart

Specifies the contact center ID value in the RCS requests sent to Interaction Recording Web Services (RWS). If this value is not specified, the contact center ID information is derived from the **/api/v2/ops/contact-centers** request sent to RWS.

trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). For more information, see [Configuring TLS connection to Interaction Recording Web Services](#).

[+] cors Section

allowed-origins

Default Value: Empty

Valid Values: Any comma-separated list

Changes Take Effect: After restart

Specifies the allowed origins list that is attached in the HTTP response Access-Control-Allow-Origins header, sent to a cross-origin request.

allowed-headers

Default Value: X-Requested-With, Content-Type, Accept, Origin, Cookie, authorization, ssid, url, ContactCenterId, X-CSRF-TOKEN, Range

Valid Values: Any comma-separated list

Changes Take Effect: After restart

Specifies the allowed headers list that is attached in the HTTP response Access-Control-Allow-Headers header, sent to a cross-origin request.

allowed-methods

Default Value: GET, POST, PUT, DELETE, OPTIONS

Valid Values: Any comma-separated list

Changes Take Effect: After restart

Specifies the allowed methods list that is attached in the HTTP response Access-Control-Allow-

Methods header, sent to a cross-origin request.

allow-credentials

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Specifies the value sent in the Access-Control-Allow-Credentials header of the HTTP response to a cross-origin request.

[+] keystore Section

max-read-attempts

Default Value: 5

Valid Values: -1 or a positive integer. When using -1, RCS will retry endlessly.

Changes Take Effect: After restart

Specifies the maximum number of attempts to read keystore during startup.

read-interval

Default Value: 1

Valid Values: 1-30

Changes Take Effect: After restart

Specifies time interval, in seconds, between the keystore read attempts.

[+] log Section

suppress-debug-loggers

Default Value: Empty string

Valid Values: Any package name. The package name can contain the wildcard (*) character. For example, org.apache.http.wire.*.

Changes Take Effect: After restart

Suppresses the debug logs by Jetty for the specified package names.

Tenant-Level Options

The following describes the options for the Tenant.

[+] recording.archive Section

interval

Default Value: 1

Valid Values: 0–30

Changes Take Effect: After restart

Specifies how often, in days, the archiving process runs. If set to 0, archiving is disabled for the tenant.

retentiontime

Default Value: 60

Valid Values: Between 1 and 5*365 (5 years)

Changes Take Effect: After restart

Specifies how long (in days) to maintain the recordings in the system before they are archived by RCS.

outputfolder

Default Value: archive

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the destination folder where the archived recordings are stored.

speechminerurl

Default Value: https://<host or IP address of the SpeechMiner host>/speechminer

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the SpeechMiner database destination where the recording metadata is stored.

user

Default Value: archiveuser

Valid Values: Any string

Changes Take Effect: After restart

Specifies the SpeechMiner username that is required for authentication.

password

Default Value: changeit

Valid Values: Any string

Changes Take Effect: After restart

Specifies the SpeechMiner password that is required for authentication.

speechminer-trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to SpeechMiner Interaction Receiver. For more information, see [Configuring TLS connection to SpeechMiner Interaction Receiver](#).

Recording Plug-in for GAX

The following describes the options for the Plug-in for GAX (Genesys Administrator Extension).

Tenant-Level Options

[+] recording Section

rccurl

Default Value: `https://rcshost:8080`

Valid Values: Any URL

Changes Take Effect: After restart

Specifies the URL to the Recording Crypto Server.

htcc_base_url

Default Value: Empty

Valid Values: Interaction Recording Web Services load balancer URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; `http://<Interaction Recording Web Services IP Address>:8081`

Changes Take Effect: After restart

Specifies the HTCC Base URL.

trusted_ca_rcs

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Recording Crypto Server (RCS).

trusted_ca_rws

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Interaction Recording Web Services (RWS).

GAX-Level Options

[+] rcs Section

htcc_base_url

Default Value: Empty

Valid Values: Interaction Recording Web Services load balancer URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; http://<Interaction Recording Web Services IP Address>:8081

Changes Take Effect: After restart

Specifies the HTCC Base URL.

trusted_ca_rcs

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Recording Crypto Server (RCS).

trusted_ca_rws

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Interaction Recording Web Services (RWS).

htcc_http_timeout

Default Value: 20

Valid Values: Integers > 0

Changes Take Effect: After restart

Specifies the HTTP timeout value (in seconds) for HTCC API.

httpclient.cookiepolicy

Default Value: BEST_MATCH

Valid Values: rfc2109, rfc2965, BEST_MATCH, BROWSER_COMPATIBILITY, NETSCAPE, IGNORE_COOKIES

Changes Take Effect: After restart

Allows a user to specify which cookie policy to use for HTCC to work around the load balancer settings.

rccs_keepalive_interval

Default Value: 60

Valid Values: Integers > 0

Changes Take Effect: After restart

The interval in seconds the Plugin will be sending a request to RCS to keep the user session alive.

htcc_keepalive_interval

Default Value: 60

Valid Values: Integers > 0

Changes Take Effect: After restart

The interval in seconds the Plugin will be sending a request to HTCC API to keep the user session alive.

show_cert_with_upload_privilege

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Configures whether or not the GAX requires the **RECORD_KEY_UPLOAD** privilege to show the **Screen Recording Certificates** menu.

Recording Muxer Script

The following describes the options for the Recording Muxer Script.

[+] htcc Section

base_uri

Default Value: Empty

Valid Values: Any URL

Changes Take Effect: After restart

Specifies the host and port of the Interaction Recording Web Services server (or Web Services server if you're using version 8.5.210.02 or earlier) (for example, https://<web_services_host>:<web

`services port>/).`

`contact_center_id`

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the unique identifier of the contact center.

`username`

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

`password`

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

`trusted-ca`

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). For more information, see [Configuring TLS connection to Interaction Recording Web Services](#).

[+] rcs Section

`base_uri`

Default Value: Empty

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the host and port of the Recording Crypto Server (for example, `https://<Recording Crypto Server host>:<Recording Crypto Server port>`).

username

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the Recording Crypto Server account belonging to the contact center specified by the **contact_center_id** option in the **htcc** section.

password

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the Recording Crypto Server account belonging to the contact center specified by the **contact_center_id** option in the **htcc** section.

trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). For more information, see [Configuring TLS connection to Recording Crypto Server](#).

[+] processing Section

auto_clean_temp_folder

Default Value: 1

Valid Values: 1, 0

Changes Take Effect: After restart

If set to 1, Recording Muxer Script will automatically delete the recording files from the temp folder.

batch_read_screen_recording_metadata

Default Value: 1

Valid Values: Bulk API = 1 / GWS request = <>1

Changes Take Effect: After restart

Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.

check_matching_username

Default Value: 1

Valid Values: 1, 0

Changes Take Effect: After restart

If set to 1, Recording Muxer Script will mux a screen recording with a matching call recording only when the screen recording contains the same username value as in the call recording metadata. Note that Recording Muxer Script will always mux the recordings regardless of the **check_matching_username** value if the username is "UNKNOWN", undefined, or empty in the call recording metadata.

clean_temp_folder_timeout

Default Value: 43200

Valid Values: Integer

Changes Take Effect: After restart

Determines how often the recording files are cleaned up in the temp folder.

split_window_enabled

Default Value: 0

Valid Values: 1, 0

Changes Take Effect: After restart

Setting this parameter to 1 enables splitting the window into sub-intervals in order to improve Elasticsearch performance when querying RWS. If set to 0, this feature is disabled.

max_interval_minutes

Default Value: 30

Valid Values: 0 to total minutes between `window_past` and `window_past_older_than` values.

Changes Take Effect: After restart

If **split_window_enabled** is set to 1, **max_interval_minutes** specifies the maximum duration of the sub-interval in minutes.

muxer_type

Default Value: Empty

Valid Values: primary, backup

Changes Take Effect: After restart

Determines the primary Recording Muxer Script instance or backup Recording Muxer Script instance. If you select not to use Sharding, the system ignores this value.

muxer_id

Default Value: -1

Valid Values: A non-negative integer starting with 0 (every Muxer ID is incremented by 1)

Changes Take Effect: After restart

A unique Muxer ID. If you select not to use Sharding, the value should be empty or -1.

The Recording Muxer Script backup instance ID is automatically calculated as follows: $(id + 1) \% \text{total number of muxers}$

total_muxers

Default Value: Empty

Valid Values: $\max(\text{muxer_id}) + 1$

Changes Take Effect: After restart

The total number of Recording Muxer Script instances deployed (excluding the backup). If **muxer_id** is less than 0, the system ignores this value.

window_past

Default Value: 720

Valid Values: Any integer

Changes Take Effect: After restart

Specifies how many minutes into the past to search for those call recordings that are to be combined with the screen recordings.

window_past_older_than

Default Value: 5

Valid Values: Any integer

Changes Take Effect: After restart

Specifies how many minutes into the past to search for those call recordings that are to be combined with the screen recordings. The age of the call recording is determined by the current time and the endTime metadata. If left empty, the window_past parameter is used to search for call recordings. If not empty, then both the window_past and the window_past_older_than parameters are used to search for call recordings.

min_poll_interval

Default Value: 5

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the duration, in seconds, for the Recording Muxer Script to poll Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for the new recordings to be multiplexed.

ffmpeg

Default Value: ffmpeg

Valid Values: Any valid relative or absolute file path

Changes Take Effect: After restart

Specifies the ffmpeg executable with full directory path. If ffmpeg is specified in the executable path,

ffmpeg is sufficient to run the command.

ffprobe

Default Value: ffprobe

Valid Values: Any valid relative or absolute file path

Changes Take Effect: After restart

Specifies the path to the ffprobe executable file. If ffprobe is specified in the file path, ffprobe is sufficient to run the command.

openSSL

Default Value: openssl

Valid Values: Any valid relative or absolute file path

Changes Take Effect: After restart

Specifies the path to the openSSL executable file. If openSSL is specified in the file path, openSSL is sufficient to run the command.

encrypt_always

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to force the combined call and screen recordings to be uploaded as encrypted even if the source screen recording is not encrypted. Set to 0 if the screen recording is encrypted. Set to 1 if the screen recording was not encrypted.

temp_dir

Default Value: Empty

Valid Values: Any valid path

Changes Take Effect: After restart

Specifies the absolute path to the directory for storing the temporary files.

save_temp_file

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to delete the temporary files when the processing is complete. If set to 0, the temporary files are deleted when no longer needed. If set to 1, the temporary files are not deleted.

[+] webDav Section

username

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to allow read and write access to the WebDAV storage server.

password

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to allow read and write access to the WebDAV storage server..

trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to WebDAV. For more information, see [Configuring TLS connection to WebDAV](#).

[+] advanced Section

worker_threads

Default Value: 4

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the number of parallel processing threads to allow multiple call recordings and screen recording pairs to be combined in parallel.

pagination

Default Value: 100

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the number of records that are returned per page.

query_slice_size

Default Value: 100

Valid Values: < 0

Changes Take Effect: After restart

Defines the maximum number of call recording records whose screen recordings should be queried.

call_recording_extra_query_string

Default Value: If left empty, its value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if RWS version `>= 8.5.201.14`, otherwise, "" (empty string).

Valid Values: `callerPhoneNumber`, `dialedPhoneNumber`, `userName`, `userData`

Changes Take Effect: After restart

Specifies what the Recording Muxer Script is to query for with the Interaction Recording Web Services API (or Web Services API if you're using version 8.5.210.02 or earlier) along with `startTime` and/or `endTime` that are determined by "window_past" and/or "window_past_older_than" configurations. When non-empty (for example, `userData=SRSScreenRecordingStateStarted&userName=some_user`), the Recording Muxer Script is run against all the records found with the GET `<RWS_URL>/api/v2/ops/contact-centers/%s/recordings?startTime=%s&endTime=%s&<call_recording_extra_query_string>` query. If the result returns the `nextUri` attribute, the link will be followed with the records to run against. Once the records are traversed and processed, the Recording Muxer Script will exit.

call_recording_query_string

Default Value: Empty

Valid Values: `callerPhoneNumber`, `dialedPhoneNumber`, `startTime`, `endTime`, `userName`, `userData`

Changes Take Effect: After restart

Specifies what the Recording Muxer Script is to query for with the Interaction Recording Web Services API (or Web Services API if you're using version 8.5.210.02 or earlier). When non-empty (for example, `startTime=0&endTime=1000`), the Recording Muxer Script is run against all the records found with the GET `<RWS_URL>/api/v2/ops/contact-centers/%s/recordings?<call_recording_query_string>` query. If the result returns the `nextUri` attribute, the link will be followed with the records to run against. Once the records are traversed and processed, the Recording Muxer Script will exit. If left empty, the Recording Muxer Script periodically issues a query with the parameters generated internally to look at last N hours of records, combines the records, and never exits.

max_overlap_allowed

Default Value: 0

Valid Values: Any integer

Changes Take Effect: After restart

Specifies how much overlap time, in milliseconds, needs to occur before deciding to truncate the overlapping duration. When the muxed file's `startTime` and `stopTime` overlaps with the newly uploaded recording the muxed file's audio or video needs to be truncated to discard the previously silence/black filled duration.

video_padding_slice_length_ms

Default Value: 5000

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the duration, in milliseconds, of the prepended padded video slice. Set this parameter if the

video starts later than the audio.

mark_screen_recording_label

Default Value: 1

Valid Values: 1, 0

Changes Take Effect: After restart

If set to 1, Recording Muxer Script will automatically apply the label "screenRecording" to the associated call recording metadata after muxing.

[+] logfile Section

max_log_file_size_mb

Default Value: 21

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the maximum size, in megabytes, of the Recording Muxer Script log file before starting a new log file. If set to 0, the Recording Muxer Script log file will not roll over.

logfile_path

Default Value: Empty

Valid Values: Any valid path

Changes Take Effect: After restart

Specifies the path to the log file. If left empty, the working directory is used.

level

Default Value: INFO

Valid Values: DEBUG, INFO, WARNING, ERROR, CRITICAL

Changes Take Effect: After restart

Specifies the level or severity of the events to track in the log file.

Deploying Genesys Interaction Recording in a Multi-Tenant Deployment

Installation Considerations

This section describes the deployment steps required to configure GIR in a multi-tenant environment.

Important

- If you are currently working with a single-tenant deployment do not follow the instructions on this page, instead following the instructions in the [Migrate Genesys Interaction Recording from a Single Tenant to a Multi-Tenant Deployment](#) page.
- Each component should be sized according to the tenant sizing needs.
- GIR does not support WWE, when configured with a multi-tenant deployment.
- If you are using Voice Processor instead of RPS, you must deploy separate instances of Voice Processor for each tenant.

Once the deployment steps are completed, each tenant will include the following items:

[+] Show items.

- Users (that is, only users within a tenant are allowed to access GIR).
- Permissions.
- Access control of voice and screen recordings for search, view and playback.
- The ability to view the agent hierarchy.
- Recording conditions (full time recording, selective recording).
- Recording retention policies.
- Recording storage location and policies.
- Recording backup policies.
- Audit logs.
- Use of encryption keys.
- Administration of encryption keys.
- Screen recording policies.

- Quality management functionalities.

Multi-tenant Components

The following is a list of the components that can be deployed as multi-tenant components:

[+] Show multi-tenant components.

Important

When working with shared components across multiple tenants, you must verify that the components are sized properly and that shared components can handle the aggregate recording volume for all tenants.

- Management Framework
- GAX
- SQL Server
- SQL Server Reporting Services
- Cassandra
- Elasticsearch
- GVP Resource Manager
- GVP Media Control Platform

Deploying Multi-Tenancy

This section provides the tasks required to install and configure the Genesys components and features for Genesys Interaction Recording (GIR).

For the following steps, substitute:

- Tenant-specific RPS load balancer URL = <loadbalancer>/t1/rp
- Tenant-specific RCS load balancer URL = <loadbalancer>/t1/rcs
- Tenant-specific SMIR load balancer URL = <loadbalancer>/t1/interactionreceiver
- Tenant-specific WebDAV load balancer URL = <loadbalancer>/t1/recordings
- Tenant-specific Interaction Recording Web Services load balancer URL = <loadbalancer>/t1

...where <loadbalancer>/t1 refers to the load balancer which is described in [step 21 \(Load Balancing\)](#), and t1 is a tenant-specific identifier—subsequent tenants will use t2, t3, and so on.

To successfully deploy GIR in a multi-tenant deployment, you must perform the following procedures

in the order presented:

1. **Genesys Administrator Extension**
Follow the instructions on this page.
2. **Interaction Recording Web Services** (or **Web Services and Applications** if you're using version 8.5.210.02 or earlier)
In a multi-tenant deployment, each tenant must deploy a separate instance of Interaction Recording Web Services (Web Services). Consider and perform the following instructions while performing the steps in the **Interaction Recording Web Services (Web Services and Applications)** pages:
WebDAV [+] Show steps.
 - In a multi-tenant deployment, GIR should deploy a separate instance of Interaction Recording Web Services (Web Services), to store and manage the recording files. Each tenant should deploy a separate WebDAV Server instance. For details, follow the instructions in the **Deploy the WebDAV Storage Server** section.

Cassandra [+] Show steps.

- Cassandra is a shared service across multiple tenants. To deploy Cassandra, follow the instructions in the **Deploy the Cassandra Database** section.
- Each tenant must have a separate keyspace for Interaction Recording Web Services on Cassandra. Follow the steps in **Initializing Cassandra** with the following exceptions:
 - When deploying the **ks-schema** file, replace the keyspace name from **sipfs** to a tenant-specific name such as **sipfs1**, as follows:

```
CREATE KEYSPACE sipfs1 WITH replication = {'class':
'SimpleStrategy', 'replication_factor': '2'} AND durable_writes = true;
```

- After the keyspace is created, update the schema file **cf-schema.cql** by changing the first line from **sipfs** to the tenant-specific keyspace name **sipfs1**, as follows:

```
CREATE TABLE sipfs1.accounts (
  key text,
  column1 text,
  value blob,
  PRIMARY KEY (key, column1)
) WITH COMPACT STORAGE
AND CLUSTERING ORDER BY (column1 ASC)
AND bloom_filter_fp_chance = 0.01
AND caching = '{"keys":"ALL", "rows_per_partition":"NONE"}'
AND comment = ''
AND compaction = {'class':
'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}
AND compression = {'sstable_compression':
'org.apache.cassandra.io.compress.SnappyCompressor'}
AND dclocal_read_repair_chance = 0.0
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.1
AND speculative_retry = 'NONE';
```

Elasticsearch [+] Show steps.

- In a multi-tenant deployment, the recommended deployment approach for Elasticsearch is a standalone Elasticsearch cluster shared across multiple tenants. By default, each tenant owns separate indexes. The minimum number of Elasticsearch nodes that should be deployed in a High Availability (HA) environment is 3. To deploy Elasticsearch properly for a multi-tenant deployment, refer to either the [Elasticsearch](#) page or to the Prerequisites section in the Web Services and Applications Deployment Guide (if you're using GIR version 8.5.210.02 or earlier).

Interaction Recording Web Services **[+]** Show steps.

- If you are deploying Interaction Recording Web Services (Web Services) for a multi-tenancy deployment, you must create a separate Interaction Recording Web Services (Web Services) instance for each tenant (Note: Each instance can be a cluster of 2 or more High Availability nodes):
 1. If you are using version 8.5.210.02 or earlier, follow the steps in the [Installing](#) section of the *Web Services and Applications Deployment* guide. Otherwise, follow the instructions in the [Installing](#) page in this guide. **Note:** that the RWS instructions are different depending on whether you are using Interaction Recording Web Services by itself, or Interaction Recording Web Services together with Web Services.
 2. You must create a separate Cluster application for each tenant (note that this is shared per tenant between Interaction Recording Web Services and Web Services if you are using both services), and a separate Node application per tenant per server instance.
 - For the Cluster application:
Change the name to `IRWS_Cluster_<tenant>` or `WS_Cluster_<tenant>`, depending for which service the cluster is being created, where `<tenant>` is the tenant name.
In the **Tenants** tab, click **Add**, select the tenant object that you want to configure for Interaction Recording Web Services (or Web Services) and click **OK**.
 - For the Node application:
For a standalone deployment of Interaction Recording Web Services, make the connection to the `IRWS_Cluster_<Tenant>` application and name the application `IRWS_Node_<Tenant>`, where `<tenant>` is the tenant name.
For Web Services (when using version 8.5.210.02 or earlier) or a deployment where Interaction Recording Web Services is being used together with Web Services, make the connection to the `WS_Cluster_<Tenant>` application and name the application `WS_Node_<Tenant>` (when using version 8.5.210.02 or earlier), or `IRWS_Node_<Tenant>` (when using Interaction Recording Web Services together with Web Services) where `<tenant>` is the tenant name.
 3. For each Interaction Recording Web Services (Web Services) tenant instance, the **Jetty genconfig/application.yaml** must have the following parameters:
 - `keyspace:` <Cassandra keyspace assigned to this tenant as per "Create separate keyspace">
 - `nodes:` <the shared Cassandra nodes>
 4. For each Interaction Recording Web Services (Web Services) tenant instance, the **Jetty genconfig/application.yaml** must have the following parameters (if you are using Web Services and Application version 8.5.201.09 or earlier modify the **server-settings.yaml** and not **application.yaml**):
 - `externalApiUrlV2:` `http://<tenant-specific Interaction Recording Web Services load balancer URL>/api/v2`
 - `internalApiUrlV2:` `http://<tenant-specific Interaction Recording Web Services load balancer URL>/internal-api`
 - `'''undocumentedExternalApiUrl'''`: `http://<tenant-specific Interaction`

Recording Web Services load balancer URL>/internal-api

- `applicationName`: <the WS_Node_<Tenant> name assigned for this tenant>
- `crClusterName`: <Elasticsearch cluster name as per cluster.name>
- `crossOriginSettings/allowedOrigins`: <tenant-specific Interaction Recording Web Services Servers>, <tenant-specific SpeechMiner Web Servers>

5. Configure **Voice Recordings**: Follow the same instructions as the single tenant instructions. Whenever Interaction Recording Web Services server (Web Services server) is specified replace it with the tenant-specific Interaction Recording Web Services server (Web Services server) instance.

When reference is made to the storage credentials, use the tenant-specific WebDAV Server credentials as configured in the WebDAV Server section above.

6. Configure **Screen Recordings**: Follow the same instructions as the single tenant instructions. Whenever Interaction Recording Web Services server (Web Services server) is specified replace it with the tenant-specific Interaction Recording Web Services server (Web Services server) instance.

When reference is made to the storage credentials, use the tenant-specific WebDAV Server credentials as configured in the WebDAV Server section above.

3. SIP Server

In a multi-tenant deployment, each tenant must deploy a separate SIP Server tenant instance, each with its own tenant-specific switch object.

The GIR deployment instructions for SIP Server are the same as those for a single tenant.

4. Interaction Concentrator (ICON)

When following the instructions in this page, use a tenant-specific instance of ICON and ICON DB.

5. Recording Plug-in for GAX

Execute the **Solution Deploy SPD** file for each tenant, in order to create the appropriate tenant Access Groups, Roles, and Permissions. For additional information, refer to the [Example Solution SPD File](#) page.

Follow the instructions in the **"Multi-Tenant Environment"** subsection within the **"Configure for Screen Recording"** section.

6. Depending on the component you are using between Voice Processor and Recording Processor Script, follow one of the two sets of instructions below:

Voice Processor

Follow the instructions in this page, noting the following.

- a. Deploy separate instances of the Voice Processor for each tenant.
- b. Deploy a separate Voice Processor database for each tenant (these may all reside within a common database server if desired, as long as the total load on the Voice Processors does not exceed 30 calls per second).
- c. Configure the following settings in each instance's **settings-override.yml** to point to the tenant's RWS cluster and Voice Processor database:
 - **rwsBaseUri**
 - **nodeRpsDb**
- d. The **rps-provisioning** settings group should be established in each tenant's RWS.
- e. Verify that the Voice Processor host names, and the configured **authUsername** and **authPassword** for each Voice Processor instance match the configuration in each tenant's IVR profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.

Recording Processor Script (RPS)

Follow the instructions in this page except for the following.

- a. Deploy separate instances of the Recording Processor for each tenant.
- b. Replace the following configuration parameters with tenant-specific instances of Interaction Recording Web Services (or Web Services and Applications if you're using version 8.5.210.02 or earlier) and ICON DB:
 - `htcc.base_uri`—Set this to the Tenant-specific Interaction Recording Web Services load balancer URL.
 - `htcc.username`
 - `htcc.password`
 - `icon_db_servers`
- c. Verify that the username and password match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.

7. Recording Crypto Server (RCS)

When deploying RCS use a tenant-specific instance of RCS.

Follow the instructions in this page, except for the following: **[+] Show steps.**

- In the **RCS application object**, replace the following parameters with tenant-specific instances of Interaction Recording Web Services:
 - `htcc.baseurl`—Set to the Tenant-specific Interaction Recording Web Services load balancer URL.
 - `htcc.user`
 - `htcc.password`
 - `cors.allow-origins`—Set to the Tenant-specific Interaction Recording Web Services load balancer URL.
- In the **Annex** tab for the specific tenant object:
 1. Create a new section called **Recording**.
 2. Create a parameter called: `rcsurl`.
 3. Set its value to the Tenant-specific RCS load balancer URL.
 4. In the `[recording.archive]` section, set `speechminerurl` = `<Tenant-specific SMIR load balancer URL>`.

8. Genesys Voice Platform (GVP)

GVP is a shared resource for all tenants. Follow the instructions in this link to deploy the **Resource Manager** and **Media Control Platform** that are shared for all tenants.

When creating the tenant-specific **IVR** profile, change the steps as follows: **[+] Show steps.**

- **In Step 1** on the [Genesys Voice Platform \(GVP\)](#) page:
 - a. Under **GAX**, switch the view to the tenant you want to configure.
 - b. Navigate to **Configuration > System > Configuration Manager** and under **Voice Platform**, select **Voice Platform Profiles** and click **New**.
- **In Step 3** on the [Genesys Voice Platform \(GVP\)](#) page:
 - Change **recordingclient.callrec_authorization** to the tenant-specific `rp_username` and

tenant-specific `rp_password`. See [these settings under Configuring Basic Authorization](#).

- **In Step 4** on the [Genesys Voice Platform \(GVP\)](#) page:
In the **Recording** tab, following the same instructions, but change the following parameters with tenant-specific information:
 - Storage destination—`http://<Tenant-specific WebDAV load balancer URL>`
 - Storage HTTP Authorization header—`<Tenant WebDAV Server authorization>`
 - Recording Processor—URI—`http://<Tenant-specific RPS load balancer URL>/api/contact-centers/<Contact Center Domain Name>/recordings/`
 - SpeechMiner Interaction Receiver—`http://<Tenant-specific SMIR load balancer URL>`
 - SpeechMiner Interaction Receiver Authorization header—`<Tenant Interaction Receiver authorization>`

9. [Deploying Encryption](#)

Follow the instructions in this page and perform the following: **[+] Show steps**.

1. Upload tenant certificates:

- a. Log into GAX using a user account belonging to the tenant.
- b. Follow the instructions about how to configure the encryption of voice recordings in the [For Call Recordings](#) section.

3. For each tenant, configure the decryption proxy by following the steps in the [Setting up the Decryption Proxy](#) section.

Note: Replace **Web Services** and **RCS URI** with the tenant-specific addresses.

10. [Screen Recording Service](#)

Follow the instructions on this page.

When using the command line to install the SR Service, change the **/server** parameter in the setup to point to the Interaction Recording Web Services tenant instance. For example: `setup.exe /server="<Tenant-specific Interaction Recording Web Services load balancer URL>"`.

11. [Screen Recording Service - Advanced Configuration](#)

Follow the instructions in this page for each tenant and replace the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) instance with an Interaction Recording Web Services (Web Services) tenant instance.

For WDE, follow [these WDE instructions](#) and set the following:

`screen-recording.htcc.uri`: `<Tenant-specific Interaction Recording Web Services load balancer URL>`

12. [Recording Muxer Script](#)

Follow the instructions in this page to deploy tenant-specific instances of Muxer.

Replace the following parameters with tenant-specific values:

- `htcc.base_uri`—Set this to the Tenant-specific Interaction Recording Web Services load balancer URL.
- `htcc.contact_center_id`
- `htcc.username`
- `htcc.password`
- `rccs.base_uri`—Set this to the Tenant-specific RCS load balancer URL.
- `rccs.username`

- `rcs.password`
- `webdav.username`
- `webdav.password`

13. Interaction Recording Options Reference

Refer to this page for a description of the configuration options.

14. User Access

Follow the instructions in this page and consider the following: **[+] Show notes.**

Deployment Steps	Description
1	<p>In order to restrict log-in to the SpeechMiner UI, a new Configuration Manager object must be created for each tenant. Backup the default Configuration Manager object, since this object is accessible by all users from all tenants.</p> <p>For the tenant-specific Configuration Manager object, the permission must be restricted to tenant users (for example, TenantN\Administrators and TenantN\Users), and super administrators.</p>
2	<p>The agent hierarchy for each tenant must be configured to ensure the agents for each tenant are on a different agent hierarchy. While the previous step prevents a user from connecting to a SpeechMiner UI associated with a tenant the specific user is not allowed to use, the agent hierarchy ensures a unified hierarchy for all users in the Management Framework. For example, <code>recording.agent_hierarchy</code> for users in Tenant 1 would start with <code>/Tenant1</code>. Sub-teams could be a child of <code>/Tenant1</code>.</p>
3	<p>Users of each tenant will be assigned to the appropriate Roles in each tenant created by the Recording SPD. The environment tenant should not be expected to have the Roles created by the SPD.</p>
4	<p>Users of each tenant will be assigned to the appropriate Access Groups in each tenant (but not the Environment tenant). Although the Recording SPD always creates a "/" for each tenant, it is recommended (per step 2) to create a <code>/TenantX</code> Access Group object and assign it to the appropriate tenant users.</p>

15. Speech and Text Analytics (SpeechMiner)

When following the instructions in this page create a separate SpeechMiner instance for each tenant. To do this, you must configure the following items for each tenant: **[+] Show items.**

SpeechMiner Database

Deployment Steps	Description
1	A shared SQL Server instance can be used by all SpeechMiner tenants. All SpeechMiner machines must be configured to connect to the same SQL Server and create a separate SpeechMiner database for each tenant.
2	A separate SpeechMiner database must be configured for each tenant. When installing the database portion of the SpeechMiner installation, create a new database for each tenant.

SQL Server Reporting Services

Deployment Steps	Description
1	A shared SQL Server Reporting Service can be used by all SpeechMiner tenants. All SpeechMiner machines must be configured to connect to the same SQL Server and create a separate SpeechMiner database for each tenant.

SpeechMiner Server Configuration

Deployment Steps	Description
1	<p>For each tenant, create separate application objects for SpeechMiner components:</p> <ul style="list-style-type: none">• <Speechminer prefix>_ClientApplications• <Speechminer prefix>_InteractionReceiver• <Speechminer prefix>_Platform• <Speechminer prefix>_Web <p>Where <Speechminer prefix> is a name assigned for the tenant. This name is used later in step 7.</p>
2	For each tenant in the SpeechMiner server, create separate UNC paths for the data directory. The data directory will hold files for Index, plus additional SpeechMiner files for the tenant.
3	After successfully installing the SpeechMiner software on each server, run SMConfig on each server, and set the Database to the tenant-specific SpeechMiner database instance. For example, speechminer855_sm2 is the database name.
4	<p>In the Sites & Machines page:</p> <ul style="list-style-type: none">• Configure the paths to folders for the tenant created in step 2• Configure machines for this tenant for Web, Interaction Receiver, and Index

Deployment Steps	Description
	<ul style="list-style-type: none">• Configure the primary/backup configuration server IP:port
5	<p>In the Report Deployment page:</p> <ol style="list-style-type: none">1. Enter the server name for the shared SQL Server Reporting Services and deploy the MRSLibrary and reports for this tenant.2. After deploying the reports, use Internet Explorer to make sure the connection to the database source is configured. The default location is http://<SSRS server>/reports.3. Click the database name that was chosen for this tenant.4. Click SME to access the data source properties.5. Verify that the data source has a proper connection.6. Click Test Connection to validate, and click Apply to save the settings.
6	Follow existing deployment instructions for License, Services, Audio, and Index.
7	<p>In the Recording tab, each tenant has different settings:</p> <ul style="list-style-type: none">• Configuration• Tenant - The tenant name as per the configuration server.• Application Name - The SpeechMiner prefix for the application objects as per step 1.• Users Access Group - The tenant-specific Access Group that contains the list of SpeechMiner users for the tenant.• User Application Name - The name of the Configuration Manager application object for the tenant as per step 1 in the User Access section above.• Interaction Receiver - The RP Authorization.• RP Authorization - Authorization for the tenant-specific RP.• Playback• RCS URI = <Tenant-specific RCS load balancer URL>

Deployment Steps	Description
	<ul style="list-style-type: none">• External RCS URI = <Tenant-specific RCS load balancer URL>• HTCC URL = <Tenant-specific Interaction Recording Web Services load balancer URL>

16. **Workspace Desktop Edition**

Follow the instructions in this page.

17. **Secure Transport Configuration (TLS)**

Follow the instructions in this page.

18. **Media Lifecycle Management**

Follow the instructions in this page and verify that the Interaction-Receiver settings group points to a tenant-specific Interaction Receiver. In the SpeechMiner section, use tenant Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and tenant Interaction Receiver instances. Set the SpeechMiner Interaction Receiver URL to the Tenant-specific SMIR load balancer URL when enabling Interaction Recording Web Services to contact Interaction Receiver.

19. **Folder Hierarchy for Recording Storage**

Follow the instructions in this page using a tenant WebDAV server instance.

20. **Feature Configuration**

Follow the instructions in this page.

Note: Audio Tones are applicable on a per-tenant basis.

21. **Load Balancing**

Follow the instructions in this page.

Deploying Interaction Recording Web Services (RWS)

Warning

The content on this page only applies to versions of Genesys Interaction Recording newer than 8.5.210.02. If you're using an earlier version, you'll need to install Web Services and Applications instead. See [Deploying Web Services and Applications for GIR](#) for details.

If you upgrade to Interaction Recording Web Services (RWS), it does not provide API support for non-GIR related Web Services, such as Workspace Web Edition.

Genesys Interaction Recording (GIR) needs the Interaction Recording Web Services component to store and manage recording files.

Complete the steps below to install and configure Interaction Recording Web Services and its supporting components:

1. [Review the Prerequisites](#). Make sure all supporting components are installed and configured.
2. [Deploy Cassandra](#)
3. [WebDAV Requirements](#)
4. [Initialize Cassandra](#)
5. [Elasticsearch](#)
6. [Install Interaction Recording Web Services \(RWS\)](#)
7. [Deploy the web application](#)
8. [Configure Interaction Recording Web Services](#)
9. [Configure additional security \(optional\)](#)
10. [Start and test Interaction Recording Web Services](#)
11. [Configure your required features](#)

Prerequisites

Before deploying and configuring Interaction Recording Web Services, make sure your system meets the following minimum requirements:

OS Requirements

See the [Genesys Interaction Recording](#) page in the *Genesys Supported Operating Environment Reference* for more detailed information and a list of all supported operating systems.

Java Requirements

- From Interaction Recording Web Services version 8.5.205.69 (or higher), you have installed the latest JDK 17 (64-bit for Linux). Alternatively, you can also install the latest standalone JRE 17 (64-bit for Linux). You can choose to download the software from an OpenJDK version of the software.
- From Interaction Recording Web Services version 8.5.205.65 (or lower), you have installed the latest JDK 8 (64-bit for Linux). Alternatively, you can also install the latest standalone JRE 8 (64-bit for Linux). You can choose to download the software from Oracle or obtain an OpenJDK version of the software.

Cassandra Requirements

Interaction Recording Web Services stores information about call and screen recordings in a Cassandra database. For each contact center, distinct column families with unique names exist for storing recording information. These column families are created when the contact center is created, and deleted if the contact center is deleted.

Important

- Interaction Recording Web Services deletes column families only if they do not contain any call recordings; otherwise they should be deleted manually from Cassandra using the `cqlsh` utility tool.
- Interaction Recording Web Services and Web Services and Applications share the same Cassandra instance within the same deployment. If you are using Interaction Recording Web Services with Web Services and Applications in the same environment, verify that your Cassandra version is the same for both components and all nodes.

Interaction Recording Web Services requires that your environment includes Cassandra 1.2 or 2.2. Genesys recommends Cassandra version 2.2. Complete the steps in these procedures below to install and configure Cassandra 2.2:

- [Deploying Cassandra 2.2](#)
 - [Installing and Configuring Cassandra 2.2](#)
 - [Upgrading to Cassandra 2.2](#)

Interaction Recording Web Services 8.5.500.17 (or higher) requires that your environment includes Cassandra 4.1. Complete the steps in these procedures below to upgrade and configure Cassandra 4.1:

- [Upgrading Cassandra to 4.1](#)

Interaction Recording Web Services 8.5.500.30 (or higher) supports install of Cassandra 4.1. Complete the steps in these procedures below to install Cassandra 4.1:

- [Installing and Configuring Cassandra 4.1](#)

Genesys Environment

Before installing and configuring the [Interaction Recording Web Services](#) 8.5.500.17 (or higher), you must have the following prerequisites:

- A [Recording Crypto Server](#) 8.5.095.22 (or higher) instance to decrypt the encrypted recordings.
- A [Recording Plugin for GAX version](#) 8.5.500.08 (or higher) for certificate updation and MLM configuration.

For more information about the required Genesys environment for GIR, refer to the [Minimum Recommended Versions](#).

Next Step

- [WebDAV Requirements](#)

Deploying Cassandra

Before you start installing and configuring Genesys Interaction Recording (GIR), you must first install and configure Cassandra. GIR supports Cassandra version 1.2 or 2.2 or 4.1.

For new deployments, use Cassandra 4.1. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the [Cassandra 4.1 documentation](#).

For general instructions and guidelines, select one of the following links:

- [Installing and Configuring Cassandra 4.1](#)
- [Upgrading to Cassandra 4.1](#)

Installing and Configuring Cassandra

Cassandra 4

Installing Cassandra4

Complete this procedure for each Cassandra node.

Prerequisites

- For new deployments, we recommend Cassandra 4.1.x. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the [Cassandra 4.1 documentation](#).
- You have installed the latest supported openJDK 8.

Start

1. [Download the latest 4.1.x version of Cassandra](#).
2. Copy the Cassandra archive to the installation directory. For example, **/usr/local**
3. Use a tar utility to extract the files. For example, `tar -zxvf apache-cassandra-4.1.9-bin.tar.gz`
4. Add directories for data, commitlog, and saved_caches. You can create these directories anywhere or in the default locations configured in the ***Cassandra_install_dir/conf/cassandra.yaml*** file. For example:
 - **/var/lib/cassandra/data**
 - **/var/lib/cassandra/commitlog**
 - **/var/lib/cassandra/saved_caches**

End

Configuring Cassandra4

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development: 1 Cassandra node (appropriate for a development or lab environment)
- Single Datacenter: 1 datacenter with a minimum of three Cassandra nodes
- Two Datacenters: 2 datacenters with a minimum of three Cassandra nodes

Important

For more complex Cassandra deployments, please consult with Genesys

Prerequisites

- [Installing Cassandra4](#)

Follow one of the procedures that matches your deployment scenario.

Important

The files modified in this procedure are typically found in the ***Cassandra_install_dir/conf*** directory.

Development

Configuring Cassandra (1 Cassandra node)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set `cluster_name` to the desired name.
 - b. Set `seeds` to the list of host name of the node. For example: `- seeds: "127.0.0.1"`
 - c. Set `listen_address` and `broadcast_address` to the IP of the node.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra4](#).
 - e. Set the `endpoint_snitch` parameter to `SimpleSnitch` (for development only).
6. Save your changes and close the file.

End

Single Datacenter

Configuring Cassandra (1 datacenter)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set the `cluster_name`. It must be the same name on all nodes.
 - b. Set `seeds` to the list of host names of all nodes. For example: `- seeds: "node1, node2, node3"`

- c. Set `listen_address` and `broadcast_address` to the IP of the node.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra4](#).
 - e. Set the `start_rpc` parameter to `true`.
 - f. Change `endpoint_snitch` to `GossipingPropertyFileSnitch`.
7. Save your changes and close the file.
 8. Open the **`cassandra-rackdc.properties`** file and update for your cluster topology. For each node in your cluster, add the following line:

```
dc=[datacenter] rack=[rackname]
```

Where:

- `[datacenter]` is the name of the datacenter for this node.
- `[rackname]` is the name of the rack for this node.

The following is a sample **`cassandra-rackdc.properties`** file for a Single Datacenter scenario:

```
dc=datacenter1 rack=rack1
```

9. Save your changes and close the file.

End

Two Datacenters

Configuring Cassandra (2 datacenters)

Start

1. Modify the **`cassandra.yaml`** file:
 - a. Set the `cluster_name`. It must be the same name on all nodes.
 - b. Set `seeds` to the list of host names of all nodes. For example: `- seeds: "node1, node2, node3, node4, node5, node6"`
 - c. Set `listen_address` and `broadcast_address` to the host name.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra4](#).
 - e. Change `endpoint_snitch` to `GossipingPropertyFileSnitch`.
6. Save your changes and close the file.
7. Open the **`cassandra-rackdc.properties`** file and update for your cluster topology. For each node in your cluster, add the following line:

```
dc=[datacenter] rack=[rackname]
```

Where:

- `[datacenter]` is the name of the datacenter for this node.

- *[rackname]* is the name of the rack for this node.

The following is a sample **cassandra-rackdc.properties** file for a Two Datacenter scenario:

For datacenter1 nodes:

```
dc=datacenter1 rack=rack1
```

For datacenter2 nodes:

```
dc=datacenter2 rack=rack1
```

8. Save your changes and close the file.

End

Verifying the Cassandra4 installation

Prerequisites

- [Configuring Cassandra](#)

Start

1. Start all Cassandra nodes using the following command: *Cassandra_install_dir/bin/cassandra -R*
2. Use the nodetool utility to verify that all nodes are connected by entering the following command: *Cassandra_install_dir/bin/nodetool status*

The following is sample output for a Single Datacenter scenario with three Cassandra nodes:

```
/genesys/apache-cassandra-4.1.4/bin$ ./nodetool status
Address      DC           Rack  Status  State  Load      Owns      Token
192.0.2.10   datacenter1 rack1  Up      Normal 14.97 MB  100.00%   -9223372036854775808
192.0.2.11   datacenter1 rack1  Up      Normal 14.97 MB  100.00%   -3074457345618258603
192.0.2.12   datacenter1 rack1  Up      Normal 14.97 MB  100.00%   3074457345618258602
```

The following is sample output for a Development scenario with a single Cassandra node:

```
/genesys/apache-cassandra-4.1.4/bin$ ./nodetool status
Address      DC           Rack  Status  State  Load      Effective-
Ownership Token
127.0.0.1    datacenter1 rack1  Up      Normal 1.89 MB
100.00%      76880863635469966884037445232169973201
```

End

Cassandra 2

Installing Cassandra2

Complete this procedure for each Cassandra node.

Prerequisites

- For new deployments, we recommend Cassandra 2.2. The procedures below are meant to serve as a quick guide on how to do this. For more detailed information, see the [Cassandra 2.2 documentation](#).
- You have installed the latest [Java SE Development Toolkit 8](#). For more information, refer to the [Java documentation](#).

Start

1. [Download the latest 2.2.x version of Cassandra](#).
2. Copy the Cassandra archive to the installation directory. For example, **/usr/local**
3. Use a tar utility to extract the files. For example, `tar -zxvf apache-cassandra-2.2.7-bin.tar.gz`
4. Add directories for data, commitlog, and saved_caches. You can create these directories anywhere or in the default locations configured in the **Cassandra_install_dir/conf/cassandra.yaml** file. For example:
 - **/var/lib/cassandra/data**
 - **/var/lib/cassandra/commitlog**
 - **/var/lib/cassandra/saved_caches**

End

Configuring Cassandra2

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development: 1 Cassandra node (appropriate for a development or lab environment)
- Single Datacenter: 1 datacenter with a minimum of three Cassandra nodes
- Two Datacenters: 2 datacenters with a minimum of three Cassandra nodes

Important

For more complex Cassandra deployments, please consult with Genesys

Prerequisites

- [Installing Cassandra4](#)

Follow one of the procedures that matches your deployment scenario.

Important

The files modified in this procedure are typically found in the ***Cassandra_install_dir/conf*** directory.

Development

Configuring Cassandra (1 Cassandra node)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set seeds to the list of host name of the node. For example: -seeds: "127.0.0.1"
 - b. Set listen_address and rpc_address to the host name.
 - c. Set data_file_directories, commitlog_directory, and saved_caches_directory to the directories you created in Step 4 of [Installing Cassandra](#).
 - d. Set the start_rpc parameter to true.
5. Save your changes and close the file.

End

Single Datacenter

Configuring Cassandra (1 datacenter)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set the cluster_name. It must be the same name on all nodes.
 - b. Set seeds to the list of host names of all nodes. For example: -seeds: "node1, node2, node3"
 - c. Set listen_address and rpc_address to the host name.
 - d. Set data_file_directories, commitlog_directory, and saved_caches_directory to the directories you created in Step 4 of [Installing Cassandra](#).
 - e. Set the start_rpc parameter to true.
 - f. Change endpoint_snitch to PropertyFileSnitch.
7. Save your changes and close the file.
8. Open the **cassandra-topology.properties** file and update for your cluster topology. For each node in your cluster, add the following line:

```
[node]=[datacenter]:[rack]
```

Where:

- *[node]* is the IP address of the node.
- *[datacenter]* is the name of the datacenter for this node.
- *[rack]* is the name of the rack for this node.

The following is a sample **cassandra-topology.properties** file for a Single Datacenter scenario:

```
192.0.2.10=datacenter1:rack1
192.0.2.11=datacenter1:rack1
192.0.2.12=datacenter1:rack1
```

9. Save your changes and close the file.

End

Two Datacenters

Configuring Cassandra (2 datacenters)

Start

1. Modify the **cassandra.yaml** file:
 - a. Set the `cluster_name`. It must be the same name on all nodes.
 - b. Set seeds to the list of host names of all nodes. For example: `-seeds: "node1, node2, node3, node4, node5, node6"`
 - c. Set `listen_address` and `rpc_address` to the host name.
 - d. Set `data_file_directories`, `commitlog_directory`, and `saved_caches_directory` to the directories you created in Step 4 of [Installing Cassandra](#).
 - e. Set the `start_rpc` parameter to `true`.
 - f. Change `endpoint_snitch` to `PropertyFileSnitch`.
7. Save your changes and close the file.
8. Open the **cassandra-topology.properties** file and update for your cluster topology. For each node in your cluster, add the following line:

```
[node]=[datacenter]:[rack]
```

Where:

- *[node]* is the IP address of the node.
- *[datacenter]* is the name of the datacenter for this node.
- *[rack]* is the name of the rack for this node.

The following is a sample **cassandra-topology.properties** file for a Two Datacenter scenario:

```
192.0.2.10=datacenter1:rack1
192.0.2.11=datacenter1:rack1
192.0.2.12=datacenter1:rack1
```



```
198.51.100.10=datacenter2:rack1
198.51.100.11=datacenter2:rack1
198.51.100.12=datacenter2:rack1
```

9. Save your changes and close the file.

End

Verifying the Cassandra2 installation

Prerequisites

- [Configuring Cassandra](#)

Start

1. Start all Cassandra nodes using the following command: *Cassandra_install_dir/bin/cassandra*
2. Use the nodetool utility to verify that all nodes are connected by entering the following command: *Cassandra_install_dir/bin/nodetool -h Cassandra_host ring*

The following is sample output for a Single Datacenter scenario with three Cassandra nodes:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool ring
Address      DC           Rack  Status  State  Load      Owns      Token
192.0.2.10   datacenter1 rack1  Up      Normal  14.97 MB  100.00%   -9223372036854775808
192.0.2.11   datacenter1 rack1  Up      Normal  14.97 MB  100.00%   -3074457345618258603
192.0.2.12   datacenter1 rack1  Up      Normal  14.97 MB  100.00%   3074457345618258602
```

The following is sample output for a Development scenario with a single Cassandra node:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool ring
Address      DC           Rack  Status  State  Load      Effective-
Ownership Token
127.0.0.1    datacenter1 rack1  Up      Normal  1.89 MB
100.00%      76880863635469966884037445232169973201
```

End

Upgrading Cassandra to 2.2

Genesys Interaction Recording (GIR) supports Cassandra versions 2.2 and 1.2. If you are using Cassandra 1.2, you can maintain this version or upgrade to Cassandra 2.2.

Directly upgrading from 1.2 to 2.2 is not supported, therefore you need to upgrade your Cassandra versions in several steps. For example 1.2 > 2.0 > 2.1 > 2.2. For more information about upgrading Cassandra, see the **Upgrading Apache Cassandra** section in [Cassandra Upgrade Guide](#).

Important

No Screen Recordings are made during the upgrade of Cassandra.

To minimize the risk of losing screen recordings, upgrade Cassandra during off-hours. During the migration, GIR still operates with voice recording capabilities. After the Cassandra cluster is back in service, you can process the metadata information for voice recordings that was saved during the migration by initiating the recovery procedure of the Recording Processor Script—as described in step 5 below.

1. Stop all Interaction Recording Web Services nodes.
2. Perform Cassandra upgrade according to the Cassandra Upgrade Guide.
3. When configuring Cassandra 2.2 according to the Datastax instructions, you must enable the thrift interface. Set the **start_rpc** parameter to true in the **cassandra.yaml** file
4. Start all Interaction Recording Web Services nodes.
5. After the upgrade, run the [LVR Recovery Script](#) to recover recordings from the **Recording Processor failed** folder and repost the recordings to Interaction Recording Web Services, as failed voice recordings accumulate while the Cassandra cluster is unavailable.

Upgrading Cassandra to 4.1

Important

- If you are using GIR with Workspace Web Edition, refrain from using a shared deployment of Cassandra. This is because Web Services and Applications support Cassandra 2.2 only and do not support Cassandra 4.1. For details, see: [Installing and configuring Cassandra](#).
- Genesys Interaction Recording Web Services (GIR RWS) version 8.5.500.17 (or higher) supports Cassandra 4.1 only.

If you are using Cassandra 2.2, you must upgrade to Cassandra 4.1. Upgrading directly from 2.2.x to 4.1 is not supported. You must upgrade your Cassandra versions incrementally for schema portability, for example, from version 2.2.x -> 3.11.11 -> 4.1.x. Requires Python 3.6 (or higher) and Java 8 for Cassandra 4.1.

For more information about upgrading Cassandra, refer the **Upgrade to Cassandra 4.1** section.

Important

No screen recordings are made during the upgrade of Cassandra.

To minimize the risk of losing screen recordings, upgrade Cassandra during off-hours. It's a best practice to perform a cluster upgrade during a scheduled maintenance window that occurs when application loads are typically lower. During the migration, GIR still operates with voice recording capabilities. After the Cassandra cluster is back in service, you can process the metadata information for voice recordings that was saved during the migration by initiating the recovery procedure of the Recording Processor Script.

Pre-requisites:

1. Ensure that the Recording Processor Script (RPS) or GIR Voice Processor (GIR VP), and Recording Web Services (RWS) nodes debug level logs are stored for any debugging or troubleshooting issues related to migration.
2. Back up all your Cassandra configuration files.
3. Take snapshots of all keyspaces on each existing Cassandra node. For more information, refer [Cassandra Documentation](#).
4. Confirm that your Cassandra environment satisfies all of the necessary [prerequisites](#).

Upgrade to Cassandra 4.1:

The following section provides the instructions for performing in-place upgrade from Cassandra 2.2.X to Cassandra 4.1.X. The recommendation procedure should be followed for quicker rollback and

minimal downtime.

Recommendation:

Warning

Genesys is not responsible for migration of existing data to the replicated Cassandra cluster on premise environments.

1. Create a replica of existing Cassandra 2.2 cluster with the data. For more information refer to [Cassandra Documentation](#).
2. Ensure that replicated Cassandra 2.2 cluster data is validated with existing RWS nodes by updating the following parameters in `application.yaml`.

`nodes`: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>] and `backup_nodes`: [ToBeChangedOrRemoved: <CASSANDRA_BACKUP_DC_NODES>] to respective replicated nodes. For more information, check [RWS Options for cassandraCluster](#).

1. Execute Migration plan on replicated Cassandra 2.2 new Cluster.

Migration Plan:

1. Stop all Interaction Recording Web Services nodes.
2. Run the schema upgrade application. This will be packaged with the RWS IP installation files. This application need to be run against one of the Cassandra 2.2 cluster nodes.

```
java -jar upgrade.jar [options]
```

options:

 - cassandraHost <host> - required, no default
 - thriftPort <port number> - optional, default=9160
 - nativePort <port number> - optional, default=9042
 - keyspace <keyspace name> - optional, default=sipfs
 - datacenter <datacenter name> - optional, default=datacenter1. The required value can be found via "nodetool status".
 - cassandraUser <username> - optional, but required if -cassandraPassword is set in `cassandra.yaml`
 - cassandraPassword <password> - optional, but required if -cassandraUser is set in `cassandra.yaml`
 - useSSL - optional, but required if cassandra is SSL enabled in `cassandra.yaml`
 - truststore <truststore> - optional, but required if -truststore is set in `cassandra.yaml`
 - storePassword <password> - optional, but required if -storePassword is set in `cassandra.yaml`

Example for SSL enabled cassandra: `java -jar upgrade.jar -cassandraHost <hostname> -datacenter DC1 -useSSL -truststore <truststore> -storePassword <password>`

Example for non-SSL enabled cassandra: `java -jar upgrade.jar -cassandraHost <hostname> -datacenter DC1`

Earlier versions of RWS communicated with Cassandra 2 using the thrift protocol. Cassandra has discontinued support for the thrift protocol, so the new version of RWS uses CQL. The upgrade application will make schema changes to the `call_recording_by_contact_center` table and all call/screen recording and log tables referenced by this table so that the data inserted via thrift will be available via CQL.

3. Perform Cassandra upgrade according to the [Cassandra Upgrade Guide](#).
4. When configuring Cassandra 4.1, you must set the following both parameters in the `cassandra.yaml` file:
 - **batch_size_warn_threshold** to 200KiB
 - **batch_size_fail_threshold** to 250KiB
 - Save and apply all the changes.
 These parameters are related to the size of CQL batches. The default for these parameters is quite

low and need to be increased to accommodate RWS inserts and update batches.

5. Upgrade the Interaction Recording Web Service nodes to the IP version 8.5.500.17 (or higher).
6. Configure Cassandra 4.1 parameters in **application.yaml** of Interaction Recording Web Service nodes. For more information, see [Configuring Interaction Recording Web Services](#).
7. Start all Interaction Recording Web Service nodes. Validate the health of RWS and Cassandra.
8. After the upgrade,
 - If using Recording Processor Script(RPS), run the **LVR Recovery Script** to recover recordings from the each **Recording Processor Script failed** folder and repost the recordings to Interaction Recording Web Services, as failed voice recordings accumulate while the Cassandra cluster is unavailable.
 - If using GIR Voice Processor (GIR VP), the GIR Voice Processor automatically retries the recording posts for up to 40 days.

Rollback:

If we are seeing unresolved errors at any phase of the migration, please rollback back the Interaction Recording Web Service nodes to the previous RWS version and previous existing Cassandra 2.2 cluster until the issue is resolved.

Limitation:

If rollback is performed once the migration procedure is done and new voice recordings are directed to Cassandra 4.1, the following scenarios will result in errors :

- Playback of the new recordings
- Backup and purge of the new recordings
- Querying of the new recordings through API.

Configuring WebDAV

Interaction Recording Web Services relies on a Web Distributed Authoring and Versioning (WebDAV) server to store and manage the GIR recording files. WebDAV is an extension of the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on World Wide Web servers. A working group of the Internet Engineering Task Force (IETF) defined WebDAV in RFC 4918.

The following information represents examples of what can be done for WebDAV. Follow these procedures to get a better understanding of what needs to be done when you use a Red Hat Enterprise Linux machine with the Apache HTTP Server.

Important

- This document provides you with basic guidelines on configuring WebDAV on RHEL. If you wish to configure WebDAV on other operating systems or if you have additional questions regarding WebDAV on RHEL, refer to the official documentation from the operating system provider.
- It is recommended that you do not install WebDAV on the same machine as Interaction Recording Web Services (RWS), since numerous deployments already install Cassandra and Elasticsearch on the same host. These are critical components for the operation of RWS. If an additional process such as WebDAV is run on the same machine as RWS, disk I/O operations will be limited and the stability of RWS may be negatively impacted.
- Authentication must be configured on the WebDAV server. This is required to ensure proper storage and management of GIR recording files.

Deploying the WebDAV Server

1. Install Apache HTTP Server and run the following command:

```
yum install httpd
```

2. Edit the **/etc/httpd/conf/httpd.conf** file, and append the following to the end of the file:

```
Alias /recordings /mnt/recordings
<Directory /mnt/recordings>
    Options Indexes MultiViews FollowSymLinks
    EnableSendfile off
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
<Location "/recordings">
```

```
DAV On
AuthType Basic
AuthName "user"
AuthUserFile /var/www/htpasswd
Require valid-user
</Location>
```

3. Open the firewall.

Because Apache HTTP Server is an HTTP server, the incoming default HTTP and/or HTTPS ports (80 and/or 443) must be open to the server.

Important

It is possible to use custom ports by changing the permitted incoming ports in the firewall, the virtual host configuration file, and the URL used to reach the Apache HTTP server.

4. Create the directory to keep the recording files, and set the permission to apache, using the following commands:

```
mkdir /mnt/recordings
chown apache:apache /mnt/recordings
chcon -R -t httpd_sys_content_t /mnt/recordings
```

Important

Due to performance concerns, Genesys does not recommend using a remote directory for WebDAV.

5. Create an Apache HTTP Server user for httpd, and configure the password. The following example creates a user called "user":

```
htpasswd -cm /var/www/htpasswd user
```

Warning

If the Recording Muxer is deployed for screen recording, make sure all WebDAV storages of the same contact center region are using the same username and password.

6. Configure the httpd to start on boot up (and start it now) using the following commands:

```
chkconfig --levels 235 httpd on
service httpd start
```

7. Test the Apache HTTP Server installation:

a. Upload a hello.world file to the Apache HTTP Server using the following command:

```
curl -T hello.world -u user:password http://myserver/recordings/hello.world
```

- b. Using a browser, open the `http://myserver/recordings/hello.world` URL. The browser will request for user credentials.

8. The Apache HTTP Server is installed.

Configuring TLS for the WebDAV Server

To configure TLS for the Apache HTTP Server on RHEL6:

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: `/etc/pki/tls/certs/localhost.crt`
 - Key: `/etc/pki/tls/private/localhost.key`
2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **`/etc/httpd/conf.d/ssl.conf`** file and modify the following lines:
 - `SSLCertificateFile /etc/pki/tls/certs/localhost.crt`
 - `SSLCertificateKeyFile /etc/pki/tls/private/localhost.key`
 3. Restart httpd by running the following command:

```
service httpd restart
```

TLS is enabled on the default HTTPS port 443.

Important

If you're using a self-signed certificate and migrating from Web Services to Interaction Recording Web Services, you'll need to complete a few more steps. See [Re-importing the Certificate for WebDAV](#) for details.

Changing storage location

You can use any one of the following methods to expand the available storage or to migrate the recording files to a new WebDAV server.

Method 1

1. Leave the existing WebDAV server in place and point the Storage Destination in IVR Profile to the new

WebDAV server. Ensure that the recordings storage destination folder in new WebDAV server has write access.

2. Configure the new WebDAV storage path in RWS Storage settings for voice recordings. For details, see [Configure the Storage Credentials for Interaction Web Services](#).
3. If the tenant has Screen Recordings enabled, configure the new WebDAV storage path in Screen Recording storage settings and set the active property to true. For more details, see [Screen Recording Storage Settings](#).
4. Update the active property to false for the existing WebDAV storage path in Screen Recording storage settings.

Important

If the WebDAV servers are load balanced using the Load Balancer, add the new WebDAV servers to the load balancer as a separate balanced address and follow Steps 1 to 4. Retain the existing WebDAV load balancer configuration to ensure that old recordings are still accessible.

Method 2

1. Copy all the existing recording files to the new WebDAV server and make sure the file path is maintained as in the old WebDAV server.
2. Make the necessary changes to the new WebDAV server to take over the IP or FQDN of the old WebDAV server.

Next Step

- [Initialize Cassandra](#)

Initializing Cassandra

Make sure you have [installed and tested Cassandra](#) before completing the procedures below.

Creating the Cassandra Keyspace

The procedures below describe how to create the Cassandra keyspace for the following scenarios:

- Development—one Cassandra node (appropriate for a development or lab environment)
- Single Datacenter—one datacenter with a minimum of three Cassandra nodes
- Two Datacenters—two datacenters with a minimum of three Cassandra nodes in each datacenter

Important

For more complex Cassandra deployments, please consult with Genesys.

Select a tab below for the procedure that matches your deployment scenario.

Development

Creating the Cassandra Keyspace (1 Cassandra Node)

Start

1. Copy the **ks-schema-local.cql** file from **installation_CD/data** to the Cassandra node host.
2. Set the replication factor to 1 (the default), if needed. Since this is a single node deployment, you don't need to modify this value. Refer to the [Cassandra documentation](#) for more information about replication factors.

```
replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}
```

3. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f ks-schema-local.cql
```

...where *cassandra_host* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Single Datacenter

Creating the Cassandra Keyspace (1 Datacenter)

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **ks-schema-prod.cql** file from *installation_CD/data* to the Cassandra node host.
2. For fault tolerance, Genesys recommends that you use at least 3 Cassandra nodes and set the replication factor to 3. Refer to the [Cassandra documentation](#) for more information about replication factors. To modify this value, change the following line:

```
replication = {'class': 'SimpleStrategy', 'replication_factor': '<replication-factor-in-your-environment>'}
```

3. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f ks-schema-prod.cql
```

...where *cassandra_host* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Two Datacenters

Creating the Cassandra Keyspace (2 Datacenters)

Complete the following procedure on one node in your Cassandra cluster.

Start

1. Copy the **ks-schema-prod_HA.cql** file from *installation_CD/data* to the Cassandra node host.
2. Modify the following content:

```
replication = {'class': 'NetworkTopologyStrategy', 'AZ1': '3', 'AZ2': '3'}
```

- a. Add the datacenter name. You can use nodetool to find the name of the datacenter by examining

the output of `nodetool status` (the tool is located in the `bin` directory of Cassandra). The following is sample output from the `nodetool`:

```
/genesys/apache-cassandra-2.2/bin$ ./nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns (effective) \
UN 192.0.2.10    4.58 MB    256        100.0%           \
UN 192.0.2.11    2.3 MB     256        100.0%           \
UN 192.0.2.12    4.11 MB    256        100.0%           \
                               Host ID      Rack
                               dab220f6-7744-4709-b2ce-d18629076a76 rack1
                               922a3442-63f9-43f7-af08-2cd62f02e28b rack1
                               913f77c3-7dc2-4d93-b643-9e0c514314d1 rack1
Datacenter: datacenter2
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns (effective) \
UN 198.51.100.10 4.16 MB    256        100.0%           \
UN 198.51.100.11 2.24 MB    256        100.0%           \
UN 198.51.100.12 4.19 MB    256        100.0%           \
                               Host ID      Rack
                               cd92c658-176a-453b-b118-9b952f78f237 rack1
                               c4afb92-59c8-450f-b9b1-79b3454c04a2 rack1
                               d6fd07b4-8f6c-487e-a574-43d6f5980ac8 rack1
```

- b. Add the replication factor. Refer to the [Cassandra documentation](#) for more information about replication factors.

Based on the `nodetool` output above, your line might be:

```
replication = {'class': 'NetworkTopologyStrategy', 'datacenter1': '3', 'datacenter2': '3'}
```

3. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f ks-schema-prod_HA.cql
```

...where *cassandra_host* is the host name (fully qualified domain name) or IP address of the Cassandra node.

End

Creating the Column Families

Complete the following procedure on one node in your Cassandra cluster.

Cassandra 4

Start

1. Copy the **cf-schema-c4-8.5.500.30.cql** file from **installation_CD/data** to the Cassandra node host.
2. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f cf-schema-c4-8.5.500.30.cql  
...where cassandra_host is the host name (fully qualified domain name) or IP address of the  
Cassandra node
```

End

Cassandra 2

Start

1. Copy the **cf-schema.cql** file from **installation_CD/data** to the Cassandra node host.
2. Run the following command to create the Cassandra schema:

```
cassandra_install_dir/bin/cqlsh cassandra_host -f cf-schema.cql  
...where cassandra_host is the host name (fully qualified domain name) or IP address of the  
Cassandra node
```

End

Next Step

- [Elasticsearch](#)

Elasticsearch

Elasticsearch 1.x (deprecated)

Interaction Recording Web Services uses [Elasticsearch](#) — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that is separate from your Interaction Recording Web Services nodes. See [Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster](#) for details. It's possible to set up a co-located Elasticsearch cluster, which means that Elasticsearch is included in your Interaction Recording Web Services nodes. See [Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster](#) for details.

Important

- If you are using GIR with Workspace Web Edition, a shared deployment of Elasticsearch should be used. Ensure you also review the Web Services and Applications documentation for Elasticsearch. For details see: [Elasticsearch](#).
- The Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 1.x version of Elasticsearch](#).

Prerequisites

Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 1.x version of Elasticsearch. Refer to the [Elasticsearch documentation](#) for details. Note that the Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Complete the following steps for each Elasticsearch node

1. Copy the **elasticsearch.yml.sample** file from the **installation_CD/config-templates/** folder, to the Elasticsearch configuration folder on a standalone machine, and rename it to **elasticsearch.yml**. If you use **.rpm** for Elasticsearch, use **/etc/elasticsearch/** as the configuration folder. If you use the **gzipped tarball**, use **\$installDir/config**.
2. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
index.analysis.analyzer.whitespace_lowercase.tokenizer: whitespace
index.analysis.analyzer.whitespace_lowercase.filter: lowercase
index.analysis.analyzer.mediaPartitionAnalyzer.tokenizer: path_hierarchy
threadpool.index.queue_size: -1
threadpool.bulk.queue_size: -1
bootstrap.mlockall: true
indices.fielddata.cache.size: 75%
indices.breaker.fielddata.limit: 80%
path.conf: <Elasticsearch configuration path>
path.data: <Elasticsearch installation path>/esdata
node.name: ToBeChanged: <name of the Elasticsearch node. Set uniquely for each node>
cluster.name: ToBeChanged: <name of the Elasticsearch cluster>
transport.tcp.port: 9300
http.port: 9200
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ToBeChanged: <comma separated list of Elasticsearch nodes>
discovery.zen.minimum_master_nodes: ToBeChanged: <set to the minimum number of master nodes>
gateway.recover_after_nodes: ToBeChanged: <calculate based on the number of Elasticsearch nodes with rule: '<NUMBER_ES_NODES> / 2 + 1'>
gateway.recover_after_time: 1m
gateway.expected_nodes: ToBeChanged: <set to the number of Elasticsearch nodes>
```

3. Copy the **installation_CD/elasticsearch/templates** folder, along with its **.json** file contents, to a new templates folder under the configuration folder of Elasticsearch (for example, **/etc/elasticsearch/templates** if you use **.rpm** for Elasticsearch, or **\$installDir/config/templates** if you use the **gzipped tarball**) on each node.
4. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/current/file-descriptors.html>.

Important

The Elasticsearch engine requires a large Metaspace space. To increase the Metaspace space, pass the following argument to the JVM used to run Elasticsearch:

```
"-XX:MaxMetaspaceSize=512m"
```

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a co-located Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 1.x version of Elasticsearch](#).

Complete the following steps for each Interaction Recording Web Services node that you want to host Elasticsearch

1. Install Elasticsearch using the latest stable 1.x version of Elasticsearch.
2. Copy the **elasticsearch.yml.sample** file from the **installation_CD/config-templates/** folder, to the Elasticsearch configuration folder on the Interaction Recording Web Services node, and rename it to **elasticsearch.yml**. If you use **.rpm** for Elasticsearch, use **/etc/elasticsearch/** as the configuration folder. If you use the **gzipped tarball**, use **\$installDir/config**.
3. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file: **Note:** **<Elasticsearch installation path>** refers to the location on which Elasticsearch has been installed.

```
index.analysis.analyzer.whitespace_lowercase.tokenizer: whitespace
index.analysis.analyzer.whitespace_lowercase.filter: lowercase
index.analysis.analyzer.mediaPartitionAnalyzer.tokenizer: path_hierarchy
threadpool.index.queue_size: -1
threadpool.bulk.queue_size: -1
bootstrap.mlockall: true
indices.fielddata.cache.size: 75%
indices.breaker.fielddata.limit: 80%
path.conf: <Elasticsearch configuration path>
path.data: <Elasticsearch installation path>/esdata
node.name: ToBeChanged: <name of the Elasticsearch node. Set uniquely for each node>
cluster.name: ToBeChanged: <name of the Elasticsearch cluster>
transport.tcp.port: 9300
http.port: 9200
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ToBeChanged: <comma separated list of Elasticsearch
```



```
nodes>
discovery.zen.minimum_master_nodes: ToBeChanged: <set to the minimum number of master
nodes>
gateway.recover_after_nodes: ToBeChanged: <calculate based on the number of
Elasticsearch nodes with rule: '<NUMBER_ES_NODES> / 2 + 1'>
gateway.recover_after_time: 1m
gateway.expected_nodes: ToBeChanged: <set to the number of Elasticsearch nodes>
```

Important

Do not forget to update **<Elasticsearch installation path>** to the appropriate value.

4. Copy the **installation_CD/elasticsearch/templates** folder, along with its **.json** file contents, to a new templates folder under the configuration folder of Elasticsearch (for example, **/etc/elasticsearch/templates** if you use **.rpm** for Elasticsearch, or **\$installDir/config/templates** if you use the **gzipped tarball**) on each node.
5. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/current/file-descriptors.html>.
6. Set the **crClusterName** option in the **application.yaml** file to the name of the cluster, as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
7. Set the **elasticSearchSettings** option in the **application.yaml** file to the appropriate values for your environment.

Important

The Elasticsearch engine requires a large Metaspace space. To increase the Metaspace space, pass the following argument to the JVM used to run Elasticsearch: **"-XX:MaxMetaspaceSize=512m"**

Migrating an Existing Elasticsearch Deployment to Schema V2

Important

The following procedure should only be performed once per contact center (that is, for each contact center tenant in a multi-tenant deployment). This procedure should not be performed for a new GIR installation.

Perform the following steps while your system is running, without service interruption.

1. Copy the **call_recordingv2_template.json** and **screen_recordingv2_template.json** files from the **installation_CD/elasticsearch/templates/** folder to the **templates** folder in each node in your Elasticsearch cluster.

2. Perform a rolling restart of each node in your Elasticsearch cluster. Stop and restart each node and wait until it is restarted and is operational before stopping and restarting the next node.
3. Prepare a new dedicated **Interaction Recording Web Services** node as follows:
 - a. Install Interaction Recording Web Services in the same way a regular **Interaction Recording Web Services** node is installed. Do not add this node to the Interaction Recording Web Services Load Balancer.
 - b. Edit the Interaction Recording Web Services **application.yaml** file, by adding the following configuration. Verify that you add lines under nodes for all the existing Interaction Recording Web Services nodes in your deployment:

```
elasticSearchSettings:
  useTransportClient: true
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: true
  ignoreClusterName: true
  pingTimeout: 10000
  nodesSamplerInterval: 10000
  enableIndexVerificationAtStartup: false
  indexPerContactCenter: true
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

3. Increase the **Hystrix** timeout for **RecordingOperationApiTaskV2** on the new Interaction Recording Web Services node by adding the following line to the Hystrix configuration:

```
hystrix.command.RecordingOperationApiTaskV2.execution.isolation.thread.timeoutInMilliseconds=<max time acceptable in milliseconds>
```

Important

<max time acceptable in milliseconds> should exceed the time that the re-indexing operation is expected to take. This value varies depending on how you elect to divide the re-indexing iterations. If you expect each re-indexing operation to take approximately one hour, then set this parameter to a value such as 7200000.

4. Determine the Contact Center ID using the following command:

```
curl -u <ops-user>:<ops-pass> http://< selected-web-services-node>: <web-services-listening-port>/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://< selected-web-services-node>: <web-services-listening-port>/api/v2/ops/contact-centers/<contact-center-id>"]}
```

Migrate Call Recording Index

1. Start the migration process for call recording by issuing the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json" "http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/features" -d '{ "uris":["schema-elasticsearch-migration-to-v2-call-recording"] }'
```

2. Immediately after performing step #1, note the current time and initialize the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json" "http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/recordings" -d '{ "operationName":"forceIndex", "from": <start-range-in-milliseconds>, "to": <stop-range-in-milliseconds>, "purgeOld":<value> }'
```

Genesys recommends to perform re-indexing in multiple iterations depending on how many records exist in Cassandra. The key aspect when determining "**from**" and "**to**" values is to use these parameters to specify the number of records to be re-indexed at a time. A reasonable estimate for the time taken to re-index can be 5,000,000 records in one hour, although this is dependent on your Cassandra and Elasticsearch deployment. Therefore, depending on the number of records in your deployment, this could be accomplished with a single iteration, where the "**from**" and "**to**" values specified cover the entire time range of content within Cassandra.

Important

purgeOld is set to **true** initially, and to **false** for all subsequent invocations.

3. Repeat the command from step #2 varying the “from” and “to” values to completely cover all recordings that exist (up to and including the time noted at the beginning of step 2), so that they are included in the new index.
For each iteration, ensure that **purgeOld** is set to **false** so that the newly created index is not removed.
4. Once the **forceIndex** commands are completed (so that the entire set of recordings have been re-indexed), configure Interaction Recording Web Services to use the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-v2-call-recording"]
}'
```

5. Verify that the Search functionality is working properly using the **GetRecordings API**. For additional information refer to [Genesys Interaction Recording API](#).

Important

SpeechMiner cannot be used to perform this validation since it uses a different mechanism to search for call recordings.

Once this procedure is completed both the old index and the new index are maintained and the new index is used for all searches.

Migrate Screen Recording Index

1. Start the migration process for screen recording by issuing the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-screen-recording"]
}'
```

2. Immediately after performing step #1, note the current time and initialize the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/screen-recordings" -d '{
  "operationName":"forceIndex",
  "from": <start-range-in-milliseconds>,'
```

```
"to": <stop-range-in-milliseconds>,  
"purgeOld":<value>  
'}
```

Genesys recommends to perform re-indexing in multiple iterations depending on how many records exist in Cassandra. The key aspect when determining "**from**" and "**to**" values is to use these parameters to specify the number of records to be re-indexed at a time. A reasonable estimate for the time taken to re-index can be 5,000,000 records in one hour, although this is dependent on your Cassandra and Elasticsearch deployment. Therefore, depending on the number of records in your deployment, this could be accomplished with a single iteration, where the "**from**" and "**to**" values specified cover the entire time range of content within Cassandra.

Important

purgeOld is set to **true** initially, and to **false** for all subsequent invocations.

3. Repeat the command from step #2 varying the "from" and "to" values to completely cover all recordings that exist (up to and including the time noted at the beginning of step 2), so that they are included in the new index.
For each iteration, ensure that **purgeOld** is set to **false** so that the newly created index is not removed.
4. Once the **forceIndex** commands are completed (so that the entire set of recordings have been re-indexed), configure Interaction Recording Web Services to use the new index by using the following command:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json"  
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-  
center-id>/features" -d '{  
  "uris":["schema-elasticsearch-v2-screen-recording"]  
'}
```

5. Verify that the Search functionality is working properly against the full range of screen recordings, by using the SpeechMiner Screen Recording grid.

Once this procedure is completed both the old index and the new index are maintained and the new index is used for all searches.

Completing the Migration

Once you have migrated both the Call Recording and Screen Recording indexes, both the old index and the new index are updated for every new recording. This process consumes additional disk space. To avoid the use of additional disk space, perform the following steps to remove the old indexes once testing has confirmed that the new indexes are fully operational:

Important

Once the following steps are performed, it will not be possible to roll back the migration.

1. Turn off the schema migration feature flag for the index being migrated, by using the following command:

For **Call Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-call-recording"]
}'
```

For **Screen Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-screen-recording"]
}'
```

2. Delete the old indexes by using the following command:

```
curl -XDELETE http://<es-node>:9200/<index-name>
```

Where:

- **<es-node>** is one of the Elasticsearch nodes in the cluster.
- **<index-name>** is the index name for the original schema:
 - **{contact-center-id}** for call recording when an embedded Elasticsearch cluster is used, or **call-recording-{contact-center-id}** for a standalone Elasticsearch cluster deployment. For example, **f3eec6cb-f624-4ac2-975e-6a60e0ebf878** or **call-recording-f3eec6cb-f624-4ac2-975e-6a60e0ebf878**.
 - **screen-recording-{contact-center-id}** for screen recording. For example, **screen-recording-f3eec6cb-f624-4ac2-975e-6a60e0ebf878**.

If you are unsure, the index names in use on Elasticsearch can be determined by using the following command (where **<es-node>** is one of the Elasticsearch nodes in the cluster):

```
curl -XGET http://<es-node>:9200/_cat/indices?v
```

Important

At this point the new Interaction Recording Services node that was used for these migration steps is no longer required and can be shut down or re-purposed.

Rolling Back the Migration

In the event of a problem with the index migration, perform the following steps to implement the old (previous) index and remove the new index.

1. If the new index was enabled as the default index, run the following command to use the old index:

For **Call Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-v2-call-recording"]
}'
```

For **Screen Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-v2-screen-recording"]
}'
```

2. Stop updates to the new index by turning off the schema migration feature flag, using the following command:

For **Call Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-
center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-call-recording"]
}'
```

For **Screen Recordings**:

```
curl -u <ops-user>:<ops-pass> -XDELETE -H "Content-Type:application/json"
```

```
"http://<selected-RWS-node>:<RWS-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/features" -d '{
  "uris":["schema-elasticsearch-migration-to-v2-screen-recording"]}
}'
```

3. To delete the new index, run the following command:

```
curl -XDELETE http://<es-node>:9200/<index-name>
```

Where:

- **<es-node>** is one of the Elasticsearch nodes in the cluster.
- **<index-name>** is the index name for the new schema:
 - **call-recording-v2-{region}-{contact-center-id}** for call recordings.
 - **screen-recording-v2-{region}-{contact-center-id}** for screen recordings.
 - **region** is the value of the **crRegion** parameter specified in the **application.yaml** file in the Interaction Recording Web Services node that was used to perform the index migration process for both the call recording index and the screen recording index.

Elasticsearch 7.16.3

Interaction Recording Web Services uses **Elasticsearch** — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster](#). You can also set up a co-located Elasticsearch cluster, which means that Elasticsearch is included in your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster](#).

Important

- If you are using GIR with Workspace Web Edition, refrain from using a shared deployment of Elasticsearch. This is because Web Services and Applications support Elasticsearch 1.x only and do not support ES 7.16.3. For details see: [Elasticsearch](#). This is applicable if you are installing Web Services and Applications version 8.5.201.09 or earlier.
- The Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 7.16.3 version of Elasticsearch](#).

Prerequisites

- Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 7.16.3 version of Elasticsearch. Refer to the [Elasticsearch documentation](#) for details. Note that the Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.
- Interaction Recording Web Services deployment version should be 8.5.204.16 or higher.
- Interaction Recording Web Services supported Elasticsearch 7.16.3 installed on RedHat 8/9 and Java 11.

Limitations of Elasticsearch 7.16.3

- Elasticsearch 7.16.3 only supports schema V3.
- Genesys is not responsible for migration of existing data to the latest version of Elasticsearch on premise environments.
- Interaction Recording Web Services does not support scan and scroll functionality on Elasticsearch 7.16.3.

Complete the following steps for each Elasticsearch node

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 7.16.3.
2. Open the following **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
```

```
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.16/file-descriptors.html>.

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a co-located Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone as co-located is discontinued after RWS version 8.5.205.69. For additional information, refer to the **latest stable 7.16.3 version of Elasticsearch**.

Complete the following steps for each Interaction Recording Web Services node that you want to host Elasticsearch

1. Refer to the **Elasticsearch documentation** for installing and configuring Elasticsearch 7.16.3.
2. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
```

```
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.16/file-descriptors.html>.
4. Set the **crClusterName** option in the **application.yaml** file to the name of the cluster, as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
5. Set the **elasticSearchSettings** option in the **application.yaml** file to the appropriate values for your environment.

Perform the following steps in each Elasticsearch node while your system is running, without service interruption

1. Create call recording schema V3:
 - a. Copy the **call_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
 - b. Create call-recording schema V3 using the following command:


```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/call-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@call_recording_v3_template.json
```
3. Create screen recording schema V3:
 - a. Copy the **screen_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
 - b. Create screen-recording V3 schema using the following command:


```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/screen-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@screen_recording_v3_template.json
```

Configure Interaction Recording Web Services using Elasticsearch 7.16.3

1. Update the **application.yaml** file on each Interaction Recording Web Services node.
2. update **useTransportClient** to be "false".
3. Add a new property **useRestClient** as follows:

```
elasticSearchSettings:
  useTransportClient: false
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: false
  ignoreClusterName: false
  pingTimeout: 5000
  nodesSamplerInterval: 5000
  useRestClient: true
```

```
restClient:
  nodes:
    - {host: <elastic-search-node1>, port: 9200}
    - {host: <elastic-search-node2>, port: 9200}
    - {host: <elastic-search-node3>, port: 9200}
waitToIndexTimeout: 5000
scanReadTimeoutSeconds: 60
scrollTimeoutSeconds: 240
countReadTimeoutSeconds: 60
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

Elasticsearch 7.17.15

Interaction Recording Web Services uses **Elasticsearch** — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster](#). You can also set up a co-located Elasticsearch cluster, which means that Elasticsearch is included in your Interaction Recording Web Services nodes. For more details, see [Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster](#).

Important

- If you are using GIR with Workspace Web Edition, refrain from using a shared deployment of Elasticsearch. This is because Web Services and Applications support Elasticsearch 1.x only and do not support ES 7.17.15. For details see: [Elasticsearch](#). This is applicable if you are installing Web Services and Applications version 8.5.201.09 or earlier.
- The Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster

by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the [latest stable 7.17.15 version of Elasticsearch](#).

Prerequisites

- Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 7.17.15 version of Elasticsearch. Refer to the [Elasticsearch documentation](#) for details. Note that the Elasticsearch deployment used by SpeechMiner cannot be used with Interaction Recording Web Services.
- Interaction Recording Web Services deployment version should be 8.5.205.32 or higher.
- Interaction Recording Web Services supported Elasticsearch 7.17.15 installed on RedHat 8/9 and Java 11.

Limitations of Elasticsearch 7.17.15

- Elasticsearch 7.17.15 only supports schema V3.
- Genesys is not responsible for migration of existing data to the latest version of Elasticsearch on premise environments.

Complete the following steps for each Elasticsearch node

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 7.17.15.
2. Open the following **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/file-descriptors.html>.

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Configuring Interaction Recording Web Services to Use a Co-located Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a co-located Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone as co-located is discontinued after RWS version 8.5.205.69. For additional information, refer to the [latest stable 7.17.15 version of Elasticsearch](#).

Complete the following steps for each Interaction Recording Web Services node that you want to host Elasticsearch

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 7.17.15.
2. Open the **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/7.17/file-descriptors.html>.
4. Set the **crClusterName** option in the **application.yaml** file to the name of the cluster, as specified by **cluster.name** in the **elasticsearch.yml** configuration file.

- Set the **elasticSearchSettings** option in the **application.yaml** file to the appropriate values for your environment.

Perform the following steps in each Elasticsearch node while your system is running, without service interruption

- Create call recording schema V3:

- Copy the **call_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- Create call-recording schema V3 using the following command:

```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/call-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@call_recording_v3_template.json
```

- Create screen recording schema V3:

- Copy the **screen_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- Create screen-recording V3 schema using the following command:

```
$ curl -XPUT http://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP port>/_template/screen-recording-v3-template?include_type_name=false -H "Content-Type:application/json" -d@screen_recording_v3_template.json
```

Configure Interaction Recording Web Services using Elasticsearch 7.17.15

- Update the **application.yaml** file on each Interaction Recording Web Services node.
- update **useTransportClient** to be "false".
- Add a new property **useRestClient** as follows:

```
elasticSearchSettings:
  useTransportClient: false
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: false
  ignoreClusterName: false
  pingTimeout: 5000
  nodesSamplerInterval: 5000
  useRestClient: true
  restClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9200}
      - {host: <elastic-search-node2>, port: 9200}
      - {host: <elastic-search-node3>, port: 9200}
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: 60
  scrollTimeoutSeconds: 240
  countReadTimeoutSeconds: 60
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

Elasticsearch 8.18.2

Interaction Recording Web Services uses **Elasticsearch** — an open-source, full-text search engine with a RESTful web interface — to index recording metadata.

For new deployments, Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Interaction Recording Web Services nodes. For more details, see **Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster** section below.

Important

Genesys recommends that you set up a cluster of Elasticsearch nodes that are separate from your Workspace Web Edition deployment and SpeechMiner deployment.

Configuring Interaction Recording Web Services to Use a Standalone Elasticsearch Cluster

You can configure Interaction Recording Web Services to work with a standalone Elasticsearch cluster by completing the steps below.

Important

Contact your Genesys representative for information about how to migrate from co-located to standalone. For additional information, refer to the **latest stable 8.18.2 version of Elasticsearch**.

Prerequisites

- Verify that a cluster of Elasticsearch nodes have been deployed and configured using the latest stable 8.18.2 version of Elasticsearch. Refer to the **Elasticsearch documentation** for details.

- Interaction Recording Web Services deployment version should be 8.5.500.24 or higher.
- Interaction Recording Web Services supported Elasticsearch 8.18.2 installed on RedHat 8/9 and Java 17.

Limitations of Elasticsearch 8.18.2

- Elasticsearch 8.18.2 only supports schema V3.
- Genesys is not responsible for migration of existing data to the latest version of Elasticsearch on premise environments.

Complete the following steps for each Elasticsearch node

1. Refer to the [Elasticsearch documentation](#) for installing and configuring Elasticsearch 8.18.2.
2. Open the following **elasticsearch.yml** configuration file in a text editor and verify that the following lines are included in the file:

```
cluster.name: <Use a descriptive name for your cluster>
node.name: <Use a descriptive name for the node>
node.attr.rack: <Add custom attributes to the node>
path.data: <Path to directory where to store the data (separate multiple locations by comma)>
path.logs: <Path to log files>
bootstrap.memory_lock: <Lock the memory on startup>
network.host: <By default, Elasticsearch is only accessible on localhost. Set a different address here to expose this node on the network>
http.port: <Set a specific HTTP port here, by default is 9200>
discovery.seed_hosts: <Pass an initial list of hosts to perform discovery when this node is started>
cluster.initial_master_nodes: <Bootstrap the cluster using an initial set of master-eligible nodes>
action.destructive_requires_name: <Requires explicit names when deleting indices>
# Security settings
xpack.security.enabled: [ToBeChangedOrRemoved: true|false]
xpack.security.enrollment.enabled: [ToBeChangedOrRemoved: true|false]
xpack.security.http.ssl:
  enabled: [ToBeChangedOrRemoved: true|false]
  keystore.path: [ToBeChangedOrRemoved: <KEYSTORE_PATH>]
xpack.security.transport.ssl:
  enabled: [ToBeChangedOrRemoved: true|false]
  verification_mode: [ToBeChangedOrRemoved: certificate]
  keystore.path: [ToBeChangedOrRemoved: <KEYSTORE_PATH>]
  truststore.path: [ToBeChangedOrRemoved: <TRUSTSTORE_PATH>]

indices.id_field_data.enabled: true
```

3. Increase the permitted number of open file descriptors for the operating system on the node by referring to the documentation at <https://www.elastic.co/guide/en/elasticsearch/reference/8.18/file-descriptors.html>.

Complete the following steps for each Interaction Recording Web Services node

Configure the **application.yaml** file as follows:

1. Set the **crClusterName** option to the name of the cluster as specified by **cluster.name** in the **elasticsearch.yml** configuration file.
2. Set the **elasticSearchSettings** option to appropriate values for your environment.

Perform the following steps in each Elasticsearch node while your system is running, without service interruption

1. Create call recording schema V3:

- a. Copy the **call_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- b. Create call-recording schema V3 using the following command:

```
$ curl -XPUT -u <USERNAME>:<PASSWORD> -H "Content-Type: application/json"
-d@<call_recording_v3_template.json> --cacert <PATH_TO_CA_CERT>
https://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP
port>/_template/call-recording-v3-template
```

3. Create screen recording schema V3:

- a. Copy the **screen_recording_v3_template.json** file from the **installation_CD/elasticsearch/templates/** folder to the **local temp** folder.
- b. Create screen-recording V3 schema using the following command:

```
$ curl -XPUT -u <USERNAME>:<PASSWORD> -H "Content-Type: application/json"
-d@<screen_recording_v3_template.json> --cacert <PATH_TO_CA_CERT>
https://<Elasticsearch Instance hostname>:<Elasticsearch Instance HTTP
port>/_template/screen-recording-v3-template
```

Configure Interaction Recording Web Services using Elasticsearch 8.18.2

1. Update the **application.yaml** file on each Interaction Recording Web Services node.
2. update **useTransportClient** to be "false".
3. Add a new property **useRestClient** as follows:

```
elasticSearchSettings:
  retriesOnConflict: 3
  useRestClient: true
  restClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9200}
      - {host: <elastic-search-node2>, port: 9200}
      - {host: <elastic-search-node3>, port: 9200}
  authentication:
    type: [ToBeChangedOrRemoved: none|basic]
    userName: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_USER_NAME>]
    password: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_PASSWORD>]
    useSSL: [ToBeChangedOrRemoved: true|false]
    truststore: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PATH>]
    truststorePassword: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PASSWORD>]
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: 60
  scrollTimeoutSeconds: 240
  countReadTimeoutSeconds: 60
```

Important

The **application.yaml** file on the Interaction Recording Web Services node must be set with the correct region information in the **crRegion** parameter within the **serverSettings** Call Recording section. This configuration is used as part of the new index name.

Next Step

- [Install Interaction Recording Web Services.](#)

Installing

To install Interaction Recording Web Services (RWS), first you need to set up the following two application objects it uses in the Genesys configuration environment:

- Cluster Application
- Node Application

If RWS is being deployed along with Web Services (GWS), then GWS must be installed first, and the Cluster Application created during the GWS installation is shared between both components.

Interaction Recording Web Services

Interaction Recording Web Services (RWS)

Creating the Application Templates

Using Genesys Administrator Extension, complete the steps below to create application templates to use for your IRWS_Cluster and IRWS_Node applications.

Start

1. To create the Genesys Generic Server template, navigate to **Configuration > Environment > Application Templates**.
2. Select **New...** and configure the properties of the template as shown below:
 - Name: IRWS_Cluster_Template
 - Type: Genesys Generic Server
 - Version: 8.5
 - State: Enabled
3. Click **Save & Close**.
4. To create the Genesys Generic Client template, select **New...** again and configure the properties of the template as shown below:
 - Name: IRWS_Node_Template
 - Type: Genesys Generic Client

- Version: 8.5
- State: Enabled

5. Click **Save & Close**.

End

Creating the IRWS Cluster Application

Start

1. Navigate to **Configuration > Environment > Applications** and click **New...**
2. In the **General** section, configure the properties of the application as shown below:
 - Name: IRWS_Cluster
 - Template: IRWS_Cluster_Template (this is the template you made in [Creating the Application Templates](#))
 - State: Enabled
 - Working Directory: .
 - Command Line: .
 - Command Line Arguments: .

Important

You need to add a "." to the Working Directory, Command Line, and Command Line Arguments fields, as shown above. These values are mandatory for all applications and must be entered to save the application object. Interaction Recording Web Services does not use these values, so the "." is used as a placeholder.

3. Choose a Host object. See [Create Host](#) in the *Management Framework Deployment Guide* for more information about Host objects.
4. Add the following connections:
 - Configuration Server (you can add CS Proxy using the [csproxy].proxy-writable=true option.)
 - [Interaction Server](#) (if supporting multimedia)
 - T-Server/SIP Server (when supporting voice)

Important

When working with dual data centers, RWS requires a connection to each Interaction Server Application Cluster (in both data centers), in order to provide support for disaster recovery with an Interaction Server between the data centers. Use the following application parameters to provide the connection between RWS and each Interaction Server Application Cluster:

- `siteName=DC1;clusterType=eservices #For DC1 Interaction Server`

Application cluster

- `siteName=DC2;clusterType=eservices` #For DC2 Interaction Server Application cluster

5. In the **Tenants** section, select a Tenant:

1. Click **Add**.
2. Choose the Environment tenant (or any other tenant that has a connection to your Configuration Server).
3. Click **OK**.

Important

This step is for adding a single tenant only. For information about multi-tenant deployments, see [Deploying Genesys Interaction Recording in a Multi-Tenant Deployment](#).

6. Add a default Listening Port:

1. Click **Add**.
2. Enter the application's Port. For instance 7000.
3. Click **OK**.

End

Creating the RWS Node Application

Start

1. Navigate to **Configuration > Environment > Applications** and click **New....**
2. In the **General** section, configure the properties of the application as shown below:
 - Name: IRWS_Node
 - Template: IRWS_Node_Template (this is the template you made in [Creating the Application Templates](#))
 - State: Enabled
3. Add the following connections:
 - Cluster application that was configured in the previous procedure.
4. Click **Save & Close**.

End

Interaction Recording Web Services with Web Services

Interaction Recording Web Services (RWS) with GWS

Creating the Application Template

Using Genesys Administrator Extension, complete the steps below to create an application template to use for your IRWS_Node applications.

Start

1. To create the Genesys Generic Client template, navigate to **Configuration > Environment > Application Templates**.
2. Select **New...** and configure the properties of the template as shown below:
 - Name: IRWS_Node_Template
 - Type: Genesys Generic Client
 - Version: 8.5
 - State: Enabled
3. Click **Save & Close**.

End

Creating the RWS Node Application

Start

1. Navigate to **Configuration > Environment > Applications** and click **New....**
2. In the **General** section, configure the properties of the application as shown below:
 - Name: IRWS_Node
 - Template: IRWS_Node_Template (this is the template you made in [Creating the Application Template](#))
 - State: Enabled
3. Add the following connections:
 - Cluster application that was configured as part of the GWS installation. For additional information, refer to the [Creating the Web Services \(WS\) Cluster Application](#) section.
4. Click **Save & Close**.

End

Next Step

- [Deploy the web application](#)

Deploying the Web Application

The final deployment step is to install Interaction Recording Web Services as a service. Complete the following steps for each Interaction Recording Web Services node.

Deploy Using Red Hat Enterprise Linux 8/9

Start

1. Create a new folder on your Interaction Recording Web Services node. For example, **ir-web-services**. This is the home folder for the web application.
2. Copy the **gir.jar** file from the installation CD to your new Interaction Recording Web Services home folder.
3. Create a new folder **/usr/lib/systemd/system/gir.service.d**
4. Copy the following files to the specified folders on your Interaction Recording Web services host:
For Red Hat Enterprise Linux 8/9
 - **installation_CD/rhel/usr/lib/systemd/system/gir.service.d/gir.conf** to the folder **/usr/lib/systemd/system/gir.service.d**
 - **installation_CD/rhel/usr/lib/systemd/system/gir.service** to the folder **/usr/lib/systemd/system**
5. Create a new folder **/usr/libexec/initscripts/legacy-actions/gir**
6. Copy the following files to the specified folders on your host:
For Red Hat Enterprise Linux 8/9
 - **installation_CD/rhel/usr/bin/gir** to the folder **/usr/bin**
 - **installation_CD/rhel/usr/libexec/initscripts/legacy-actions/gir/config** to the folder **/usr/libexec/initscripts/legacy-actions/gir**
 - **installation_CD/rhel/usr/libexec/initscripts/legacy-actions/gir/version** to the folder **/usr/libexec/initscripts/legacy-actions/gir**
7. Open **/usr/lib/systemd/system/gir.service.d/gir.conf** on your host and update the following environment variables to values appropriate for your Interaction Recording Web Services node:
 - **WorkingDirectory**—The Interaction Recording Web Services home folder you created in Step 1.
 - **Environment=GIR_TEMP**—The location where you want Interaction Recording Web Services to store temp files.
 - **Environment=GIR_CONF**—The location where you want to store the Interaction Recording Web Services configuration files. If you do not specify a value for **GIR_CONF**, Interaction Recording Web Services uses **WorkingDirectory/config**.

Important

If you are installing RWS 8.5.205.10, ensure that the environment variable `Environment=GIR_CONF` is either specified as **RWS home folder/config** or not specified.

8. Create the `GIR_CONF` folder you specified in Step 7. If you didn't set `GIR_CONF`, create a folder called **config** in **WorkingDirectory**— for example, **ir-web-services/config**.
9. Create the following configuration files in the folder you created in Step 8. You can simply copy the files from **installation_CD/config-templates** and remove the **.sample** extension. You'll learn more about the settings in these files as you go through the configuration steps for Interaction Recording Web Services and its features later in this guide.
 - **application.yaml**
 - **hystrix.properties**
 - **logback.xml**

Important

Ensure that only a single copy of the **application.yaml** file is deployed across all the GIR file locations described above.

10. Create the user group **gir**.
11. Create the **gir** user in the **gir** user group and provide the user with ownership, and read and write permissions for the following folders:
 - The folders defined in the `WorkingDirectory`, `GIR_TEMP`, and `GIR_CONF` environment variables.
 - The folder defined in the path configuration item within the logging section in the **application.yaml** file (**/var/log/jetty9** by default).
12. Set executable permissions on:
 - **/usr/bin/gir**
 - **/usr/libexec/initscripts/legacy-actions/gir/config**
 - **/usr/libexec/initscripts/legacy-actions/gir/version**
13. Use the following commands to register the new service on your host:

```
systemctl daemon-reload
systemctl enable gir.service
```

End

Next Step

- [Configuring Interaction Recording Web Services](#)

Configuring Interaction Recording Web Services

You'll need to update the **application.yaml** file on each of your **Interaction Recording Web Services** nodes to provide the basic configuration. You created this file (or Interaction Recording Web Services created it for you) as part of [Deploying the Web Application](#). In later topics, you'll learn more about modifying this file to configure additional [features](#) and [security](#). For now, review the contents below for details about each section in the **application.yaml** configuration file.

Important

When editing the **application.yaml** file, the values for the configuration options that are strings must be enclosed in double quotation marks in certain cases. Specifically:

- For string options only, the values YES, NO, ON, OFF, TRUE, FALSE (in upper or lower case) must be quoted.
- If the option is a boolean (true/false) option, then any of the values in the previous bullet can be used without quotes.
- Values that look like numbers but are treated as strings (for example; PINs, phone numbers, encryption keys), that begin with leading zeroes must be quoted.
- Avoid placing leading zeroes on numeric options; doing so will cause your option to be interpreted as an octal value.

For example, specifying `crRegion: NO` (indicating Norway) will be interpreted as `crRegion: FALSE`. Instead, this must be specified using double quotation marks `crRegion: "NO"`.

Logging Settings

The purpose of this section is to tell Interaction Recording Web Services where to find the **logback.xml** file you created (or Interaction Recording Web Services created for you) as part of [Deploying the Web Application](#) and where to save logs.

The **application.yaml.sample** file includes the following default logging section:

```
logging:
  config: logback.xml
  file: cloud.log
  path: /var/log/jetty9
```

See [logging](#) for details about all supported configuration settings for this section.

Jetty Settings

Since Jetty is embedded in Interaction Recording Web Services, you have to use the jetty section of the **application.yaml** file to tell Interaction Recording Web Services how Jetty should behave.

The **application.yaml.sample** file includes the following default jetty section:

```
jetty:
  host: [RWS_HOST]
  port: 8080
  idleTimeout: 30000
  soLingerTime: -1
  sessionMaxInactiveInterval: 1800
  enableWorkerName: true
  enableRequestLog: true
  requestLog:
    filename: yyyy_mm_dd.request.log
    filenameDateFormat: yyyy_MM_dd
    logTimeZone: GMT
    retainDays: 90
    append: true
    extended: true
    logCookies: true
    logLatency: true
    preferProxiedForAddress: true
  enableSsl: false
  ssl:
    port: 443
    securePort: 8443
    keyStorePath: [KEYSTORE_PATH]
    keyStorePassword: [KEYSTORE_PASSWORD]
    keyManagerPassword: [KEY_MANAGER_PASSWORD]
    trustStorePath: [TRUSTSTORE_PATH]
    trustStorePassword: [TRUSTSTORE_PASSWORD]
  httpOnly: true
  secure: false
  sessionCookieName: GIRJSESSIONID
```

See [jetty](#) for details about all supported configuration settings for this section.

Cassandra Cluster Settings

The settings in the **cassandraCluster** section tell Interaction Recording Web Services how your Cassandra cluster should be managed and accessed.

The **application.yaml.sample** file includes the following default **cassandraCluster** section:

```
cassandraCluster:
  thrift_port: 9160
  jmx_port: 7199
  keyspace: sipfs
  nodes: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>]
  backup_nodes: [ToBeChangedOrRemoved: <CASSANDRA_BACKUP_DC_NODES>]
  replication_factor: [ToBeChanged: <REPLICATION_FACTOR>]
  write_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
```

```

    read_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
    max_conns_per_host: 16
    max_cons: 48
    max_pending_conns_per_host: 80
    max_blocked_threads_per_host: 160

    cassandraVersion: 1.2
    useSSL: [ToBeChanged: "false" | "true"]
    truststore: [ToBeChanged: path to client truststore]
    truststorePassword: [ToBeChanged: truststore password]
    userName: [ToBeChangedOrRemoved: <CASSANDRA_USER_NAME>]
    password: [ToBeChangedOrRemoved: <CASSANDRA_USER_PASSWORD>]

```

The **application.yaml.sample** file includes the following default **cassandraCluster** section for Cassandra 4:

```

cassandraCluster:
  native_transport_port: 9042
  jmx_port: 7199
  keypace: sipfs
  nodes: [ToBeChanged: <CASSANDRA_PRIMARY_DC_NODES>]
  dataCenterName: [ToBeChanged: name of primary datacenter]
  backup_nodes: [ToBeChangedOrRemoved: <CASSANDRA_BACKUP_DC_NODES>]
  backupDataCenterName: [ToBeChangedOrRemoved: name of backup datacenter]
  switchOverUnHealthyNodes: [ToBeChangedOrRemoved: number of nodes down to prompt failover]
  replication_factor: [ToBeChanged: <REPLICATION_FACTOR>]
  write_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
  read_consistency_level: [ToBeChanged: "CL_LOCAL_QUORUM" for multi-datacenters env,
"CL_QUORUM" for single-DC env.]
  max_conns_per_host: 16
  max_cons: 48
  max_pending_conns_per_host: 80
  max_blocked_threads_per_host: 160

  cassandraVersion: 4.1.4
  useSSL: [ToBeChanged: "false" | "true"]
  truststore: [ToBeChanged: path to client truststore]
  truststorePassword: [ToBeChanged: truststore password]
  userName: [ToBeChangedOrRemoved: <CASSANDRA_USER_NAME>]
  password: [ToBeChangedOrRemoved: <CASSANDRA_USER_PASSWORD>]

```

Make sure you update all settings marked as [ToBeChanged]. See [cassandraCluster](#) for details about all supported configuration settings for this section.

Server Settings

The settings in the **serverSettings** section provide the core settings Interaction Recording Web Services needs to run your node.

The **application.yaml.sample** file includes the following default **serverSettings** section:

```
serverSettings:
  # URLs
  externalApiUrlV2: [ToBeChanged: public URL including protocol, address and port, <PUBLIC_SCHEMA_BASE_URL>]/api/v2
  internalApiUrlV2: [ToBeChanged: internal URL including protocol, address and port, <INTERNAL_SCHEMA_BASE_URL>]/internal-api
  undocumentedExternalApiUrl: [ToBeChanged: public URL including protocol, address and port, <PUBLIC_SCHEMA_BASE_URL>]/internal-api

  # Paths
  pathPrefix: [ToBeChangedOrRemoved: <PATH_PREFIX>]
  internalPathPrefix: [ToBeChangedOrRemoved: <INTERNAL_PATH_PREFIX>]

  # General
  temporaryAuthenticationTokenTTL: [ToBeChangedOrRemoved: <TEMPORARY_AUTHENTICATION_TOKEN_TTL>]
  enableCsrfProtection: false

  # Timeouts
  activationTimeout: 12000
  configServerActivationTimeout: 35000
  configServerConnectionTimeout: 15000
  connectionTimeout: 4000
  inactiveUserTimeout: 60
  reconnectAttempts: 1
  reconnectTimeout: 10000

  # OPS account
  opsUserName: [ToBeChanged: <OPS_USER_NAME>]
  opsUserPassword: [ToBeChanged: <OPS_USER_PASSWORD>]

  # CME credentials
  applicationName: [ToBeChanged: <CONFIG_SERVER_RWS_APPLICATION_NAME>]
  applicationType: CFGGenericClient
  cmeUserName: [ToBeChanged: <CONFIG_SERVER_USER_NAME>]
  cmePassword: [ToBeChanged: <CONFIG_SERVER_USER_PASSWORD>]
  syncNode: [ToBeChanged: "true"|"false"]

  # ConfigServer String Encoding
  configServerDefaultEncoding: windows-1252

  # Call Recording
  createCallRecordingCF: true
  crClusterName: [ToBeChanged: <NAME_OF_ES_CLUSTER>]
  crRegion: [ToBeChanged: <CR_REGION>]
  cryptoSecurityKey: [ToBeChanged: <CRYPTO_SECURITY_KEY>]
```

```
webDAVMaxConnection: 50
webDAVMaxTotalConnection: 500

# Multi regional supporting
nodePath: [ToBeChanged: node position in cluster, example: /<REGION>/HOST
nodeId: [ToBeChangedOrRemoved: unique value in cluster <NODE_ID>]

# SSL and CA
caCertificate: [ToBeChangedOrRemoved: <PATH_TO_CA_FILE>]
jksPassword: [ToBeChangedOrRemoved: <JKS_PASSWORD>]
webDAVTrustedCA: [ToBeChangedOrRemoved: "true" | "false" | <PATH_TO_CA_FILE>]
webDAVJksPassword: [ToBeChangedOrRemoved: <WEBDAV_JKS_PASSWORD>]
rcsTrustedCA: [ToBeChangedOrRemoved: "true" | "false" | <PATH_TO_CA_FILE>]
rcsJksPassword: [ToBeChangedOrRemoved: <RCS_JKS_PASSWORD>]
speechMinerTrustedCA: [ToBeChangedOrRemoved: "true" | "false" | <PATH_TO_CA_FILE>]
speechMinerJksPassword: [ToBeChangedOrRemoved: <SMIR_JKS_PASSWORD>]

# CORS
crossOriginSettings:
  allowedOrigins: [ToBeChangedOrRemoved: <CROSS_ALLOWED_ORIGINS>]
  allowedMethods: [ToBeChangedOrRemoved: <CROSS_ALLOWED_METHODS>]
  allowedHeaders: [ToBeChangedOrRemoved: <CROSS_ALLOWED_HEADERS>]
  allowCredentials: [ToBeChangedOrRemoved: <CROSS_ALLOW_CREDENTIALS>]
  corsFilterCacheTimeToLive: 120
  exposedHeaders: [ToBeChangedOrRemoved: <CROSS_EXPOSED_HEADERS>]

# Elasticsearch
elasticSearchSettings:
  retriesOnConflict: 3
  useTransportClient: true
  transportClient:
    nodes:
      - {host: [ToBeChanged: <ELASTIC_SEARCH_NODE1>], port: [ToBeChanged: <ELASTIC_SEARCH_PORT>]}
      - {host: [ToBeChanged: <ELASTIC_SEARCH_NODE2>], port: [ToBeChanged: <ELASTIC_SEARCH_PORT>]}
    useSniff: false
    ignoreClusterName: false
    pingTimeout: 5000
    nodesSamplerInterval: 5000
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCAN_READ_TIMEOUT_SECONDS>]
  countReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_COUNT_READ_TIMEOUT_SECONDS>]
  scrollTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCROLL_TIMEOUT_SECONDS>]
```

```
# Recording Settings
recordingSettings:
  auditLogDeletedFiles: [ToBeChangedOrRemoved: "true"|"false"]
  recordCryptoServerDecryptMaxConnection: 50
  recordCryptoServerDecryptMaxTotalConnection: 500
  recordCryptoServerDecryptSocketTimeout: 30000
  keySpaceNameSettingsCacheSecondsTTL: 300
  regionsSettingsCacheSecondsTTL: 300
  readOnlyRetryAfterSeconds: 1200

# Screen Recording
screenRecordingSettings:
  enableSameSiteCookieForScreenRecordingPlayback: [ToBeChangedOrRemoved: "true"|"false"]
  screenRecordingVoiceEnabled: [ToBeChangedOrRemoved: "true"|"false"]
  screenRecordingEServicesEnabled: [ToBeChangedOrRemoved: "true"|"false"]
  recordingInteractionEventsTTL: 172800
  clientSessionManagerCacheTTL: 60
  contactCenterInfoManagerCacheTTL: 90

# Caching Settings
cachingSettings:
  enableSystemWideCaching: [ToBeChangedOrRemoved: "true"|"false"]
  contactCenterFeaturesTTL: 30
  contactCenterSettingsTTL: 30

# Screen Recording Connections Reporting
screenRecordingConnectionReportingSettings:
  reportingEnabled: [ToBeChangedOrRemoved: "true"|"false"]
  createReportingCF: [ToBeChangedOrRemoved: "true"|"false"]
  connectionInfoHoursTTL: 168
  historyCountsMinutesTTL: 1440

# Multimedia Disaster Recovery
drMonitoringDelay: 1800

# DoS Filter Settings
enableDosFilter: [ToBeChanged: "true"|"false"]
dosFilterSettings:
  maxRequestsPerSec: 25
  delayMs: 100
  maxWaitMs: 50
  throttledRequests: 5
  throttleMs: 30000
```

```
maxRequestMs: 30000
maxIdleTrackerMs: 30000
insertHeaders: [ToBeChangedOrRemoved: <DOS_FILTER_INSERT_HEADERS>]
trackSessions: [ToBeChangedOrRemoved: <DOS_FILTER_TTACK_SESSIONS>]
remotePort: [ToBeChangedOrRemoved: <DOS_FILTER_REMOTE_PORT>]
ipWhitelist: [ToBeChangedOrRemoved: <DOS_FILTER_IP_WHITE_LIST>]

multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 67108864

# Media Life Cycle Management
backgroundScheduledMediaOperationsSettings:
  enableBackgroundScheduledMediaOperations: [ToBeChangedOrRemoved: "true"|"false"]
  schedulerThreads: 4
  schedulePollingInterval: 60
  speechMinerMaxConnection: 20
  speechMinerMaxTotalConnection: -1
  speechMinerSocketTimeout: 60000
  defaultBackupExportURI: [ToBeChangedOrRemoved: <DEFAULT_BACKUP_EXPORT_URI>]
  useFullPathInMediaFileBackup: false
  enableScanAndScroll: [ToBeChangedOrRemoved: "true"|"false"]
  scanIntervalsPerDay: [ToBeChangedOrRemoved: <SCHEDULE_MEDIA_OPERATION_SCAN_INTERVALS_PER_DAY>]

# CometD Settings
cometDSettings:
  cometdSessionExpirationTimeout: 60
  closeHttpSessionOnCometDExpiration: true
  maxSessionsPerBrowser: 1
  multiSessionInterval: 2000

# Log Header Settings
logHeaderSettings:
  enableLogHeader: [ToBeChangedOrRemoved: "true"|"false"]
  updateOnPremiseInfoInterval: 600

# Update on startup settings
updateOnStartup:
  opsCredentials: false
  features: false
```

Important

If you are using Elasticsearch 7.16.3 (or higher), refer to the below **elasticSearchSettings** section for setup.

```
# Elasticsearch
elasticSearchSettings:
  retriesOnConflict: 3
  useRestClient: true
  restClient:
    nodes: [ToBeChanged: <ELASTIC_SEARCH_REST_NODES>]
    authentication:
      type: [ToBeChangedOrRemoved: none|basic]
      userName: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_USER_NAME>]
      password: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_PASSWORD>]
    useSSL: [ToBeChangedOrRemoved: "true"|"false"]
    truststore: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PATH>]
    truststorePassword: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PASSWORD>]
  waitToIndexTimeout: 5000
  scanReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCAN_READ_TIMEOUT_SECONDS>]
  scrollTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCROLL_TIMEOUT_SECONDS>]
  countReadTimeoutSeconds: [ToBeChangedOrRemoved:
<ELASTIC_SEARCH_COUNT_READ_TIMEOUT_SECONDS>]
```

Make sure you update all settings marked as [ToBeChanged]. You should also be sure to do the following:

- Set the **applicationName** to the name of the application you created in [Creating the IRWS Node Application](#) — for example, IRWS_Node.
- In each Interaction Recording Web Services cluster or shared Interaction Recording Web Services and Web Services and Applications cluster, if both are deployed, one node in the cluster must be configured as the synchronization node: `syncNode: true`. All other nodes in the cluster must have `syncNode: false`.

Important

- To create the **ops** user and credentials in Cassandra and to enable the features in the **Interaction Recording Web Services** node, set the following parameters to true during the first Interaction Recording Web Services startup in the **application.yaml** file:

updateOnStartup

opsCredentials: true

features: true

After Interaction Recording Web Services is started, you must change both options to false for production:

updateOnStartup
opsCredentials: false
features: false

- A User object with user name set to default is a predefined object from Configuration Database and it is referred to as the Master Account. The Master Account is not alterable in any way, and you should not use it to perform regular contact center administrative tasks. For more information, see [Configuration Database](#).

The synchronization node is not responsible for importing the user name called default from Configuration Server into Cassandra, subscribing to change notifications with Configuration Server, or processing updates.

See [serverSettings](#) for details about all supported configuration settings for this section.

On Premise Settings

The settings in the **onPremiseSettings** section instruct Interaction Recording Web Services on how to communicate with the Configuration Server. The **application.yaml.sample** file includes the following default **onPremiseSettings** section:

```
# On Premise Settings (when syncNode is true)
onPremiseSettings:
  cmeHost: [ToBeChanged: <CONFIG_SERVER_HOST>]
  cmePort: [ToBeChanged: <CONFIG_SERVER_PORT>]
  backupCmeHost: [ToBeChanged: <BACKUP_CONFIG_SERVER_HOST>]
  backupCmePort: [ToBeChanged: <BACKUP_CONFIG_SERVER_PORT>]
  countryCode: [ToBeChanged: "US" | "CA" | etc]
  tlsEnabled: [ToBeChangedOrRemoved: "true"|"false"]
```

Make sure you update all settings marked as [ToBeChanged]. See [onPremiseSettings](#) for details about all supported configuration settings for this section.

Important

Note that settings under **onPremiseSettings** are used only once during the first initialization of RWS on the sync node. Further changes in the environment are retrieved from the Configuration Server directly. If a setting is configured incorrectly, please contact Genesys Customer Care for support.

Tuning the Interaction Recording Web Services Host Performance

Complete the following steps on each **Interaction Recording Web Services** node to tune the performance of the host environment.

Start

1. To optimize TCP/IP performance, add the following to the **/etc/sysctl.conf** file:

```
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 16384 16777216
net.core.somaxconn=4096
net.core.netdev_max_backlog=16384
net.ipv4.tcp_max_syn_backlog=8192
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_congestion_control=cubic
```

2. Increase the file descriptors by adding the following to the **/etc/security/limits.conf** file:

```
gir          hard nofile          100000
gir          soft nofile          100000
```

3. Run **sysctl -p** to reload the new values. These values will now always be loaded when rebooting.

End

Enabling features in the Feature Definitions file

The Feature Definitions file contains a list of features that are available for your contact center. The file is used to define features for the contact center by both Web Services (when installed) and Interaction Recording Web Services. For this reason, the procedure has a dependency on whether Web Services is being deployed along with Interaction Recording Web Services.

Perform the following operations on one of the **Interaction Recording Web Services** nodes.

Important

From Interaction Recording Web Services version 8.5.500.30 (or higher), Interaction Recording Web Services only Installations is supported .

For Web Services and Interaction Recording Web Services Installations

1. Locate the **gir-feature-definitions.json** file in the **installation_CD/config-templates** folder.

2. If you have already followed the [Enabling features in the Feature Definitions](#) file instructions from the *Web Services and Applications Deployment Guide (GWS)*, locate the **feature-definitions.json** file that was installed and edited into the **GWS_CONF** folder on the **Web Services** nodes.
If you did not already follow the **Enabling features in the Feature Definitions** file instructions, locate the **gws-feature-definitions.json** file in the **installation_CD/config-templates** folder.
3. Merge the contents of the two files together into a **feature-definitions.json** file in the **GWS_CONF** folder, as follows:
 - a. Ensure there is only one set of enclosing [...] (for example, first and last lines).
 - b. Ensure there is a comma after each { ... } excluding the last.
 - c. Ensure there are no duplicate items, for instance **api-provisioning-read** and **api-provisioning-write**.
4. Edit the file and for each feature that you want to enable for a new contact center, set the **autoAssignOnContactCenterCreate** flag to true. If you have already created your contact center or you are unsure of which Interaction Recording Web Services features to enable at this point, leave the **autoAssignOnContactCenterCreate** flags as they appear.

Important

The instructions that follow provide more detail about the Interaction Recording Web Services features and how to enable or disable them using **REST API** endpoints. For additional information, refer to [Configuring Features](#).

Merged Feature Definitions File - Example [+] [Show example.](#)

```
[
  {
    "id":"api-provisioning-read",
    "displayName":"API Provisioning Read",
    "description":"General provisioning read",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-provisioning-write",
    "displayName":"API Provisioning Write",
    "description":"General provisioning write",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-voice",
    "displayName":"Voice API",
    "description":"API for Voice",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-voice-predictive-calls",
    "displayName":"Voice API - Predictive calls",
    "description":"Enables predictive calls for a contact center",
    "autoAssignOnContactCenterCreate":true
  },
  {

```

```
    "id": "api-voice-outbound",
    "displayName": "Voice API Outbound",
    "description": "API for Outbound",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-supervisor-agent-control",
    "displayName": "API Supervisor Agent Control",
    "description": "API for Supervisors to Control Agent State",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-supervisor-monitoring",
    "displayName": "API Supervisor Monitoring",
    "description": "API for Supervisors to Monitor Agents",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-multimedia-chat",
    "displayName": "Multimedia Chat API",
    "description": "API for Multimedia Chat",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-email",
    "displayName": "Multimedia Email API",
    "description": "API for Multimedia Email",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-facebook",
    "displayName": "Multimedia Facebook API",
    "description": "API for Multimedia Facebook",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-twitter",
    "displayName": "Multimedia Twitter API",
    "description": "API for Multimedia Twitter",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-multimedia-workitem",
    "displayName": "Multimedia Workitem API",
    "description": "API for Multimedia Workitem",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-user-account-management-email",
    "displayName": "User Account Management via Email",
    "description": "API for account management via email",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-devices-webrtc",
    "displayName": "WebRTC Support",
    "description": "API for WebRTC provisioning",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-ucs-voice",
    "displayName": "Support UCS for voice",
    "description": "For support contact center in voice",
```

```
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-voice-instant-messaging",
    "displayName":"API Voice Instant Messaging",
    "description":"API for Internal Agent-to-Agent Chat",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-platform-configuration-read",
    "displayName":"Platform Configuration API - read",
    "description":"Low-level configuration API",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-platform-configuration-write",
    "displayName":"Platform Configuration API - write",
    "description":"Low-level configuration API",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-voice-recording",
    "displayName":"Voice API Recording",
    "description":"API for Voice Recording",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-voice-screenrecording",
    "displayName":"Screen Recording API (Voice)",
    "description":"API for Agent Voice Screen Recording",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-supervisor-recording",
    "displayName":"API Supervisor Recording",
    "description":"API for Call Recording Supervisor",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-multimedia-screenrecording",
    "displayName":"Screen Recording API (Multimedia)",
    "description":"API for Agent Multimedia Screen Recording",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"api-recordings-decryption-proxying",
    "displayName":"API Recordings Decryption Proxying",
    "description":"API For HTCC proxied interaction recording decryption",
    "autoAssignOnContactCenterCreate":true
  },
  {
    "id":"api-screenrecording-connection-reporting",
    "displayName":"API Screen Recording Connections Reporting",
    "description":"APIs for reporting on screen recording client connections",
    "autoAssignOnContactCenterCreate":false
  },
  {
    "id":"schema-elasticsearch-v2-call-recording",
    "displayName":"Schema Elasticsearch Call Recording Index V2",
    "description":"Elasticsearch call recording index schema v2",
    "autoAssignOnContactCenterCreate":true
  },
  {

```



```

    "id": "schema-elasticsearch-migration-to-v2-call-recording",
    "displayName": "Schema Elasticsearch Migration To Call Recording Index V2",
    "description": "Elasticsearch call recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "schema-elasticsearch-v2-screen-recording",
    "displayName": "Schema Elasticsearch Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "schema-elasticsearch-migration-to-v2-screen-recording",
    "displayName": "Schema Elasticsearch Migration To Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "data-skip-attach-screenrecording-data-to-callrecording",
    "displayName": "Skip Attaching Screen Recording Data To Call Recording",
    "description": "Whether or not to skip attaching screen recording data to call recording metadata",
    "autoAssignOnContactCenterCreate": false
  }
]

```

5. Follow the steps in the [Ensuring the Feature Definitions file is Read at Start-Up](#) section.

For Interaction Recording Web Services Only Installations

1. Locate the **gir-feature-definitions.json** file in the **installation_CD/config-templates** folder.
2. Copy the file to **feature-definitions.json** file in the **GWS_CONF** folder, and open the file.
3. For each feature that you want to enable for a new contact center, set the **autoAssignOnContactCenterCreate** flag to **true**. If you are unsure of which Interaction Recording Web Services features to enable, leave them as they appear.

Important

The instructions that follow provide more detail about the Interaction Recording Web Services features and how to enable or disable them using **REST API** endpoints. For additional information, refer to [Configuring Features](#).

Feature Definitions File - Example [\[+\] Show example.](#)

```

{
  "id": "api-provisioning-read",
  "displayName": "API Provisioning Read",
  "description": "General provisioning read",

```

```

    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-provisioning-write",
    "displayName": "API Provisioning Write",
    "description": "General provisioning write",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-voice-recording",
    "displayName": "Voice API Recording",
    "description": "API for Voice Recording",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-voice-screenrecording",
    "displayName": "Screen Recording API (Voice)",
    "description": "API for Agent Voice Screen Recording",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-supervisor-recording",
    "displayName": "API Supervisor Recording",
    "description": "API for Call Recording Supervisor",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-multimedia-screenrecording",
    "displayName": "Screen Recording API (Multimedia)",
    "description": "API for Agent Multimedia Screen Recording",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "api-recordings-decryption-proxying",
    "displayName": "API Recordings Decryption Proxying",
    "description": "API For HTCC proxied interaction recording decryption",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "api-screenrecording-connection-reporting",
    "displayName": "API Screen Recording Connections Reporting",
    "description": "APIs for reporting on screen recording client connections",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "schema-elasticsearch-v2-call-recording",
    "displayName": "Schema Elasticsearch Call Recording Index V2",
    "description": "Elasticsearch call recording index schema v2",
    "autoAssignOnContactCenterCreate": true
  },
  {
    "id": "schema-elasticsearch-migration-to-v2-call-recording",
    "displayName": "Schema Elasticsearch Migration To Call Recording Index V2",
    "description": "Elasticsearch call recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "schema-elasticsearch-v2-screen-recording",
    "displayName": "Schema Elasticsearch Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2",
    "autoAssignOnContactCenterCreate": true
  },
  {
  }
}

```

```
    "id": "schema-elasticsearch-migration-to-v2-screen-recording",
    "displayName": "Schema Elasticsearch Migration To Screen Recording Index V2",
    "description": "Elasticsearch screen recording index schema v2 migration support",
    "autoAssignOnContactCenterCreate": false
  },
  {
    "id": "data-skip-attach-screenrecording-data-to-callrecording",
    "displayName": "Skip Attaching Screen Recording Data To Call Recording",
    "description": "Whether or not to skip attaching screen recording data to call recording metadata",
    "autoAssignOnContactCenterCreate": false
  }
]
```

4. Follow the steps in the [Ensuring the Feature Definitions file is Read at Start-Up](#) section.

Ensuring the Feature Definitions file is Read at Start-Up

The Feature Definitions file is by default not read at start-up.

To ensure that it is read at start-up:

1. Add the following setting to **application.yaml** under the **serverSettings** section, on one of the **Interaction Recording Web Services** nodes:

```
updateOnStartup:
  features: true
```

2. Restart the **Interaction Recording Web Services** node.
3. Ensure you remove the setting after Interaction Recording Web Services has been started.

Important

Instructions about starting can be found in the [Starting and Testing](#) page.

Next Step

- [Configure additional security \(optional\)](#).

Configuring Security

Web Services adheres to the standards described in the Open Web Application Security Project (OWASP) Top 10—see the [OWASP website](#) for details—and has adopted several methods of ensuring security, for example:

- Errors are logged locally to prevent information leakage through API requests.
- User sessions have a timeout option.
- Cross Site Request Forgery Protection

Read on for details about the additional security configurations that Interaction Recording Web Services includes.

Transport Layer Security (TLS)

Complete the procedures below to configure TLS for connections received by Interaction Recording Web Services, and for connections from Interaction Recording Web Services to the following:

- Configuration Server
- SIP Server
- Interaction Server
- WebDAV
- Recording Crypto Server
- SpeechMiner Interaction Receiver
- Cassandra

Configuring TLS on the Server Side for Interaction Recording Web Services

1. Enable SSL on Jetty by configuring the **SSL** section of the **application.yaml** file using the following parameters:

```
enableSsl: true
ssl:
  port: 443
  keyStorePath: keystore
  keyStorePassword: storepwd
```

For more information on the parameters, see [ssl](#).

Important

On Unix-based systems, port 443 is protected; typically, only the superuser root can open it. For security reasons, it is not recommended to run the server as root. Therefore, Genesys recommends that you bind it to a non-protected port. Typically, any port above 1024 can be used (for example, you could set it to 9443). If you want to continue using port 443, see [Setting Port 80 Access for a Non-Root User](#) in the Jetty documentation.

2. Acquire the certificate and private keys.
3. To load a certificate and private keys (jetty.crt), navigate to the GWS_HOME/etc directory and run the following commands:

```
keytool -keystore keystore -import -alias jetty -file jetty.crt -trustcacerts
```
4. When prompted for the keystore password, enter the default: storepwd
5. Restart Interaction Recording Web Services (Web Services).

To create a self-signed certificate for non-production purposes:

1. Run the following in GWS_HOME/etc:

```
keytool -genkey -keyalg RSA -keystore keystore -alias jetty -ext SAN=dns:<server_dns_name>,ip:<server_ip_address>
```
2. When prompted for the keystore password, enter the default: storepwd
For more information about configuring SSL, see [Configuring SSL/TLS](#).

To change the certificate:

1. Remove the existing certificate using the following command:

```
keytool -keystore keystore -delete -alias jetty
```
2. Acquire the certificate and private key in a X509 PEM file (for example, jetty.crt).
3. Load the certificate using the following command:

```
keytool -keystore keystore -import -alias jetty -file jetty.crt -trustcacerts
```
4. Restart Interaction Recording Web Services (Web Services).

To change the keystore password:

1. Execute the following command:

```
keytool -keystore keystore -storepasswd
```
2. Encode the new password using the following command:

```
java -cp lib/jetty-http-xxx.jar:lib/jetty-util-xxx.jar org.mortbay.jetty.security.Password <your password here>
```

Configuring TLS connections to Configuration Server, SIP Server, and Interaction

Server

Interaction Recording Web Services can use a secured Transport Layer Security (TLS) connection mechanism to connect to Configuration Server, Interaction Server, and SIP Server. When configured, Interaction Recording Web Services connects to secure ports on Configuration Server, Interaction Server, and SIP Server; verifies the server's certificate; and encrypts/decrypts network traffic. You can configure secured connections to Configuration Server, Interaction Server, and SIP Server in the following ways:

- [Minimal configuration for Configuration Server, SIP Server, and Interaction Server](#)
- [Validate the Certificate Against the CA](#)

Note that each connection is configured independently, but a similar mechanism is used to configure each connection.

Prerequisites

Before configuring Interaction Recording Web Services, make sure the secure port on the server is configured as described in [Introduction to Genesys Transport Layer Security](#) in the [Genesys Security Deployment Guide](#) and that certificates for the server and the Certificate Authority are configured and available.

Minimal configuration for Configuration Server, SIP Server, and Interaction Server

In this configuration, Interaction Recording Web Services does not check the certificate against the Certificate Authority, but all traffic is encrypted. To configure Interaction Recording Web Services with minimal configuration, all you need to do is configure a connection to a secured port on Configuration Server, SIP Server, and Interaction Server. You can do this using *either* of the following methods:

- For the initial connection to Configuration Server, set the `tlsEnabled` option to `true` in the **onPremiseSettings** section of the RWS **application.yaml** file on the RWS node that is configured to be the sync node. This creates a secured connection to Configuration Server the first time Interaction Recording Web Services starts.
- For an environment that is already configured with Configuration Server synchronization enabled, you can make changes with Configuration Server as described in the [Genesys Security Deployment Guide](#). These changes are synchronized back to the Cassandra database from Configuration Server.

Important

Configuration Server supports the auto-upgrade port connection for secure communication from other GIR components; however, a secure port (listening mode of type secured) must be used for connectivity from RWS to the Configuration Server.

Ensure that connections from the Cluster Application being used by RWS (either `IRWS_Cluster` or `WS_Cluster`; see [Installing Interaction Recording Web Services](#) for more information) specify the appropriate secure port on each of the servers.

Validate the Certificate Against the CA

The procedure to validate the certificate against the CA is common to Configuration Server, SIP Server and Interaction Server.

Ensure you have completed the procedure described in the [Minimal configuration for Configuration Server, SIP Server, and Interaction Server](#) section.

To support the client-side certificate check, Interaction Recording Web Services needs the public key for the Certificate Authority (CA). Interaction Recording Web Services supports the PEM and JKS key storage formats, but recommends using JKS because it's compatible with both Cassandra and HTTPS.

To validate the certificate against the CA, specify the path to a file containing the trusted CA in the **caCertificate** parameter in the **application.yaml** file. By specifying this parameter, this CA will be checked against the server CA for validation.

Important

Only a single CA can be used to validate the certificate from Configuration Server, SIP Server, and Interaction Server.

If the configured server certificate matches the hostname of the server for any of the following fields, then Interaction Recording Web Services will validate the certificate.

- Issuer CN
- Subject CN
- Subject Alternative Name DNS

To validate the certificate against the CA, complete the following steps.

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. If you plan to use a JKS file, you can generate it from a PEM file by importing the PEM certificate, as shown here:

```
keytool -importcert -file ca_cert.pem -keystore ca_cert.jks
```

2. Once you have the **ca_cert.jks** file, place it in a location accessible from your Interaction Recording Web Services host, such as:
 - A local folder on the Interaction Recording Web Services host

- A shared folder

3. Configure the following options in the **serverSettings** section of the **application.yaml** file:

- For a PEM file, set **caCertificate** to the location of the file. For example:

```
caCertificate: /opt/ca_cert.pem
```

- For a JKS file, set **caCertificate** to the location of the file and set **jksPassword** to the password for the key storage. For example:

```
caCertificate: /opt/ca_cert.jks
jksPassword: pa$$word
```

End

Configuring TLS for Connections to WebDAV

By default, Interaction Recording Web Services checks the WebDAV server's certificate against a Certificate Authority using the Java default trustStore **caCerts**. To configure Interaction Recording Web Services with a customized trustStore configuration or to disable certificate validation, set the following options in the **application.yaml** configuration file.

Name	Parent	Value	Default
webDAVTrustedCA	serverSettings	<p>Configures TLS certificate validation when Interaction Recording Web Services connects to a WebDAV server. Valid values are <code>true</code>, <code>false</code>, or a path to a file containing one or more CA certificates.</p> <ul style="list-style-type: none">• If set to <code>true</code>, the certificate that WebDAV presents will be validated by caCerts in <code>\$JAVA_HOME/jre/lib/security</code>.• If set to <code>false</code>, the certificate that WebDAV presents will not be validated.• Any other value is considered as a path to a file containing a certificate for a Certificate Authority and RWS will use it to validate the WebDAV certificate. Both PEM and JKS	true

Name	Parent	Value	Default
		key storage formats are supported. If the specified file does not exist, Interaction Recording Web Services will exit during initialization.	
webDAVJksPassword	serverSettings	The password for the key storage for WebDAV if the specified CA file is in JKS format. You can specify an encrypted password. For more information on encrypting a password, see Password Encryption .	Empty

Configuring TLS for Connections to Recording Crypto Server

By default, Interaction Recording Web Services checks the Recording Crypto Server's certificate against a Certificate Authority using the Java default trustStore **caCerts**. To configure Interaction Recording Web Services with a customized trustStore configuration or to disable certificate validation, set the following options in the **application.yaml** configuration file.

Name	Parent	Value	Default
rcsTrustedCA	serverSettings	<p>Configures TLS certificate validation when Interaction Recording Web Services connects to the Recording Crypto Server. This property can be set to <code>true</code>, <code>false</code>, or a path to a file containing one or more CA certificates.</p> <ul style="list-style-type: none"> If set to <code>true</code>, the certificate that RCS presents will be validated by caCerts in <code>\$JAVA_HOME/jre/lib/security</code>. If set to <code>false</code>, the certificate that RCS presents will not be validated. 	true

Name	Parent	Value	Default
		<ul style="list-style-type: none"> Any other value is considered as a path to a file containing one or more CA certificates and RWS will use it to validate the RCS certificate. The key storage format can be either PEM or JKS. If the specified file does not exist, Interaction Recording Web Services will exit during initialization. 	
rcsJksPassword	serverSettings	The password for the key storage for RCS if the specified CA file is in JKS format. You can specify an encrypted password. For more information on encrypting a password, see Password Encryption .	Empty

Configuring TLS for Connections to SpeechMiner Interaction Receiver

By default, Interaction Recording Web Services checks the SpeechMiner Interaction Receiver's certificate against a Certificate Authority using the Java default trustStore **caCerts**. To configure Interaction Recording Web Services with a customized trustStore configuration or to disable certificate validation, set the following options in the **application.yaml** configuration file.

Name	Parent	Value	Default
speechMinerTrustedCA	serverSettings	<p>Configures TLS certificate validation when Interaction Recording Web Services connects to SpeechMiner Interaction Receiver. Valid values are true, false, or a path to a file containing one or more CA certificates.</p> <ul style="list-style-type: none"> If set to true, the certificate that the SpeechMiner Interaction Receiver 	true

Name	Parent	Value	Default
		<p>presents will be validated by caCerts in <code>\$JAVA_HOME/jre/lib/security</code>.</p> <ul style="list-style-type: none"> If set to false, the certificate that the SpeechMiner Interaction Receiver presents will not be validated. Any other value will be considered as a path to a file containing one or more CA certificates and RWS will use it to validate the SpeechMiner Interaction Receiver certificate. The key storage format can be either PEM or JKS. If the specified file does not exist, Interaction Recording Web Services will exit during initialization. 	
speechMinerJksPassword	serverSettings	<p>The password for the key storage for SpeechMiner Interaction Receiver if the specified CA file is in JKS format. You can specify an encrypted password. For more information on encrypting a password, see Password Encryption.</p>	Empty

Configuring TLS for Connections with Cassandra

Genesys supports Transport Layer Security (TLS) for connections from Interaction Recording Web Services to Cassandra and between Cassandra nodes. You can configure secured connections for the following scenarios:

- [Secure Connections from Interaction Recording Web Services to Cassandra](#)
- [Secure Connections from Interaction Recording Web Services to Cassandra 4](#)

- [Secure Connections between Cassandra Nodes](#)

Secure Connections from Interaction Recording Web Services to Cassandra

Prerequisites

- You have installed [Bash](#), [Java keytool](#), and [OpenSSL](#).

Complete the following steps to configure TLS for connections from Interaction Recording Web Services to Cassandra.

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. Create the server-side keystore with a self-signed certificate and the client-side truststore — which contains the public part of server certificate — with the following commands:

```
#!/bin/bash
#generate keypair
keytool -genkeypair -alias cassandra -keyalg RSA -keysize 1024 -dname
"CN=<Cassandra node hostname>, OU=Test, O=Test Ltd, C=US" -keystore
server.jks
-storepass password -keypass password
#export certificate
keytool -exportcert -alias cassandra -file client.pem -keystore
server.jks -storepass password -rfc
#create client truststore and import certificate
keytool -importcert -alias cassandra -file client.pem -keystore
client.jks -storepass password -noprompt
```

2. Create a self-signed root authority, use it to sign the server certificate, store it to **server.jks** and create the client-side truststore, which trusts all certificates signed with root authority. Run the following commands:

```
#!/bin/sh

#generate self-signed root certificate
keytool -genkeypair -alias root -keyalg RSA -keysize 1024 -validity 3650 -dname
"CN=TestRoot, OU=Dev, O=Company, C=US" -keystore root.jks
-storepass password -keypass password

#export root certificate
keytool -exportcert -alias root -file root.crt -keystore root.jks -storepass password

#generate server-side certificate
keytool -genkeypair -alias server -keyalg RSA -keysize 1024 -validity 3650 -dname
"CN=TestServer, OU=Dev, O=Company, C=US"
-keystore server.jks -storepass password -keypass password
```

```
#create the sign request for server certificate
keytool -certreq -alias server -keystore server.jks -file server.csr -storepass password
-keypass password

#export private key of root auth: need later for signing the server certificate
keytool -v -importkeystore -srckeystore root.jks -srcalias root -destkeystore root.p12
-deststoretype PKCS12 -noprompt
-destkeypass password -srckeypass password -destalias root -srcstorepass password
-deststorepass password

openssl pkcs12 -in root.p12 -out private.pem -password pass:password -passin
pass:password -passout pass:password
rm root.p12

#sign the certificate
openssl x509 -req -CA private.pem -in server.csr -out server.crt -days 3650
-CAcreateserial -passin pass:password
rm private.pem
rm private.srl
rm server.csr

#import root certificate to client side trust store
keytool -importcert -alias root -file root.crt -keystore client.jks -storepass password
-noprompt

#import root certificate to server side key store
keytool -importcert -alias root -file root.crt -keystore server.jks -storepass password
-noprompt
rm root.crt

#import certificate sign reply into server-side keystore
keytool -import -trustcacerts -alias server -file server.crt -keystore server.jks
-storepass password -keypass password
rm server.crt
```

3. Configure Cassandra to use your generated certificates for the client connection by setting the `client_encryption_options` in the **cassandra.yaml** file. For example:

```
client_encryption_options:
  enabled: true
  keystore: <absolute path to server.jks file>
  keystore_password: password
  #the password specified in while creating storage
  # For the purpose of the demo the default settings were used.
  # More advanced defaults below:
  #protocol: TLS
  #algorithm: SunX509
  #store_type: JKS
  #cipher_suites: [TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA]
```

Important

To enable support for encryption, you must have the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction installed.

4. Confirm the Cassandra nodes can start successfully:
 - a. Edit the **conf/log4j-server.properties** file and uncomment the following line:

```
log4j.logger.org.apache.cassandra=DEBUG
```

- b. Start Cassandra and check the logs. If the configuration was successful, you shouldn't see any errors.

- c. Edit the **conf/log4j-server.properties** file and comment the following line to disable the functionality:

```
log4j.logger.org.apache.cassandra=DEBUG
```

- d. Check that SSL-to-client is working successfully using cqlsh.

- Confirm that unsecured connections aren't possible by starting cqlsh locally—this forces it to connect to the Cassandra instance running on localhost. You should expect to see the exception in the cqlsh output.

```
cqlsh `hostname`
```

- Confirm that secured connections are possible by configuring cqlsh with SSL encryption. Create a PEM key which will be used in the **.cqlshrc** file:

```
// Create PEM for client
keytool -importkeystore -alias cassandra -srckeystore
server.jks -destkeystore server.p12 -deststoretype PKCS12
openssl pkcs12 -in server.p12 -out client.pem -nodes
```

- Create a **.cassandra/cqlshrc** file in your home or client program directory. The following settings must be added to the file as described below. When validate is enabled, the host in the certificate is compared to the host of the machine that it is connected to verify that the certificate is trusted.

```
[connection]
hostname = <hostname of Cassandra node>
port = 9042
factory = cqlshlib.ssl.ssl_transport_factory
```

```
[ssl]
certfile = /path/to/client.pem
# Optional, true by default
validate = true
```

- Verify that you can connect to Cassandra using cqlsh:

```
$ cqlsh --ssl
Connected to HTCC Cassandra Cluster at
ci-vm378.us.int.genesyslab.com:9042.
[cqlsh 5.0.1 | Cassandra 2.2.3 | CQL spec 3.3.1 | Native
protocol v4]
Use HELP for help.
```

- Configure Interaction Recording Web Services to use SSL with Cassandra. On each Interaction Recording Web Services node, edit the **application.yaml** file as follows:

```
cassandraCluster:
  useSSL: true
  trustStore: /path/to/client.jks
  truststorePassword: password
```

- Restart each Interaction Recording Web Services node:

```
systemctl restart gir
```

End

Secure Connections from Interaction Recording Web Services to Cassandra 4

Start

1. Create a self-signed root authority, use it to sign the server certificate, store it to server-side keystore and create the client-side truststore, which trusts all certificates signed with root authority. Run the following commands:

```
#!/bin/bash
#generate self-signed root certificate
keytool -genkeypair -alias root -keyalg RSA -keysize 2048 -validity 3650 -dname
"CN=TestRoot, OU=Dev, O=Company, C=US" -keystore root.jks -storepass password
-keypass password -ext KeyUsage=digitalSignature,keyCertSign -ext
BasicConstraints=ca:true

#export root certificate
keytool -exportcert -alias root -file root.crt -keystore root.jks -storepass password

#export private key of root auth: need later for signing the server certificate
keytool -v -importkeystore -srckeystore root.jks -srcalias root -destkeystore
root.p12 -deststoretype PKCS12 -noprompt -destkeypass password -srckeypass password
-destalias root -srcstorepass password -deststorepass password
openssl pkcs12 -in root.p12 -out private.pem -password pass:password -passin
pass:password -passout pass:password -legacy
rm root.p12

#import root certificate to client side trust store
keytool -importcert -alias root -file root.crt -keystore client.jks -storepass
password -noprompt

#import root certificate to server side key store
keytool -importcert -alias root -file root.crt -keystore server.jks -storepass
password -noprompt
rm root.crt
```

2. Create the server-side keystore with a self-signed certificate and the client-side truststore — which contains the public part of server certificate — with the following commands:

Important

In case of Multi Node Cassandra, the following steps should be executed for each Cassandra node.

```
#!/bin/bash
#generate server-side certificate
keytool -genkeypair -alias <node> -keyalg RSA -keysize 2048 -dname "CN=<Cassandra
node>, OU=Test, O=Test Ltd, C=US" -keystore server.jks -storepass password -keypass
password

#export certificate
keytool -exportcert -alias <node> -file <node>_client.pem -keystore server.jks
-storepass password -rfc

#create and import certificate to client truststore
keytool -importcert -alias <node> -file <node>_client.pem -keystore client.jks
-storepass password -noprompt

#create the sign request for server certificate
```

```
keytool -certreq -alias <node> -keystore server.jks -file <node>.csr -storepass
password -keypass password

#sign the certificate
openssl x509 -req -CA private.pem -in <node>.csr -out <node>.cert -days 3650
-CAcreateserial -passin pass:password

rm private.pem
rm private.srl
rm <node>.csr

#import certificate sign reply into server-side keystore
keytool -import -trustcacerts -alias <node_cert> -file <node>.cert -keystore server.jks
-storepass password -keypass password
rm <node>.cert
```

3. Continue from Step.3 in [Secure Connections from Interaction Recording Web Services to Cassandra](#).

End

Secure Connections between Cassandra Nodes

When you enable SSL for connections between Cassandra nodes, you ensure that communication between nodes in the Cassandra cluster is encrypted, and that only other authorized Cassandra nodes can join the cluster.

The steps below show you how to create a single certificate to be used by all Cassandra nodes in the cluster. This simplifies cluster management because you don't need to generate a new certificate each time you add a new node to the cluster, which means you don't need to restart all nodes to load the new certificate.

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

Start

1. Generate a keystore and truststore. See Step 1 of [Secure Connections from Interaction Recording Web Services to Cassandra](#) for details.
2. On each Cassandra node in the cluster, set **server_encryption_options** in the **cassandra.yaml** file. For example:

```
server_encryption_options:
  internode_encryption: all
  keystore: <absolute path to keystore >
  keystore_password: <keystore password - somePassword in our sample>
  truststore: <absolute path to truststore>
  truststore_password: <truststore password - somePassword in our sample>
```

3. Check the Cassandra logs. If the configuration was successful, you shouldn't see any errors.

End

Configuring TLS for Connections with Elasticsearch and Elasticsearch basic authentication

Genesys supports Transport Layer Security (TLS) and authentication for connections from Interaction Recording Web Services (RWS) to Elasticsearch 8 and also for communication between Elasticsearch nodes. You can configure these connections manually or use Elasticsearch's built-in autoconfiguration tools.

- [Secure Connections from Interaction Recording Web Services to Elasticsearch 8](#)
- [Secure Connections between Elasticsearch Nodes](#)

Secure Connections from Interaction Recording Web Services to Elasticsearch 8

Configuring RWS with Elasticsearch Default certificates:

Important

The steps described in this procedure are meant to be an example for developers and should not be used in production. For a production environment, you should follow your own company's security policies for creating and signing certificates.

1. Elasticsearch 8 comes with security features enabled by default, even without any additional configuration. On the first startup, it automatically:
 1. Enables basic authentication.
 2. Generates a random password for the built-in elastic superuser.
 3. Enables HTTPS on localhost with a self-signed certificate.
 4. Generates default certificates and keystores for HTTP and transport layers.

Once Elasticsearch 8 has successfully started, the following files are generated in the Elasticsearch configuration directory (/etc/elasticsearch or where the **elasticsearch.yml** resides).

File	Purpose
http_ca.crt	The Certificate Authority (CA) certificate that signed the HTTP layer certificate of this cluster. Clients connecting over HTTPS should trust this CA.
http.p12	A PKCS#12 keystore containing the key and certificate for the HTTP layer of this node.
transport.p12	A PKCS#12 keystore containing the key and certificate for the transport layer, used for secure communication between all nodes in the cluster.

2. RWS configuration sample
 1. We can use http_ca.crt to create a truststore for RWS connecting to Elasticsearch over HTTPS.
 2. We can find <elastic user password in Elasticsearch logs.

3. Configure Interaction Recording Web Services to use SSL with Elasticsearch. On each Interaction Recording Web Services node, edit the **application.yaml** file as follows:

```
elasticSearchSettings:
  ...
  restClient:
    nodes:
      - {host: <elastic-search-node>, port: 9200}
    authentication:
      type: basic
      userName: elastic
      password: <ELASTIC_USER_PASSWORD>
    useSSL: true
    truststore: rws-client.jks # contains http_ca.crt
    truststorePassword: password
```

3. Restart each Interaction Recording Web Services node:

```
systemctl restart gir
```

Configuring RWS with Elasticsearch Manually generated certificates:

1. Generate Elasticsearch Node Keystore & Certificate

```
keytool -genkeypair -alias <elasticsearch-node> -keyalg RSA -keysize 2048 -dname
"CN=<Elasticsearch node hostname>, OU=Test, O=Test Ltd, C=US" -keystore es-server.jks
-storepass password -keypass password
keytool -exportcert -alias <elasticsearch-node> -file es-node.pem -keystore es-
server.jks -storepass password -rfc
```

2. Create Client-side Truststore for RWS

```
keytool -importcert -alias elasticsearch-node -file es-node.pem -keystore rws-
client.jks -storepass password -noprompt
```

3. To Create Additional Users in elasticsearch

```
<Elasticsearch_installation_directory>/bin/elasticsearch-users useradd <rws-user> -p
<password> -r <superuser>
```

4. Configure SSL options for Elasticsearch by editing the `elasticsearch.yml` in all elasticsearch nodes. For example:

```
xpack.security.enabled: true
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.keystore.path: /path/to/es-server.jks
xpack.security.http.ssl.keystore.password: password
```

5. Restart Elasticsearch services and check the logs. If the configuration was successful, you shouldn't see any errors.

1. Verify cluster health:

```
curl -u <rws-user>:<password> --cacert /path/to/ca.crt
https://<host>:9200/_cluster/health
```

6. Configure Interaction Recording Web Services to use SSL with Elasticsearch. On each Interaction Recording Web Services node, edit the **application.yaml** file as follows:

```
elasticSearchSettings:
  ...
  restClient:
    nodes:
```

```
- {host: <elastic-search-node1>, port: 9200}
authentication:
  type: basic
  userName: <ELASTIC_SEARCH_USER_NAME>
  password: <ELASTIC_SEARCH_PASSWORD>
useSSL: true
truststore: rws-client.jks
truststorePassword: password
```

7. Restart each Interaction Recording Web Services node:

```
systemctl restart gir
```

Configuring RWS with Elasticsearch Existing certificates:

1. Make sure you have the Elasticsearch CA certificate file.
2. Create the RWS truststore in .p12 format using keytool:

```
keytool -importcert -trustcacerts -alias elastic-ca -file elastic-stack-ca.crt
-keystore rws-truststore.p12 -storetype PKCS12
```

Elasticsearch Utilities:

1. Below are some useful references from the official Elastic documentation and community resources:
 1. How to enable security, passwords, HTTPS, etc. → [Set up minimal security for Elasticsearch](#)
 2. How to enable HTTPS for the REST API. → [Configure TLS/SSL on HTTP layer](#)
 3. Details about the elasticsearch-certutil helper script. → [Elasticsearch certutil documentation](#)
 4. Which users and roles are already available. → [Built-in users and roles](#)

Secure Connections between Elasticsearch Nodes

1. How to secure inter-node communication. → [Configure TLS/SSL on Transport layer](#)
2. Edit the elasticsearch.yml according to the configuration. For manual configuration example:

```
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.keystore.path: /path/to/es-server.jks
xpack.security.transport.ssl.keystore.password: password
xpack.security.transport.ssl.truststore.path: /path/to/es-server.jks
xpack.security.transport.ssl.truststore.password: password
xpack.security.transport.ssl.verification_mode: certificate
```

Cassandra Authentication

Interaction Recording Web Services supports Cassandra authentication, which validates incoming user connections to the Cassandra database. Implementing Cassandra authentication requires you to perform configuration in both Cassandra and Interaction Recording Web Services.

Configure Cassandra Authentication

The user account and password required for authentication are managed inside the **cassandra.yaml** file. Configure Cassandra authentication according to the [Cassandra 2.2 documentation](#).

Interaction Recording Web Services Configuration

To support Cassandra authentication, configure the appropriate credentials in the **cassandraCluster** section of the **application.yaml** file:

```
cassandraCluster:
  thrift_port: 9160
  jmx_port: 7199
  ...
  userName: <superuser name>
  password: <superuser password>
  ...
```

Important

- If the **userName** and **password** are not configured, RWS will connect to Cassandra anonymously.
- To encrypt the password for added security, see [Password Encryption](#).

Password Encryption

For added security, consider encrypting your passwords in the **application.yaml** file by using the following procedure:

1. Run the RWS application with the **--encrypt** parameter followed by the password you need to encrypt. For example, if the password is "ops":

```
$ java -jar gir.jar --encrypt ops
CRYPT:an03xPrxLAu9p==
```

RWS will encrypt and print the password. The server will not actually start.

2. Copy the printed encrypted password and paste into the **application.yaml** file. For example:

```
webDAVJksPassword: CRYPT:an03xPrxLAu9p==
```

The server only decrypts passwords that start with the **CRYPT:** prefix. Passwords without the **CRYPT:** prefix are considered plain text and remain unmodified.

CSRF Protection

Interaction Recording Web Services provides protection against Cross Site Request Forgery (CSRF) attacks. For general information and background on CSRF, see the [OWASP CSRF Prevention Cheat Sheet](#).

Important

If CSRF protection is enabled, then the label/tagging and deletion prevention functionality cannot be used in SpeechMiner, as SpeechMiner does not support CSRF.

To set up Cross Site Request Forgery protection, set the following options in the **serverSettings** section of the **application.yaml** file on each of your Interaction Recording Web Services nodes:

- **enableCsrfProtection**—determines whether CSRF protection is enabled on the Web Services node.
- **crossOriginSettings**—specifies the configuration for cross-origin resource sharing in Interaction Recording Web Services. Make sure this option has the **exposedHeaders** setting with a value that includes X-CSRF-HEADER,X-CSRF-TOKEN.

For example, your configuration might look like this:

```
enableCsrfProtection: true
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-Type,Accept,
Origin,Cookie,authorization,ssid,surl,ContactCenterId"
  allowCredentials: true
  exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

For more information about CSRF protection in the Interaction Recording Web Services API, see [Cross Site Request Forgery Protection](#).

CORS Filter

Interaction Recording Web Services supports Cross-Origin Resource Sharing (CORS) filter, which allows applications to request resources from another domain. For general information and background on CORS, see [Cross-Origin Resource Sharing](#).

To set up Cross-Origin Resource Sharing, make sure you set the **crossOriginSettings** option in the **serverSettings** section of the **application.yaml** file on each of your Interaction Recording Web Services nodes . It specifies the configuration for cross-origin resource sharing in Interaction Recording Web Services. Make sure this option has the **exposedHeaders** setting with a value that includes X-CSRF-HEADER,X-CSRF-TOKEN.

For example, your configuration might look like this:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,X-CSRF-TOKEN"
  allowCredentials: true
  exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

For more information about CORS in the Interaction Recording Web Services API, see [Cross-Origin Resource Sharing](#).

Interaction Recording Web Services Authentication Flow

Interaction Recording Web Services provides authentication in the following sequence:

1. Configuration Server Authentication

- If a request contains a basic authentication header and Configuration Server authentication is enabled for this contact center, Configuration Server authentication is applied.
 - If successful, user is authenticated and execution flow proceeds to the authorization stage.
 - If authentication headers are not present, Configuration Server authentication is disabled, or authentication fails, execution flow proceeds to the next step.

2. Interaction Recording Web Services Authentication

- If a request contains a basic authentication header and Configuration Server authentication is not enabled for this contact center, Interaction Recording Web Services authentication is applied.
 - If successful, user is authenticated and execution flow proceeds to the authorization stage.
 - If authentication headers are not present or authentication fails, execution flow proceeds to the next step.

Next Step

- [Starting and Testing](#)

Starting and Testing

Once you've installed and configured Interaction Recording Web Services, you're ready to start the individual nodes and confirm the service is working.

Starting the Interaction Recording Web Services Nodes

Complete the following steps for each Interaction Recording Web Services node, starting with the **syncNode**.

Important

Verify that the SR Service is active before an agent attempts to log into Workspace Web Edition (WWE).

To create the **ops** user and credentials in Cassandra and to enable the features in the Interaction Recording Web Services node, set the following parameters to true during the first Interaction Recording Web Services startup in the **application.yaml** file:

```
updateOnStartup  
opsCredentials: true  
features: true
```

RHEL 8/9

Start the RWS Service by entering the following command:
`sudo systemctl start gir`

Important

After Interaction Recording Web Services is started, you must change both options to false for production:

```
updateOnStartup  
opsCredentials: false  
features: false
```

Testing Interaction Recording Web Services

Complete the steps below to verify each Interaction Recording Web Services node is up and running.

Start

1. Type the following URL into a web browser:

`http://ws_host:ws_port/api/v2/diagnostics/version`

- *ws_host*—The host name or IP address for the Interaction Recording Web Services node.
- *ws_port*—The port for the Interaction Recording Web Services node.

For example, the URL might be `http://192.0.2.20:8080/api/v2/diagnostics/version`

If the request is successful, the version is printed in the browser:

```
{"statusCode":0,"version":"8.5.200.96"}
```

End

Next Step

- [Configure your required features](#)

Configuring Features

Review the sections below for more information about how to configure Interaction Recording Web Services to use the specified features.

Configuration for Voice Recordings

Interaction Recording Web Services requires a specific configuration for GIR **call** recordings to work correctly. The following sections describe how to configure Interaction Recording Web Services for call recordings.

Configuring the Interaction Recording Web Services Parameters

1. To support call recordings, it's important that you update the following settings in the `serverSettings` section of the **application.yaml** file:
 - `undocumentedExternalApiUrl`
 - `createCallRecordingCF`
 - `crClusterName`
 - `crRegion`
 - `cryptoSecurityKey`
 - `webDAVMaxConnection`
 - `webDAVMaxTotalConnection`
 - `nodePath`
 - `recordingSettings`, in particular **`recordCryptoServerDecryptMaxConnection`**, **`recordCryptoServerDecryptMaxTotalConnection`** and **`recordCryptoServerDecryptSocketTimeout`**
 - `multiPartResolverMaxUploadSize`
 - `multiPartResolverMaxInMemorySize`
 - `backgroundScheduledMediaOperationsSettings`, in particular **`enableBackgroundScheduledMediaOperations`** and **`defaultBackupExportURI`**
2. Determine the contact center ID for Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
  curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
  Recording Web Services port>/api/v2/ops/contact-centers; echo
}
```

Interaction Recording Web Services returns the following output:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
```

```
Recording Web Services port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>"}}}
```

- Using a text editor, create a new file called `add_voice_features` with the following content:

```
{
  "uris":[
    "/api/api-voice-recording",
    "/api/api-supervisor-recording",
    "schema-elasticsearch-v2-call-recording"
  ]
}
```

- Execute the following command:

```
{
curl -u ops:ops -X POST -d @add_voice_features
http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/features
--header "Content-Type: application/json"; echo
}
```

Configuring the Storage Credentials for Interaction Recording Web Services

Enable Voice Recording

Start

- Determine the contact center ID on Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services Port>/api/v2/ops/contact-centers; echo
}
```

The following output is returned:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services Port>/api/v2/ops/contact-centers/<contact center ID (in hex
format)>"}}
```

Important

Use the <contact center ID (in hex format)> in all subsequent commands.

- Using a text editor, create a new file called `create_table` with the following content:

```
{
  "operationName":"createCRCF"
}
```

Important

You do not need to create the table manually when the **createCallRecording** option is set to **true** in the **application.yaml** file. The table will be automatically created by Interaction Recording Web Services (RWS).

3. Execute the following command:

```
{
  curl -u ops:ops -X POST -d @create_table http://<Interaction Recording Web Services
  Server>:<Interaction Recording Web Services Port>/api/v2/ops/
  contact-centers/<contact center ID (in hex format)>/recordings
  --header "Content-Type: application/json"; echo
}
```

End

Enable Storage

Start

1. Using a text editor, create a new file called `recording_settings` with the following content:

```
{
  "store": [
    {
      "webDAV": {
        "userName": "user1",
        "password": "password1",
        "uri": "http://apache1/recordings"
      }
    },
    {
      "webDAV": {
        "userName": "user2",
        "password": "password2",
        "uri": "http://apache2/recordings"
      }
    }
  ]
}
```

Important

The URI in `recording_settings` is case sensitive and must match the URI in the IVR Profile. For example:

```
"uri": "http://GENESYSREC1/recordings"
is not the same as
"uri": "http://genesysrec1/recordings"
```

2. Execute the following command:

```
{
```

```
curl -u ops:ops -X POST -d @recording_settings
  http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/settings/call-
recordings
--header "Content-Type: application/json"; echo
}
```

End

Configuring the Call Recording Audit Log

Interaction Recording Web Services provides an audit log for the following recording operations:

- Playback of the recording media file
- Deletion of the recording file

Complete the steps below to configure the audit log:

Start

1. Stop Interaction Recording Web Services using the following command:

```
sudo service gir stop
```
2. Edit the **GWS_HOME/etc/logback.xml** file and update the configuration to include INFO level messaging. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Example LOGBACK Configuration File
  http://logback.qos.ch/manual/configuration.html
-->
<configuration scan="true">
  <appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <filter class="ch.qos.logback.classic.filter.LevelFilter">
      <level>INFO</level>
      <onMatch>ACCEPT</onMatch>
      <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY for
printing only INFO-->
    </filter>
    <file>${jetty.logs}/recording.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <maxHistory>720</maxHistory><!-- 1 Month -->
    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}] [%X{req.userAgent}]
[%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
    </encoder>
  </appender>
  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${jetty.logs}/cloud.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <!-- hourly rollover -->
      <fileNamePattern>${jetty.logs}/cloud-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <!-- keep 5 days' worth of history -->
      <maxHistory>120</maxHistory>
    </rollingPolicy>
  </appender>
</configuration>
```

```

    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} %-5level [%X{principal.name}]
[%X{session}] [%X{contactCenter}] [%thread] %X{req.requestURI} %X{req.queryString}
%logger{36} %msg%n</pattern>
    </encoder>
  </appender>
  <logger name="com.<domain>.cloud.v2.api.controllers.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>.cloud.v2.api.tasks.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>" level="WARN" />
  <logger name="com.<domain>.cloud" level="DEBUG" />
  <logger name="com.<domain>.cloud.rtreporting" level="WARN" />
  <logger name="com.<domain>.salesforce.security" level="INFO" />

  <root level="WARN">
    <appender-ref ref="FILE" />
  </root>
</configuration>

```

3. For MLM, create a **RECORDING** appender if it does not exist. For example:

```

<appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <filter class="ch.qos.logback.classic.filter.LevelFilter">
    <level>INFO</level>
    <onMatch>ACCEPT</onMatch>
    <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY for
printing only INFO-->
  </filter>
  <file>${jetty.logs}/recording.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd}.gz</fileNamePattern>
    <maxHistory>720</maxHistory><!-- 1 Month -->
  </rollingPolicy>
  <encoder>
    <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}] [%X{req.userAgent}]
[%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
  </encoder>
</appender>

```

4. Add the following loggers for the **RECORDING** appender:

```

<logger name="com.genesyslab.cloud.v2.api.controllers.callrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.controllers.screenrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.callrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.interactionrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.screenrecording">
  <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.settings">

```

```

    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.genesyslab.cloud.v2.media.scheduler">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.genesyslab.cloud.v2.media.task">
    <appender-ref ref="RECORDING" />
  </logger>

```

For more information about Logback, see [Logback configuration](#).

5. Start GIR using the following command:
`sudo service gir start`
6. Review the audit log. Open the <LOG_PATH>/recording.log file, where <LOG_PATH> is the path parameter for the logging section in your application.yaml. By default, this is /var/log/jetty9. The following example shows that two recordings are requested for playback and deletion:

```

10/28/2013 15:46:03.203 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/
537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

```

```

10/28/2013 15:46:03.341 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] failed

```

```

10/28/2013 15:46:10.946 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

```

```

10/28/2013 15:46:11.033 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] succeed

```

```

10/28/2013 15:46:52.179 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid0 Delete metadata and media-files of call-recording is requested. contact-center
[46284f2f-d615-4329-957a-f5341edfd5d7], call-recording [recid0]

```

```

10/28/2013 15:46:52.216 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/

```

```

recid0 Delete metadata and media-files of call-recording failed. contact-center
[46284f2f-d615-4329-957a-f5341edfd5d7], call-recording [recid0]

10/28/2013 15:46:56.253 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete metadata of call-recording is requested. contact-center [46284f2f-
d615-4329-957a-f5341edfd5d7], call-recording [recid1]

10/28/2013 15:46:56.420 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete metadata of call-recording succeeded. contact-center [46284f2f-
d615-4329-957a-f5341edfd5d7], call-recording [recid1]

```

End

Configuring the API Thread Pool

Interaction Recording Web Services provides properties for the Call Recording API thread pool by configuring the **hystrix.properties** file.

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name]. execution.isolation.thread. timeoutInMilliseconds	N/A	The hystrix timeout. The default value is set to 6000.
hystrix.threadpool.[API Pool Name] .coreSize	N/A	The thread pool size. The default value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateCallRecordingApiTaskV2	ApiCreatePool	Create call recording.
DeleteCallRecordingApiTaskV2	ApiDeletePool	Delete call recording.
GetCallRecordingApiTaskV2	ApiGetPool	Get call recording metadata.
GetCallRecordingCFInfoApiTaskV2	ApiGetPool	Get call recording CF Information.
GetCallRecordingMediaApiTaskV2	ApiGetPool	Streaming call recording media.
QueryCallRecordingApiTaskV2	ApiQueryPool	Query call recording metadata.

For more information about the Call Recording API, see the [Genesys Interaction Recording API Reference](#).

Configuration for Screen Recordings

As with call recordings, Interaction Recording Web Services requires a specific configuration for GIR **screen** recordings to work correctly. The following sections describe how to configure Interaction Recording Web Services for screen recordings.

Configuring the Interaction Recording Web Services Parameters

Complete the steps below to support screen recordings:

Start

1. For **Cassandra 2.2**, Update the following settings in the serverSettings section of the **application.yaml** file. Your configuration should look something like this:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: <Interaction Recording Web Services Servers>,<SpeechMiner Web
Servers>
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,Range"
  allowCredentials: true
screenRecordingSettings:
  screenRecordingEServicesEnabled: true
  screenRecordingVoiceEnabled: true
screenRecordingConnectionReportingSettings:
  reportingEnabled: true
  createReportingCF: true
multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 67108864
```

For **Cassandra 4.1**, Update the following settings in the serverSettings section of the **application.yaml** file. Your configuration should look something like this:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: <Interaction Recording Web Services Servers>,<SpeechMiner Web
Servers>
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-
Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,Range"
  allowCredentials: true
screenRecordingSettings:
  screenRecordingEServicesEnabled: true
  screenRecordingVoiceEnabled: true
screenRecordingConnectionReportingSettings:
  reportingEnabled: false
  createReportingCF: false
multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 67108864
```

Make the following changes to the example above:

- Change <Interaction Recording Web Services Servers> and <SpeechMiner Web Servers> to the HTTP/HTTPS addresses of the Interaction Recording Web Services instances and SpeechMiner Web Servers.

- **multipartResolverMaxUploadSize** controls the maximum allowed size (in bytes) for a screen recording video file that can be uploaded to Interaction Recording Web Services. This parameter should be aligned with **maxDurationMinutes**, so if you change its value, ensure that you also consider the **maxDurationMinutes** value specified within the *Advanced Configuration for the Screen Recording Service* section in the [Deploying the Screen Recording Service - Advanced Configuration](#) page. The maximum size of a file that can be uploaded by the Screen Recording Service must be less than or equal to the **multipartResolverMaxUploadSize**.

2. Determine the contact center ID on Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
  curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
  Recording Web Services port>/api/v2/ops/contact-centers; echo
}
```

Interaction Recording Web Services returns the following output:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>"]}
```

3. For **Cassandra 2.2**, Using a text editor, create a new file called `add_screen_features` with the following content:

```
{
  "uris":[
    "/api/api-voice-screenrecording",
    "/api/api-multimedia-screenrecording",
    "/api/api-screenrecording-connection-reporting",
    "schema-elasticsearch-v2-screen-recording"
  ]
}
```

For **Cassandra 4.1**, Using a text editor, create a new file called `add_screen_features` with the following content:

```
{
  "uris":[
    "/api/api-voice-screenrecording",
    "/api/api-multimedia-screenrecording",
    "schema-elasticsearch-v2-screen-recording"
  ]
}
```

4. Execute the following command:

```
{
  curl -u ops:ops -X POST -d @add_screen_features
  http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
  Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/features
  --header "Content-Type: application/json"; echo
}
```

- Use the **api-voice-screenrecording** parameter for voice interactions, and use the **api-multimedia-screenrecording** parameter for non-voice interactions.
- Use the **api-screenrecording-connection-reporting** parameter to enable the collection of information about Screen Recording Services client connections for the contact center. **This feature is not applicable for Cassandra 4.1.**
- If you wish to direct the SpeechMiner UI to Interaction Recording Web Services instead of Recording Crypto Server for decryption of screen recordings, add the **api-recordings-decryption-proxying** parameter to the list of features enabled for the contact center above. Note that this requires

additional configuration.

- Using a text editor, create a new file called `create_stats_table`, with the following content: **This step is not applicable for Cassandra 4.1.**

```
{
  "operationName": "CreateReportingCFs"
}
```

- Execute the following command: **This step is not applicable for Cassandra 4.1.**

```
{
  curl -u ops:ops -X POST -d @create_stats_table http://<Interaction Recording Web
  Services Server>:<Interaction Recording Web Services Port>/api/v2/ops/contact-
  centers/<contact center ID (in hex format)>/screen-recording-connections --header
  "Content-Type: application/json"; echo
}
```

End

Configuring the Storage Credentials for Interaction Recording Web Services

Complete the steps below to configure storage credentials for Interaction Recording Web Services.

Start

- Determine the contact center ID on Interaction Recording Web Services using the following command with the ops username and password (ops:ops):

```
{
  curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction
  Recording Web Services port>/api/v2/ops/contact-centers; echo
}
```

Interaction Recording Web Services returns the following output:

```
{"statusCode":0,"uris":["http://<Interaction Recording Web Services Server>:<Interaction
Recording Web Services port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>"]}
```

Important

Use the `<contact center ID (in hex format)>` construction in all subsequent commands.

- Using a text editor, create a new file called `create_table`, with the following content:

```
{
  "operationName": "createCRCF"
}
```

- Execute the following command:

```
{
```

```
curl -u ops:ops -X POST -d @create_table http:// <Interaction Recording Web Services
Server>:<Interaction Recording Web Services Port>/api/v2/ops/
contact-centers/<contact center ID (in hex format)>/screen-recordings
--header "Content-Type: application/json"; echo
}
```

4. Enable storage for a single or multiple locations:

Important

Within the storage settings, the same location can be specified multiple times if you have inactive ("active": false) settings specified as well as "active": true. However, you must ensure that for a specific location, only one value has "active": true set. For additional information about storage settings, refer to [Interaction Recording Web Services \(Web Services\) Group Settings](#). See the **Property Descriptions** section for details about the supported property values.

- For a **single** location:

a. Using a text editor, create the create_single_location file:

```
{
  "name": "storage",
  "location": "/",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "<webdav user>",
        "password": "<webdav password>",
        "storagePath": "<webdav uri>"
      }
    }
  ]
}
```

Important

Replace <webdav user>, <webdav password>, <webdav uri> with the appropriate values.

b. Execute the following command:

```
{
curl -u ops:ops -X POST -d @create_single_location http:// <Interaction Recording Web
Services Server>:<Interaction Recording Web Services Port>/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
}
```

- For **multiple** locations:

a. Using a text editor, create the create_first_location file:

```
{
  "name": "storage",
```

```

    "location": "<node_location>",
    "value": [
      {
        "storageType": "webDAV",
        "active": true,
        "credential": {
          "userName": "<webdav user>",
          "password": "<webdav password>",
          "storagePath": "<webdav uri>"
        }
      }
    ]
  }
}

```

- b. Execute the following command:

```

{
curl -u ops:ops -X POST -d @create_first_location http://<Interaction Recording
Web Services Server>:<Interaction Recording Web Services Port>/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
}

```

Important

Replace <node_location>, <webdav user>, <webdav password>, <webdav uri> with the appropriate values. The values for the <node_location> are similar to the **nodePath** settings in the **application.yaml** file, but allow a hierarchical representation. For example, an Interaction Recording Web Services node uses a storage setting with a location of "/US" in the nodePath set to "/US/AK" or "/US/HI".

For more information on hierarchical location setting, go [\[\[1\]\]](#)

- c. Repeat steps a and b for each location required.

End

For more information on the properties of this settings group, see [Interaction Recording Web Services Settings Groups](#).

Configuring the API Thread Pool

Interaction Recording Web Services provides properties for the Call Recording API thread pool by configuring the **hystrix.properties** file.

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name]. execution.isolation.thread. timeoutInMilliseconds	N/A	The hystrix timeout. The default value is set to 6000.
hystrix.threadpool.[API Pool]	N/A	The thread pool size. The default

Property/API Name	Thread Pool Name	Description
Name] .coreSize		value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateScreenRecordingApiTaskV2	ApiUploadPool	Create screen recording
DeleteScreenRecordingMediaApiTaskV2	ApiDeletePool	Delete screen recording
GetScreenRecordingApiTaskV2	ApiGetPool	Get screen recording metadata
GetScreenRecordingMediaApiTaskV2	ApiStreamPool	Stream screen recording media
QueryScreenRecordingApiTaskV2	ApiQueryPool	Query screen recording metadata

For more information about the Call Recording API, see the [Genesys Interaction Recording API Reference](#).

Configuration Options

You can set the configuration options below in the corresponding sections of the **application.yaml** file on your Interaction Recording Web Services nodes. For details, see [Configuring Interaction Recording Web Services](#).

Important

When editing the **application.yaml** file, the values for the configuration options that are strings must be enclosed in double quotation marks in certain cases. Specifically:

- For string options only, the values YES, NO, ON, OFF, TRUE, FALSE (in upper or lower case) must be quoted.
- If the option is a boolean (true/false) option, then any of the values in the previous bullet can be used without quotes.
- Values that look like numbers but are treated as strings (for example; PINs, phone numbers, encryption keys), that begin with leading zeroes must be quoted.
- Avoid placing leading zeroes on numeric options; doing so will cause your option to be interpreted as an octal value.

For example, specifying `crRegion: NO` (indicating Norway) will be interpreted as `crRegion: FALSE`. Instead, this must be specified using double quotation marks `crRegion: "NO"`.

logging

Settings in this section are listed under **logging**.

config

Default Value: `logback.xml`

Valid Values: A valid path

Mandatory: No

Specifies the path to the **logback.xml** file. You created this file (or Interaction Recording Web Services created it for you) as part of [Deploying the Web Application](#).

file

Default Value: `cloud.log`

Valid Values: A valid file name

Mandatory: No

Specifies the name of the log file. This value is stored in `${LOG_FILE}` which may be used in **logback.xml**.

path

Default Value: /var/log/jetty9

Valid Values: A valid path

Mandatory: No

Specifies the path to the log file. This value is stored in `${LOG_PATH}` which may be used in **logback.xml**.

jetty

Settings in this section are listed under **jetty**.

host

Default Value: 0.0.0.0

Valid Values: A host name or IP address

Mandatory: No

Specifies the host name or IP address of the Jetty host. This value should be the same as GWS_HOST you defined as part of [Deploying the Web Application](#).

port

Default Value: 8080

Valid Values: A valid port

Mandatory: No

Specifies the port of the Jetty host. This value should be the same as GWS_PORT you defined as part of [Deploying the Web Application](#).

idleTimeout

Default Value: 30000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum idle time, in milliseconds, for a connection.

soLingerTime

Default Value: -1

Valid Values: An integer greater than 0, or -1 to disable

Mandatory: No

Specifies the socket linger time.

sessionMaxInactiveInterval

Default Value: 1800

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the period, in seconds, after which a session is deemed idle and saved to session memory.

enableWorkerName

Default Value: true

Valid Values: true, false

Mandatory: No

Specifies whether to add the WorkerName parameter into the sessionCookieName cookie.

enableRequestLog

Default Value: true

Valid Values: true, false

Mandatory: No

Enables request logging. If you set the value to true, you must also set values for the [requestLog](#) option.

requestLog

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
filename	No	yyyy_mm_dd.cloud-request.log	Specifies the log file name format.
filenameDateFormat	No	yyyy_MM_dd	Specifies the log file name date format.
logTimeZone	No	GMT	Specifies the timestamp time zone used in the log.
retainDays	No	90	Specifies the time interval, in days, for which Jetty should retain logs.
append	No	true	Specifies whether Jetty appends to the request log file or starts a new file.
extended	No	true	Specifies whether Jetty logs extended data.
logCookies	No	true	Specifies whether Jetty logs request cookies.
logLatency	No	true	Specifies whether Jetty logs the request latency.
preferProxiedForAddress	No	true	Specifies whether Jetty logs IP address or the IP address from the X-Forwarded-For request header.

Mandatory: No

Specifies how Jetty should handle request logging. For example:

```
enableRequestLog: true
requestLog:
  filename: yyyy_mm_dd.cloud-request.log
  filenameDateFormat: yyyy_MM_dd
  logTimeZone: GMT
  retainDays: 90
  append: true
  extended: true
  logCookies: false
  logLatency: true
  preferProxiedForAddress: true
```

These options only take effect if **enableRequestLog** is set to true.

enableSsl

Default Value: false

Valid Values: true, false

Mandatory: No

Enables Secure Sockets Layer support. If you set the value to true, you must also set values for the **ssl** option.

ssl

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
port	No	443	The SSL port. This option is the equivalent of the Jetty "https.port" variable.
securePort	No	8443	The port to which integral or confidential security constraints are redirected. This option is the equivalent of the Jetty "jetty.secure.port" variable.
idleTimeout	No	30000	The maximum idle time, in milliseconds, for a connection.
soLingerTime	No	-1	The socket linger time. A value of -1 disables this option.
keyStorePath	No	None	The keystore path.
keyStorePassword	No	None	The keystore password.
keyManagerPassword	No	None	The key manager password.

Name	Mandatory	Default Value	Description
keyStoreProvider	No	None	The keystore provider.
keyStoreType	No	JKS	The keystore type.
trustStorePath	No	None	The truststore path.
trustStorePassword	No	None	The truststore password.
trustStoreProvider	No	None	The truststore provider.
trustStoreType	No	JKS	The truststore type.
needClientAuth	No	None	Set this option to true if SSL needs client authentication.
wantClientAuth	No	None	Set this option to true if SSL wants client authentication.
certAlias	No	None	The alias of the SSL certificate for the connector.
validateCerts	No	None	Set this option to true if the SSL certificate has to be validated.
validatePeerCerts	No	None	Set this option to true if SSL certificates of the peer have to be validated.
trustAll	No	None	Set this option to true if all certificates should be trusted if there is no keystore or truststore.
renegotiationAllowed	No	None	Set this option to true if TLS renegotiation is allowed.
excludeCipherSuites	No	None	Specifies the array of cipher suite names to exclude from enabled cipher suites.
includeCipherSuites	No	None	Specifies the array of cipher suite names to include in enabled cipher suites.
endpointIdentificationAlgorithm	No	None	Specifies the endpoint identification algorithm. Set this option to "HTTPS" to enable hostname verification.
includeProtocols	No	None	The array of protocol names (protocol versions) to include for use on this engine.

Name	Mandatory	Default Value	Description
excludeProtocols	No	None	The array of protocol names (protocol versions) to exclude from use on this engine.

Mandatory: No

Specifies how Jetty should handle support for Secure Sockets Layer. For example:

```
enableSsl: true
ssl:
  port: 443
  securePort: 8443
  idleTimeout: 30000
  soLingerTime: -1
```

These options only take effect if **enableSsl** is set to true.

httpOnly

Default Value: true

Valid Values: true, false

Mandatory: No

If true, it sets an HTTP-only flag for session cookies.

secure

Default Value: false

Valid Values: true, false

Mandatory: No

If true, it sets a secure cookie flag for session cookies.

sessionCookieName

Default Value: GIRJSESSIONID

Valid Values: Any string which can be used as a cookie name as per [RFC 6265](#)

Mandatory: No

Defines the name of the session cookie used by Interaction Recording Web Services.

sessionCookieName can only contain the following characters:

- Letters: a-z or A-Z
- Digits: 0-9
- Hyphen (-)
- Underscore (_)

cassandraCluster

Settings in this section are listed under **cassandraCluster** for Cassandra 4.1.

native_transport_port

Default Value: 9042

Valid Values: A valid port

Mandatory: No

Specifies the port used by CQL. This is required if not using the default port.

dataCenterName

Default Value: datacenter1

Valid Values: Name of the primary datacenter used by the new CQL driver

Mandatory: No

Name of the primary datacenter used by the new CQL driver. The required value can be found by running `nodetool status` on the Cassandra instance. This is required if not using the default datacenter name.

backupDataCenterName

Default Value: datacenter1

Valid Values: Name of the secondary datacenter used by the new CQL driver if there is a secondary datacenter

Mandatory: No

Name of the secondary datacenter used by the new CQL driver if there is a secondary datacenter. The required value can be found by running `nodetool status` on the Cassandra instance. This is required if using a Multi datacenter Cassandra. and not using the default secondary datacenter name.

switchOverUnHealthyNodes

Default Value: -1

Valid Values: The number of nodes in the primary Cassandra datacenter that need to fail before RWS will attempt to switch to the backup datacenter.

Mandatory: No

The number of nodes in the primary Cassandra datacenter that need to fail before RWS will attempt to switch to the backup datacenter. The default value is -1, meaning that no attempt to switchover will be made. This is required only if using a Multi datacenter Cassandra.

jmx_port

Default Value: 7199

Valid Values: A valid port

Mandatory: No

Specifies the port Cassandra uses for Java Manage Extension (JMX).

keyspace

Default Value: sipfs

Valid Values: A valid keyspace name

Mandatory: Yes

Specifies the name of the Cassandra keyspace. This name should be the same as the keyspace name you set while [Creating the Cassandra keyspace](#). If you used the keyspace creation scripts that come with Interaction Recording Web Services, then you can leave this value as sipfs.

nodes

Default Value: None

Valid Values: A comma-separated list of IP addresses or host names

Mandatory: Yes

Specifies the Cassandra node IP addresses or host names.

backup_nodes

Default Value: None

Valid Values: A comma-separated list of IP addresses or host names

Mandatory: No

Specifies the backup Cassandra node IP addresses or host names. This option is intended for deployments that have two separate Cassandra data centers — Interaction Recording Web Services switches from primary to backup if the primary nodes are unavailable. If your deployment is small with only one data center, you can ignore this option.

replication_factor

Default Value: None

Valid Values: An integer less than or equal to the number of nodes in the cluster

Mandatory: Yes

Specifies a replication factor appropriate for your Cassandra topology. This value must be the same as the replication factor you set in Step 2 of the [Creating the Cassandra keyspace](#) procedure.

read_consistency_level

Default Value: None

Valid Values: CL_ONE, CL_QUORUM, CL_LOCAL_QUORUM

Mandatory: Yes

Specifies the read consistency level appropriate for your Cassandra topology:

Development (1 Cassandra node)	Single Datacenter (1 datacenter with a minimum of three Cassandra nodes)	Two Datacenters (datacenters with a minimum of three Cassandra nodes in each datacenter)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

write_consistency_level

Default Value: None

Valid Values: CL_ONE, CL_QUORUM, CL_LOCAL_QUORUM

Mandatory: Yes

Specifies the write consistency level appropriate for your Cassandra topology:

Development (1 Cassandra node)	Single Datacenter (1 datacenter with a minimum of three Cassandra nodes)	Two Datacenters (datacenters with a minimum of three Cassandra nodes in each datacenter)
CL_ONE	CL_QUORUM	CL_LOCAL_QUORUM

max_conns_per_host

Default Value: 16

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of connections to allocate for a single host's pool.

max_cons

Default Value: 48

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of connections in the pool.

max_pending_conns_per_host

Default Value: 80

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of pending connection attempts per host.

max_blocked_threads_per_host

Default Value: 160

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of blocked clients for a host.

cassandraVersion

Default Value: 4.1.4

Valid Values: Cassandra version

Mandatory: No

Specifies the Cassandra version for your Interaction Recording Web Services deployment.

useSSL

Default Value: None

Valid Values: true, false

Mandatory: No

Specifies whether Cassandra should use Secure Sockets Layer (SSL). This option is only valid for Cassandra versions 1.2.x and higher.

truststore

Default Value: None

Valid Values: A valid path.

Mandatory: No

Specifies the path to the truststore.

truststorePassword

Default Value: None

Valid Values: A valid password.

Mandatory: No

Specifies the password for the truststore.

userName

Default Value: None

Valid Values: A valid Cassandra username.

Mandatory: No

Specifies the username if Cassandra is configured to use authentication.

password

Default Value: None

Valid Values: A valid Cassandra password.

Mandatory: No

Specifies the password if Cassandra is configured to use authentication.

serverSettings

Settings in this section are listed under **serverSettings**.

URLs

externalApiUrlV2

Default Value: None

Valid Values: A public schema-based URL ending with /api/v2.

Mandatory: Yes

Specifies the prefix used for resources in the public API. In a development environment, the host and port should be set to the host name or IP address of the Interaction Recording Web Services node. In a production environment, the host and port should be set to the host name or IP address of the load balancer in a production environment. For example, `https://192.0.2.20/api/v2`.

`internalApiUriV2`

Default Value: None

Valid Values: A public schema-based URL ending with `/internal-api`.

Mandatory: Yes

Specifies the prefix used for internal resources. In a development environment, the host and port should be set to the host name or IP address of the Interaction Recording Web Services node. In a production environment, the host and port should be set to the host name or IP address of the load balancer in a production environment. For example, `http://192.0.2.20/internal-api`.

`undocumentedExternalApiUri`

Default Value: None

Valid Values: A public schema-based URL ending with `/internal-api`.

Mandatory: Yes

Specifies the reachable Interaction Recording Web Services server address for the SpeechMiner UI and the Screen Recording Service. For example, `http://192.0.2.20:8090/internal-api`

Paths

`pathPrefix`

Default Value:

Valid Values: A valid prefix

Mandatory: No

Specifies a prefix that Interaction Recording Web Services adds to the relative URIs it includes in responses. For example, if you set **pathPrefix** to `/api/v2` and make the following request:

```
GET http://localhost:8080/api/v2/devices
```

Interaction Recording Web Services returns the following response:

```
{
  "statusCode":0,
  "paths":[
    "/api/v2/devices/971ed91d-82bf-490b-94d2-02d240165764",
    "/api/v2/devices/a3f9e854-54d8-4260-bea3-d6e450ee7df0"
  ],
  "uris":[
    "http://localhost:8080/api/v2/devices/7c7ab1f7-e596-41bc-9ff4-4a12c489865f",
    "http://localhost:8080/api/v2/devices/a3f9e854-54d8-4260-bea3-d6e450ee7df0"
  ]
}
```

Notice that paths includes relative URIs with the `/api/v2` prefix.

`internalPathPrefix`

Default Value: Empty

Valid Values: A valid prefix

Mandatory: No

Specifies a prefix that Interaction Recording Web Services adds to the relative URIs that it includes in responses to internal APIs. See `pathPrefix` for details.

General

`temporaryAuthenticationTokenTTL`

Default Value: 300

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the time to live, in seconds, for the temporary authentication token.

`enableCsrfProtection`

Default Value: false

Valid Values: true, false

Mandatory: No

Enables cross site request forgery protection. If you set the value to `true`, make sure you use the default values for **exposedHeaders** in the `crossOriginSettings` option. If you have already updated the **exposedHeaders**, just make sure the values include the defaults.

Important

If CSRF protection is enabled, then the label/tagging and deletion prevention functionality cannot be used in SpeechMiner, as SpeechMiner does not support CSRF.

Timeouts

`activationTimeout`

Default Value: 12000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for connecting to any Genesys server (except Configuration Server). This may include several individual attempts if the initial attempt to connect is unsuccessful.

Important

The activation timeout for Configuration Server is specified with the **configServerActivationTimeout** option.

configServerActivationTimeout

Default Value: 35000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for connecting to Configuration Server. This may include several individual attempts if the initial attempt to connect is unsuccessful.

configServerConnectionTimeout

Default Value: 15000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for an individual connection attempt to Configuration Server.

connectionTimeout

Default Value: 4000

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the timeout, in milliseconds, for an individual connection attempt to any Genesys server (except Configuration Server).

Important

The connection timeout for Configuration Server is specified with the **configServerConnectionTimeout** option.

inactiveUserTimeout

Default Value: 60

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the interval, in seconds, at which the inactive user cleanup process is run by the server. This process is run to invalidate HTTP sessions for users who have been deleted or whose user roles have changed.

reconnectAttempts

Default Value: 1

Valid Values: An integer greater than 0.

Mandatory: Yes

Specifies the number of attempts Interaction Recording Web Services makes to connect to any Genesys server before attempting to connect to the backup.

reconnectTimeout

Default Value: 10000

Valid Values: An integer greater than 0.

Mandatory: Yes

Specifies the timeout, in milliseconds, between the reconnect attempts.

OPS account

opsUserName

Default Value: None

Valid Values: Any alphanumeric value that can include special characters

Mandatory: Yes

Specifies the name of the Interaction Recording Web Services super user. Interaction Recording Web Services creates or updates this user at startup if **opsCredentials** is set to `true` in the **updateOnStartup** section of the **application.yaml** file.

opsUserPassword

Default Value: None

Valid Values: Any alphanumeric value, including special characters

Mandatory: Yes

Specifies the password for the Interaction Recording Web Services super user. Interaction Recording Web Services creates or updates the password for the **ops** user at startup if **opsCredentials** is set to `true` in the **updateOnStartup** section of the **application.yaml** file.

CME credentials

applicationName

Default Value: None

Valid Values: A valid application name

Mandatory: Yes

The name of the Interaction Recording Web Services node application object in Configuration Server. For example, `IRWS_Node`.

applicationType

Default Value: None

Valid Values: A valid application type

Mandatory: Yes

The type of the Interaction Recording Web Services node application object in Configuration Server. This value should be `CFGGenericClient`.

cmeUserName

Default Value: None

Valid Values: A valid Configuration Server user

Mandatory: Yes

The username that the Interaction Recording Web Services server uses to connect to Configuration Server.

Important

Genesys recommends that you use the provided "default" account in Configuration Server. It is possible to use a different account, but you must take care in configuring the user's account permissions. Outside of a lab setting, this is best done in consultation with Genesys.

cmePassword

Default Value: None

Valid Values: A valid password

Mandatory: Yes

The password for the Configuration Server user Interaction Recording Web Services uses to connect to Configuration Server.

syncNode

Default Value: false

Valid Values: true, false

Mandatory: Yes (for sync node), No (all other nodes)

Specifies whether the node is the synchronization node. This node is responsible for importing objects from Configuration Server into Cassandra, subscribing to change notifications with Configuration Server, and processing updates.

Important

In each Interaction Recording Web Services cluster or shared Interaction Recording Web Services and Web Services and Applications cluster, if both are deployed, one node in the cluster must be configured as the synchronization node: `syncNode: true`. All other nodes in the cluster must have `syncNode: false`.

ConfigServer String Encoding

configServerDefaultEncoding

Default Value: windows-1252

Valid Values: A valid java string encoding

Mandatory: No

The configuration server can be installed in one of two modes. One mode uses UTF-8 for encoding strings; the other uses the default character encoding of the machine that the configuration server is installed on. If you are using UTF-8, RWS will communicate using UTF-8 and this parameter is not used. If you are not using UTF-8, this value should be set to the value of the default string encoding of the machine that the configuration server is installed on.

Call Recording

createCallRecordingCF

Default Value: false

Valid Values: true, false

Mandatory: No

Specifies if the column families required for Call Recording will be created when a contact center is created.

crClusterName

Default Value: None

Valid Values: A valid cluster name

Mandatory: Yes

Specifies the name of the cluster to enable search functionality in Elasticsearch. The value must be the same for all Interaction Recording Web Services nodes in the cluster and must match the **cluster.name** parameter configured in **elasticsearch.yml** for each Elasticsearch node. For example, if there are five nodes in the Interaction Recording Web Services cluster, all five nodes must have the same value as in the **crClusterName** option.

crRegion

Default Value: None

Valid Values: String

Mandatory: Yes

Specifies the name of the region where the Interaction Recording Web Services node is located. Ensure that this value is the same on all RWS nodes.

cryptoSecurityKey

Default Value: None

Valid Values: A valid security key

Mandatory: Yes

Specifies the security key used for encryption for call recording settings stored in the database. The value must be the same for all Interaction Recording Web Services nodes in the cluster. For example, if there are five nodes in the Interaction Recording Web Services cluster, all five nodes must have the same value as in the **cryptoSecurityKey** option.

webDAVMaxConnection

Default Value: 50

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of WebDAV client TCP connections to each route. When the number of WebDAV client requests to the same WebDAV server are less than this value, a new TCP connection is established for better performance. Otherwise, the new request is queued until any ongoing request finishes.

webDAVMaxTotalConnection

Default value: 10 * value of [webDAVMaxConnection](#)

Valid Values: An integer greater than 0.

Mandatory: No

Specifies the maximum number of TCP connections from the Interaction Recording Web Services node to all WebDAV storage.

Multi regional supporting

nodePath

Default Value: None

Valid Values: A location and node ID, separated by a "/" — for example, /US/node1

Mandatory: Yes

Specifies the location and ID of the Interaction Recording Web Services node within the deployment topology. This value must be unique across the deployment. For example, a value of /US/node1 means that the node is located in the US region and has an ID of "node1". The node ID can be the hostname, the IP address, or any other unique identifier.

nodeId

Default Value: None

Valid Values: Any unique identifier, such as the node host name or IP address

Mandatory: No

Specifies the unique identifier for the Interaction Recording Web Services node. Each node in a cluster must have a unique nodeId.

SSL and CA

caCertificate

Default Value: None

Valid Values: Path to a signed certificate or empty

Mandatory: No

Specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if .jks, you can also set [jksPassword](#)). The certificate will be used if the IRWS_Cluster application uses [Transport Layer Security \(TLS\)](#) to connect to the Configuration Server, SIP Server, and Interaction Server. If left empty, or if the parameter is not specified, the certificates returned from the servers will not be validated.

jksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in [caCertificate](#), when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see [Password Encryption](#).

webDAVTrustedCA

Default Value: true

Valid Values: true, false, or a path to a file containing a certificate for a Certificate Authority

Mandatory: No

When using a secure connection to the WebDAV Server, controls whether the certificate is validated, and how. If set to false, the certificate presented by the WebDAV Server will not be validated. If set to true, the certificate presented by the WebDAV Server will be validated by **caCerts** in \$JAVA_HOME/jre/lib/security. Otherwise, specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if it is in .jks format, you can also set **webDAVJksPassword**).

webDAVJksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in **webDAVTrustedCA** when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see **Password Encryption**.

rcsTrustedCA

Default Value: true

Valid Values: true, false, or a path to a file containing a certificate for a Certificate Authority

Mandatory: No

When using a secure connection to the Recording Crypto Server, controls whether the certificate is validated, and how. If set to false, the certificate presented by RCS will not be validated. If set to true, the certificate presented by RCS will be validated by **caCerts** in \$JAVA_HOME/jre/lib/security. Otherwise, specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if it is in .jks format, you can also set **rcsJksPassword**).

rcsJksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in **rcsTrustedCA** when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see **Password Encryption**.

speechMinerTrustedCA

Default Value: true

Valid Values: true, false, or a path to a file containing a certificate for a Certificate Authority

Mandatory: No

When using a secure connection to the SpeechMiner Interaction Receiver, controls whether the certificate is validated, and how. If set to false, the certificate presented by the SpeechMiner Interaction Receiver will not be validated. If set to true, the certificate presented by the SpeechMiner Interaction Receiver will be validated by **caCerts** in \$JAVA_HOME/jre/lib/security. Otherwise, specifies the path to a file containing a certificate for a Certificate Authority. The file must be in the .pem or .jks format (if it is in .jks format, you can also set **speechMinerJksPassword**).

speechMinerJksPassword

Default Value: None

Valid Values: Password for the key storage

Mandatory: No

Specifies the password for the key storage set in [speechMinerTrustedCA](#) when the certificate is in .jks format. You can specify an encrypted password. For more information on encrypting a password, see [Password Encryption](#).

CORS

crossOriginSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
allowedOrigins	No	None	Specifies a comma-separated list of allowed origins supported by this Interaction Recording Web Services node. For example, http://*.genesys.com , http://*.genesyslab.com
allowedMethods	No	GET,POST,PUT,DELETE,OPTIONS	Specifies a comma-separated list of HTTP methods supported by the server.
allowedHeaders	No	X-Requested-With,Content-Type,Accept,Origin,Cookie,X-CSRF-TOKEN,authorization,ssid,surl,ContactCenterId	Specifies whether to include the Access-Control-Allow-Headers header as part of the response to a pre-flight request. This specifies which header field names can be used during the actual request.
allowCredentials	No	true	Specifies the value of the Access-Control-Allow-Credentials header. This should typically be left at the default value.
corsFilterCacheTimeToLive	No	120	Specifies for how long (in seconds) the cross origin settings are cached before being reloaded.
exposedHeaders	No	X-CSRF-HEADER,X-CSRF-TOKEN	Specifies which custom headers are allowed in

Name	Mandatory	Default Value	Description
			cross-origin HTTP responses. This should typically be left at the default value. If you do modify the value and you enable the enableCsrfProtection option, make sure the value for exposedHeaders includes X-CSRF-HEADER, X-CSRF-TOKEN.

Mandatory: No

Specifies the configuration for cross-origin resource sharing in Interaction Recording Web Services. For example:

```
...
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: http://*.genesys.com, http://*.genesyslab.com
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId"
  allowCredentials: true
  exposedHeaders: "X-CSRF-HEADER,X-CSRF-TOKEN"
```

Elasticsearch

elasticSearchSettings

Default Value: None**Valid Values:**

Name	Mandatory	Default Value	Description
retriesOnConflict	No	3	Controls how many times to retry if there is a version conflict when updating a document.
waitToIndexTimeout	No	5000	Specifies the length of time (in milliseconds) that the Interaction Recording Web Services will wait while Elasticsearch is indexing data.
scanReadTimeoutSeconds	No	60	Specifies the length of time that the Interaction Recording Web Services waits for Elasticsearch to return results from setting up a scan and

Name	Mandatory	Default Value	Description
			scroll search request.
countReadTimeoutSecondsNo		60	Specifies the length of time that the Interaction Recording Web Services waits for Elasticsearch to return results from a count search request.
scrollTimeoutSeconds	No	240	Specifies how long Elasticsearch should keep the Search Context alive when handling scan and scroll requests from the Muxer and MLM components. This value must be long enough to process each batch of results. However, it does not need to be long enough to process all data. You can change this value based on the performance results in your environment.
useTransportClient	No	true	Specifies whether Interaction Recording Web Services should use a transport client for Elasticsearch. This is only applicable for Elasticsearch 1.x.
transportClient	Yes, if useTransportClient is true.	Values specified in TransportClientSettings	Specifies the configuration Interaction Recording Web Services should use for the transport client. For details see TransportClientSettings in the next table. This is only applicable for Elasticsearch 1.x.
useRestClient	no	false	Specifies whether Interaction Recording Web Services should use a REST client for Elasticsearch. This is only applicable for Elasticsearch 7.16.3 or higher.
restClient	Yes, if useRestClient is true.	Values specified in RestClientSettings .	Specifies the configuration Interaction Recording Web Services should use for the REST

Name	Mandatory	Default Value	Description
			client. For details, see RestClientSettings in the following tables. This is only applicable for Elasticsearch 7.16.3 or higher.

TransportClientSettings

Name	Mandatory	Default Value	Description
nodes	Yes, if useTransportClient is true.	null	Specifies the list of Elasticsearch nodes the transport client should connect to.
useSniff	no	false	Specifies if the transport client should use sniffing functionality and perform auto-discovery of Elasticsearch nodes in the cluster.
ignoreClusterName	no	false	Specifies if Interaction Recording Web Services should ignore the name of the cluster when connecting to the cluster.
pingTimeout	no	5000	Specifies, in milliseconds, the ping timeout for Elasticsearch nodes.
nodesSamplerInterval	no	5000	Specifies, in milliseconds, how often Interaction Recording Web Services should sample/ping the Elasticsearch nodes listed and connected.

Mandatory: No

Specifies the configuration for Elasticsearch in Interaction Recording Web Services. For example:

```
...
elasticSearchSettings:
  retriesOnConflict: 2
  waitToIndexTimeout: 5000
  useTransportClient: true
  transportClient:
    nodes:
      - {host: 127.0.0.1, port: 9300}
    useSniff: true
    ignoreClusterName: true
    pingTimeout: 10000
```

```

nodesSamplerInterval: 10000
scanReadTimeoutSeconds: 60
countReadTimeoutSeconds: 60
scrollTimeoutSeconds: 240

```

RestClientSettings

Name	Mandatory	Default Value	Valid Values	Description
nodes	Yes, if useRestClient is true.	null	A comma-separated list of IP addresses or host names as given in sample.	Specifies the list of Elasticsearch nodes the REST client should connect to.
authentication.type	No	none	none, basic	Authentication method.
authentication.userName	Yes, if Elasticsearch authentication type is basic .	None	A valid Elasticsearch username.	Specifies the username if Elasticsearch is configured to use authentication.
authentication.password	Yes, if Elasticsearch authentication type is basic .	None	A valid Elasticsearch password.	Specifies the password if Elasticsearch is configured to use authentication.
useSSL	No	None	true,false	Specifies whether Elasticsearch should use Secure Sockets Layer (SSL). This option is only valid for Elasticsearch versions 8.18.2 and higher.
truststore	No	None	A valid path.	Specifies the path to the truststore.
truststorePassword	No	None	A valid password.	Specifies the password for the truststore.

Mandatory: No

Specifies the configuration for Elasticsearch in Interaction Recording Web Services. For example:

```

...
elasticSearchSettings:
  retriesOnConflict: 3
  useRestClient: true
  restClient:
    nodes: [ToBeChanged: <ELASTIC_SEARCH_REST_NODES>]
    authentication:
      type: [ToBeChangedOrRemoved: none|basic]
      userName: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_USER_NAME>]
      password: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_PASSWORD>]
    useSSL: [ToBeChangedOrRemoved: "true"|"false"]

```

```

truststore: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PATH>]
truststorePassword: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_TRUSTSTORE_PASSWORD>]
waitToIndexTimeout: 5000
scanReadTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCAN_READ_TIMEOUT_SECONDS>]
scrollTimeoutSeconds: [ToBeChangedOrRemoved: <ELASTIC_SEARCH_SCROLL_TIMEOUT_SECONDS>]
countReadTimeoutSeconds: [ToBeChangedOrRemoved:
<ELASTIC_SEARCH_COUNT_READ_TIMEOUT_SECONDS>]

```

Recording

recordingSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
auditLogDeletedFiles	No	None	If set to true, Interaction Recording Web Services generates an audit log for each individual recording file that is deleted.
recordCryptoServerDecryptMaxConnection	Yes	50	Specifies the maximum TCP connections to each Recording Crypto Server instance defined in local-decrypt-uri-prefix settings. See Screen Recording Local Decrypt URI Prefix for details.
recordCryptoServerDecryptMaxTotalConnection	Yes	10 * recordCryptoServerDecryptMaxConnection	Specifies the maximum TCP connections to all Recording Crypto Server instances defined in local-decrypt-uri-prefix settings. See Screen Recording Local Decrypt URI Prefix for details.
recordCryptoServerDecryptSocketTimeout	Yes	30000	Specifies the socket timeout, in milliseconds, for TCP connections to Recording Crypto Server instances defined in local-decrypt-uri-prefix settings. See Screen Recording Local Decrypt URI Prefix for details.
keyspaceNameSettingsCacheSecondsTTL	No	300	Specifies the time to live in seconds of the keyspace name for a given contact center and location from

Name	Mandatory	Default Value	Description
			Cassandra in a cache.
regionsSettingsCacheSecondsTTL	No	300	Specifies the time to live in seconds of a regions setting for a location stored in Cassandra in a cache.

Mandatory: No

Specifies the configuration for recording in Interaction Recording Web Services. For example:

```
recordingSettings:
  auditLogDeletedFiles: true
  recordCryptoServerDecryptMaxConnection: 50
  recordCryptoServerDecryptMaxTotalConnection: 500
  recordCryptoServerDecryptSocketTimeout: 30000
  regionsSettingsCacheSecondsTTL: 300
```

Screen Recording

screenRecordingSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
enableSameSiteCookieForScreenRecordingPlayback	No	false	Specifies whether Interaction Recording Web Services will return the SameSite=None and Secure cookie attributes on the cookie used when playing back screen recordings from the SpeechMiner browser application. Important: Before enabling this option, ensure that the connection between the SpeechMiner browser application and RWS is configured to use HTTPS. If you set this option to true and are using HTTP, the cookie will not be returned by the browser.
screenRecordingVoiceEnabled	No	false	Specifies whether the current Interaction Recording Web Services node supports screen recording for voice interactions. If set to false, the node rejects CometD requests from

Name	Mandatory	Default Value	Description
			the Screen Recording Service for agents with the voice channel.
screenRecordingEServicesEnabled	Enabled	false	Specifies whether the current Interaction Recording Web Services node supports screen recording for non-voice interactions. If set to false, the node rejects CometD requests from the Screen Recording Service for agents with the eServices channel.
recordingInteractionEventsTTL	TTL	172800	Specifies the time to live (TTL) for Cassandra to cache a screen recording interaction event.
clientSessionManagerCacheTTL	TTL	60	Specifies the TTL for the Interaction Recording Web Services node to cache agent information (such as the agent's name) so that the node doesn't have to read the information from Interaction Recording Web Services on each request.
contactCenterInfoManagerCacheTTL	Cache TTL	90	Specifies the TTL for the Interaction Recording Web Services node to cache contact center information so that the node doesn't have to read the information from Interaction Recording Web Services on each request.

Mandatory: No

Specifies the screen recording configuration parameters. For example:

```
...
screenRecordingSettings:
  enableSameSiteCookieForScreenRecordingPlayback: false
  screenRecordingVoiceEnabled: false
  screenRecordingEServicesEnabled: false
  recordingInteractionEventsTTL: 172800
  clientSessionManagerCacheTTL: 60
  contactCenterInfoManagerCacheTTL: 90
```

Screen Recording Connections Reporting

reportingEnabled

Default Value: false

Valid Values: true, false

Mandatory: No

Enables the Screen Recording Connection Reporting feature.

createReportingCF

Default Value: false

Valid Values: true, false

Mandatory: No

Specifies if the column families required for Screen Recording Connection reporting will be created when a contact center is created.

connectionInfoHoursTTL

Default Value: 7 * 24

Valid Values: Integer

Mandatory: No

Specifies the Time To Live (in hours) to use when writing columns to the src_rep_node_<id> column family. Screen Recording Service (SR Service) connections older than the Time to Live will not be listed in the SR Service connection information queries.

historyCountsMinutesTTL

Default Value: 24 * 60

Valid Values: Integer

Mandatory: No

Specifies the Time To Live (in minutes) to use when writing columns to the src_rep_hist_<id> column family. This number determines the maximum number of values that can be reported for a statistic in historic count queries.

Multimedia Disaster Recovery

drMonitoringDelay

Default Value: 1800

Valid Values: Integer

Mandatory: No

Specifies the interval (in seconds) that will be used for monitoring Disaster Recovery synchronization.

Caching

cachingSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
contactCenterFeaturesTTL	No	30	The TTL, in seconds, for contact-center feature IDs in cache.
contactCenterSettingsTTL	No	30	The TTL, in seconds, for contact-center custom settings in cache.

Mandatory: No

Specifies how Interaction Recording Web Services should handle various caching scenarios. For example:

```
...
cachingSettings:
  contactCenterFeaturesTTL: 30
  contactCenterSettingsTTL: 30
```

DoS Filter

enableDosFilter

Default Value: false

Valid Values: true, false

Mandatory: No

Enables the denial of service filter. If you set the value to true, you must also set values for the [dosFilterSettings](#) option.

dosFilterSettings

Default Value: None

Valid Values:

Name	Mandatory	Default Value	Description
maxRequestsPerSec	No	25	Specifies the maximum number of requests from a connection per second. Requests that exceed this are first delayed, then throttled.
delayMs	No	100	Specifies the delay, in milliseconds, imposed on all requests over the rate limit, before they are considered at all. Valid values: <ul style="list-style-type: none"> • -1 = reject request • 0 = no delay • Any other number = delay in milliseconds

Name	Mandatory	Default Value	Description
maxWaitMs	No	50	Specifies the length of time, in milliseconds, to blocking wait for the throttle semaphore.
throttledRequests	No	5	Specifies the number of requests over the rate limit that are able to be considered at once.
throttleMs	No	30000	Specifies the length of time, in milliseconds, to asynchronously wait for semaphore.
maxRequestMs	No	30000	Specifies the length of time, in milliseconds, to allow the request to run.
maxIdleTrackerMs	No	30000	Specifies the length of time, in milliseconds, to keep track of request rates for a connection, before deciding that the user has gone away, and discarding the connection.
insertHeaders	No	true	If set to true, DoSFilter headers are inserted into the response.
trackSessions	No	true	If set to true, the usage rate is tracked by session if a session exists.
remotePort	No	false	If set to true and session tracking is not used, then the rate is tracked by IP address + port (effectively connection).
ipWhitelist	No	""	A comma-separated list of IP addresses that is not rate limited.

Mandatory: No

Specifies how Interaction Recording Web Services should handle denial of service. For example:

```
...
enableDosFilter: true
dosFilterSettings:
  maxRequestsPerSec: 30
  ipWhitelist: 192.168.0.1,192.168.0.2
```

These options only take effect if **enableDosFilter** is set to true.

multiPartResolverMaxUploadSize**Default Value:** 536870912**Valid Values:** Integer**Mandatory:** Yes

This parameter should be aligned with `maxDurationMinutes`, so if you change its value, ensure that you also consider the `maxDurationMinutes` value specified within the Advanced Configuration for the Screen Recording Service section in the [Deploying the Screen Recording Service - Advanced Configuration](#) page. The maximum size of a file that can be uploaded by the Screen Recording Service must be less than or equal to the `multiPartResolverMaxUploadSize`.

multiPartResolverMaxInMemorySize**Default Value:** 67108864**Valid Values:** Integer**Mandatory:** Yes

Specifies the maximum allowed size (in bytes) before uploads are written to disk.

Media Life Cycle management**backgroundScheduledMediaOperationsSettings****Default Value:** None**Valid Values:**

Name	Mandatory	Default Value	Description
<code>enableBackgroundScheduledMediaOperations</code>	No	false	Specifies whether this Interaction Recording Web Services node can perform MLM operations.
<code>schedulerThreads</code>	No	4	Specifies the number of scheduler worker threads.
<code>schedulePollingInterval</code>	No	60	Specifies how often, in seconds, Interaction Recording Web Services polls for gir-scheduler settings and synchronizes the rule schedule.
<code>speechMinerMaxConnection</code>	No	20	Specifies the maximum number of concurrent TCP connections for the same route when Interaction Recording Web Services issues API requests to SpeechMiner.
<code>speechMinerMaxTotalConnection</code>	No	-1	Specifies the size of the connection pool when

Name	Mandatory	Default Value	Description
			Interaction Recording Web Services issues API requests to SpeechMiner. If the value of this option is less than 1, Interaction Recording Web Services sets the size of the pool to the value <code>speechMinerMaxConnection * 10</code> .
<code>speechMinerSocketTimeout</code>	No	60000	Specifies how long Interaction Recording Web Services should wait, in milliseconds, for the SpeechMiner API response before timing out.
<code>defaultBackupExportURI</code>	No	None	Specifies the location to store backed up recordings. For example, file:///tmp/archLocDefault.
<code>useFullPathInMediaFileBackup</code>	No	false	Specifies whether to include the full path or file name only during an MLM backup operation
<code>enableScanAndScroll</code>	No	false	Specifies whether to turn on the feature where MLM uses Elasticsearch scan and scroll queries to determine the recording IDs on which to act.
<code>scanIntervalsPerDay</code>	No	24	When MLM is configured to use Elasticsearch scan and scroll queries to determine the recording IDs on which to act, this parameter determines the number of scan intervals used in a day of recordings. Reduce this value to reduce the number of Elasticsearch scan queries performed by an MLM Task for its work, assuming that all other things remain equal. Reducing this value also increases the lifetime of the search context

Name	Mandatory	Default Value	Description
			<p>created by each Elasticsearch scan query, which in turn increases the number of open file descriptors in use by Elasticsearch.</p> <p>Note: When configuring, ensure that the number of seconds in a day (i.e. 24 * 60 * 60) is exactly divisible by the configured value.</p>

Mandatory: No

Specifies the configuration for Interaction Recording Web Services to schedule purge and backup events. For example:

```
backgroundScheduledMediaOperationsSettings:
  enableBackgroundScheduledMediaOperations: true
  schedulerThreads: 4
  schedulePollingInterval: 60
  speechMinerMaxConnection: 20
  speechMinerMaxTotalConnection: -1
  speechMinerSocketTimeout: 60000
  defaultBackupExportURI:
  useFullPathInMediaFileBackup: false
  enableScanAndScroll: true
  scanIntervalsPerDay: 24
```

CometD

cometDSettings

Default Value: None**Valid Values:**

Name	Mandatory	Default Value	Description
cometdSessionExpirationTimeout		60	Specifies the timeout for the CometD session to expire on disconnect. It might take an additional minute for the session to be closed after it expires. If you set this option to -1, the session never expires. An agent can log in again before the end of this timeout to disable session expiration.
closeHttpSessionOnCometDExpiration		true	Enables or disables HTTP session invalidation when CometD times out.

Name	Mandatory	Default Value	Description
maxSessionsPerBrowser	No	1	Specifies the maximum number of sessions (tabs/frames) allowed to long poll from the same browser; a negative value allows unlimited sessions.
multiSessionInterval	No	2000	Specifies the period of time, in milliseconds, for the client normal polling period, in case the server detects more sessions (tabs/frames) connected from the same browser than allowed by the maxSessionsPerBrowser parameter. A non-positive value means that additional sessions will be disconnected.

Mandatory: No

Specifies the configuration for the CometD-specific transport server embedded into the Interaction Recording Web Services application. For example:

```
cometDSettings:
  cometdSessionExpirationTimeout: 60
  closeHttpSessionOnCometDExpiration: true
  maxSessionsPerBrowser: 2
  multiSessionInterval: 4000
```

Log header

enableLogHeader

Default Value: false

Valid Values: true, false

Mandatory: No

Specifies whether Interaction Recording Web Services includes a header in its main log file. This header contains key information about the Interaction Recording Web Services installation, including the version, start time, libraries, and any applicable settings from the **application.yaml** file.

updateOnPremiseInfoInterval

Default Value: 600

Valid Values: Integer

Mandatory: No

Specifies a period (in seconds) during which the premise environment log header information is updated.

updateOnStartup

opsCredentials

Default Value: false

Valid Values: true, false

Mandatory: No

Changes take effect: When the Interaction Recording Web Services server is started or restarted. Specifies whether to update the stored ops credentials to the values specified in the **opsUserName** and **opsUserPassword** parameters.

features

Default Value: false

Valid Values: true, false

Mandatory: No

Changes take effect: When the Interaction Recording Web Services server is started or restarted. Specifies whether to update the supported features to the list specified in the **feature-definitions.json** file. See [Enabling features in the Feature Definitions file](#) for details.

onPremiseSettings

Settings in this section are listed under **onPremiseSettings**.

Important

- The following settings should be specified for the sync node (syncNode: true). They are not required on the other nodes in the cluster.
- Note that settings under **onPremiseSettings** are used only once during the first initialization of RWS on the sync node. Further changes in the environment are retrieved from the Configuration Server directly. If a setting is configured incorrectly, please contact Genesys Customer Care for support.

cmeHost

Default Value: None

Valid Values: A valid IP address or host name

Mandatory: Yes (for sync node), No (all other nodes)

Specifies the Configuration Server host name (FQDN) or IP address.

cmePort

Default Value: None

Valid Values: A valid port

Mandatory: Yes (for sync node), No (all other nodes)
Specifies the Configuration Server port.

backupCmeHost

Default Value: None

Valid Values: A valid IP address or host name

Mandatory: Yes (for sync node), No (all other nodes)

Specifies the backup Configuration Server host name (FQDN) or IP address. You should only configure this option if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

backupCmePort

Default Value: None

Valid Values: A valid port

Mandatory: Yes (for sync node), No (all other nodes)

Specifies the backup Configuration Server port. You should only configure this option if there is a backup Configuration Server in the Genesys environment and you want high-availability support.

countryCode

Default Value: None

Valid Values: A two-letter country code

Mandatory: Yes (for sync node), No (all other nodes)

The premise contact center's country code. For example, "US".

tlsEnabled

Default Value: None

Valid Values: true, false

Mandatory: Yes (for sync node), No (all other nodes)

Specifies whether Interaction Recording Web Services should use a secure connection to the Configuration Server.

Deploying Web Services and Applications for GIR

Warning

- The content on this page only applies to version 8.5.210.02 or earlier of Genesys Interaction Recording. If you're using a later version, you'll need to install the Interaction Recording Web Services component instead. See [Deploying Interaction Recording Web Services](#) for details.
- If you upgrade to Interaction Recording Web Services, it does not provide API support for non-GIR related Web Services, such as Workspace Web Edition.

Genesys Interaction Recording (GIR) needs Web Services to store and manage the recording files.

Web Services uses the following major components:

- [WebDAV Server](#)—The file management device that stores and manages the GIR recording files
- [Cassandra Database](#)—The java-based cluster-schemed database for Web Services to store interaction metadata.
- [Web Services Server](#)—A REST API server that pushes and pulls the interaction metadata to and from the Cassandra database.
- [Workspace Web Edition](#)—The web-based Agent Desktop.

Important

- Screen Recording for voice interactions requires that Agent Info specify the default agent place in the Default Place field. The default place must be assigned at least one DN. If a DN is not assigned to the default place, the agent will not be associated with a device and Screen Recording will not produce interactions for the agent.
- The 8.5.201.29 release of Genesys Web Services and Applications is not supported with the Genesys Interaction Recording Solution.
- The URI in recording_settings is case sensitive and must match the URI in the IVR Profile. For example:

```
"uri": "http://GENESYSREC1/recordings"
```

is not the same as:

```
"uri": "http://genesysrec1/recordings"
```

The following steps describe how to deploy the Web Services components for GIR.

Deploy the WebDAV Storage Server

1. Install WebDAV, by running the following command:

```
yum install httpd
```

2. Edit the `/etc/httpd/conf/httpd.conf` file, and append the following to the end of the file:

```
Alias /recordings /mnt/recordings
<Directory /mnt/recordings>
    Options Indexes MultiViews FollowSymLinks
    EnableSendfile off
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
<Location "/recordings">
    DAV On
    AuthType Basic
    AuthName "user"
    AuthUserFile /var/www/htpasswd
    Require valid-user
</Location>
```

3. Open the firewall. Because WebDAV is an HTTP server, the incoming default HTTP and/or HTTPS ports (80 and/or 443) must be open to the server.

Important

It is possible to use custom ports by changing the permitted incoming ports in the firewall, the virtual host configuration file, and the URL used to reach the WebDAV server.

4. Create the directory to keep the recording files, and set the permission to Apache, using the following command:

```
mkdir /mnt/recordings
chown apache:apache /mnt/recordings
```

Important

Due to performance concerns, Genesys does not recommend using a remote directory for WebDAV.

5. Create a WebDAV user for httpd, and configure the password. The following example creates a user called "user":
`htpasswd -c /var/www/htpasswd user`

Warning

If the Recording Muxer is deployed for screen recording, make sure all WebDAV storages of the same contact center region are using the same username and password.

6. Configure the httpd to start on boot up (and start it now) using the following command:

```
chkconfig --levels 235 httpd on  
service httpd start
```

7. Test the WebDAV installation.

- a. Upload a `hello.world` file to the WebDAV server using the following command:

```
curl -T hello.world -u user:password http://myserver/recordings/hello.world
```

- b. Using a browser, open the `http://myserver/recordings/hello.world` URL. The browser will request for user credentials.

8. The WebDAV server is installed.

Deploy the Cassandra Database

Web Services stores the information about call recordings in a Cassandra database. For each contact center, the distinct column families with unique names exist for storing call recording information. These column families are created when the contact center is created, and deleted when contact center is deleted.

To deploy the Cassandra database for GIR, see the [Installing and Configuring Cassandra](#) section of the *Web Services and Applications Guide*.

Important

Web Services deletes column families only if they do not contain any call recordings; otherwise they should be deleted manually from Cassandra using the `cassandra-cli` tool.

Deploy Web Services and Applications

To install and configure Web Services and Applications, see the [Web Services and Applications Guide](#).

For Voice Recordings

Web Services requires a specific configuration in addition to the configuration that is described in the *Web Services and Applications Deployment Guide* for GIR **call** recordings to work correctly. The

following sections describe how to configure Web Services for call recordings.

Configuring the Web Services Parameters

To configure Web Services for Genesys Interaction Recording, add parameters to the **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the **server-settings.yaml** instead).

[+] Show the Parameters

Parameter Name	Mandatory	Description	Type	Default Value
enableBackgroundScheduledMediaOperations	N	Specifies whether to allow Web Services to schedule purge and backup events.	Boolean	True
createCallRecordingCn	N	Specifies whether to create a call recording column family (CRCF) for a new contact center.	Boolean	False
crClusterName	Y	Specifies the name of the elasticsearch cluster name.	Non-empty String	None Note: This is a mandatory parameter, and the value must be the same for all Web Services nodes in the cluster. For example, if there are five nodes in the Web Services cluster, all five nodes must have the same crClusterName value.
crRegion	N	Specifies the name of the region where the Web Services node resides.	Non-empty String	None
cryptoSecurityKey	Y	Specifies the security key used for Web Services encryption of the recording settings in the database.	Non-empty String	None Note: This is a mandatory parameter, and the value must be the same for all Web Services nodes in the cluster. For example, if there are five nodes in the Web Services cluster, all five nodes must have the same cryptoSecurityKey value.
defaultBackupExportURL	N	Specifies the location to store	Non-empty String	None

Parameter Name	Mandatory	Description	Type	Default Value
		backed up recordings. For example, file:///tmp/archLocDefault '.		
multiPartResolverMaxUploadSize		Specifies the maximum size, in KB, of the recording file.	Integer	536870912
multiPartResolverMaxInMemorySize		Specifies the maximum length of time allowed to upload a recording file.	Integer	536870912
nodePath	Y	Specifies the location and ID of the Workspace Web Edition & Web Services node within the deployment topology. This value must be unique across the deployment. For example, a value of /US/node1 means that the node is located in the US region and has an ID of "node1". The node ID can be the hostname, the IP address, or any other unique identifier.	Non-empty String	None
recordCryptoServerDecryptMaxConnection	N	Specifies the maximum TCP connections to each Recording Crypto Server instance defined in local-decrypt-uri-prefix settings. Note: This option applies to the Web Services version 8.5.200.85 and later only.	Integer	50
recordCryptoServerDecryptMaxTotalConnection	N	Specifies the maximum TCP	Integer	10 * recordCryptoServerDecryptMax

Parameter Name	Mandatory	Description	Type	Default Value
		connections to all Recording Crypto Server instances defined in local-decrypt-uri-prefix settings. Note: This option applies to the Web Services version 8.5.200.85 and later only.		
recordCryptoServerDecryptSocketTimeout		Specifies the socket timeout, in milliseconds, for TCP connections to Recording Crypto Server instances defined in ' <i>local-decrypt-uri-prefix</i> ' settings. Note: This option applies to the Web Services version 8.5.200.85 and later only.	Integer	30000
webDAVMaxConnections		Specifies the maximum TCP connections for each WebDAV Storage.	Integer	50
webDAVMaxTotalConnections		Specifies the maximum TCP connections the Web Services node allows to all WebDAV Storages.	Integer	10 * webDAVMaxConnections
undocumentedExternalApiUrl		Specifies the reachable Web Services Server address for the SpeechMiner UI, and the Screen Recording Client. Note: This option applies to the Web Services version 8.5.200.40 and later only.	String	http://<IP Address>:8090/internal-api

Configuring the Elasticsearch Engine

The Web Services Call Recording API uses the elastic search as the query engine. A configuration file

is required if call recording is enabled (for example, **JETTY_HOME/resources/elasticsearch.yml**).

[+] Show the Steps to Configure Elasticsearch

Configure the **JETTY_HOME/resources/elasticsearch.yml** file as follows:

```
index.analysis.analyzer.whitespace_lowercase.tokenizer: whitespace
index.analysis.analyzer.whitespace_lowercase.filter: lowercase

transport.tcp.port: 9200
http.port: 9300

discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: <comma separated list of HTCC nodes which host the ES>
discovery.zen.minimum_master_nodes: 2

gateway.recover_after_nodes: 2
gateway.recover_after_time: 1m
gateway.expected_nodes: 3

threadpool.index.queue_size: -1
threadpool.bulk.queue_size: -1

path.conf: <Path to genconfig folder>/elasticsearch
path.data: <Path to the folder where ES stores its data>
```

For more configuration information, see <http://www.elasticsearch.org/guide/>.

The Elasticsearch engine also requires a large PermGen space.

To increase the PermGen space:

- Add the following to your **JAVA_OPTIONS**:

```
JAVA_OPTIONS="-XX:MaxPermSize=512m -Djsse.enableSNIExtension=false"
```

- If you are using **/etc/default/jetty**, add:

```
JAVA_OPTIONS="-Xmx2048m -XX:MaxPermSize=512m -Xms2048m -Djsse.enableSNIExtension=false"
```

Important

The Elasticsearch index is saved in the **Jetty-Home/data** directory—for example, **/opt/jetty/data**.

Rebuilding the Elasticsearch Index

If you must upgrade your Jetty 8 version to Jetty 9 version, you might need to add the elasticsearch data file to the new Web Services cluster.

To move the elasticsearch data:

- Rebuild the elasticsearch index using the following command:

```
curl -XPOST "http://<FE VM host>/api/v2/ops/contact-centers/<ID contact center>/recordings"
-d '{ "operationName":"forceIndex", "from":<Time of previous 'green' state or backup
snapshot>}'
```

The command above executes the forceIndex operation and is used to rebuild the elasticsearch index when needed. The following information provides additional details for this API.

HTTP Request

POST
.../api/v2/ops/contact-centers/{id}/recordings

Request Body

```
{
  "operationName":"forceIndex",
  "from":1369272257713,
  "to":1369275857713,
  "purgeOld":true
}
```

The following table describes the request body attributes:

Attributes	Type	Mandatory	Description
operationName	String	Y	The name of the operation. In this case it is forceIndex.
from	Long Integer	Y	The Java time stamp (in milliseconds) which equals the UNIX time * 1000. This is the time stamp from which the records are re-indexed.
to	Long Integer	N	The Java time stamp (in milliseconds) which equals the UNIX time * 1000. This is the time stamp to which the records are re-indexed. If not specified, the

Attributes	Type	Mandatory	Description
			current time of the request processing is used.
purgeOld	Boolean	N	Specifies whether the old index should be deleted prior to re-indexing. This attribute is necessary if the Web Services updated version uses indexes with a different structure. The default value is false.

Restarting Web Services

For more information on starting and stopping Web Services, see the [Web Services Deployment Guide](#).

Configuring the Storage Credentials for Web Services

To enable voice recording:

1. Determine the contact center ID on Web Services using the following command with the ops username and password (ops:ops):

```
curl -u ops:ops http://<Web Services Server>:8080/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://<Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>"]}
```

Important

Use the <contact center ID (in hex format)> in all subsequent commands.

2. In a text editor, create the create_table file using the following command:

```
{
  "operationName":"createCRCF"
}
curl -u ops:ops -X POST -d @create_table http://htcc:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>/recordings
--header "Content-Type: application/json"; echo
```

To enable storage:

1. Using a text editor, create a new file called recording_settings with the following content:

```
{
```

```
"store": [
  {
    "webDAV": {
      "userName": "user1",
      "password": "password1",
      "uri": "http://apache1/recordings"
    }
  },
  {
    "webDAV": {
      "userName": "user2",
      "password": "password2",
      "uri": "http://apache2/recordings"
    }
  }
]
```

Important

The URI in `recording_settings` is case sensitive and must match the URI in the IVR Profile. For example:

```
"uri": "http://GENESYSREC1/recordings"
```

is not the same as:

```
"uri": "http://genesysrec1/recordings"
```

2. Execute the following command:

```
{
  curl -u ops:ops -X PUT -d @recording_settings
    http://<Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex
format)/settings/recordings
    --header "Content-Type: application/json"; echo
}
```

Configuring the Call Recording Audit Log

Web Services provides an audit log for the following call recording operations:

- Playback of the recording media file
- Deletion of the call recording file

To configure the audit log:

1. Stop the Web Service Jetty using the following command:
`sudo service jetty stop`
2. Update the Jetty LogBack Configuration:
 - Edit the `/opt/jetty/resources/logback.xml` file to include INFO level messaging **[+] Show example**
:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Example LOGBACK Configuration File
  http://logback.qos.ch/manual/configuration.html
-->
<configuration scan="true">
  <appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <filter class="ch.qos.logback.classic.filter.LevelFilter">
      <level>INFO</level>
      <onMatch>ACCEPT</onMatch>
      <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY for
printing only INFO-->
    </filter>
    <file>${jetty.logs}/recording.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <maxHistory>720</maxHistory><!-- 1 Month -->
    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}]
[%X{req.userAgent}] [%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
    </encoder>
  </appender>
  <appender name="FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>${jetty.logs}/cloud.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
      <!-- hourly rollover -->
      <fileNamePattern>${jetty.logs}/cloud-%d{yyyy-MM-dd-HH}.gz</fileNamePattern>
      <!-- keep 5 days' worth of history -->
      <maxHistory>120</maxHistory>
    </rollingPolicy>
    <encoder>
      <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} %-5level [%X{principal.name}]
[%X{session}] [%X{contactCenter}]
[%thread] %X{req.requestURI} %X{req.queryString} %logger{36} %msg%n</pattern>
    </encoder>
  </appender>
  <logger name="com.<domain>.cloud.v2.api.controllers.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>.cloud.v2.api.tasks.callrecording">
    <appender-ref ref="RECORDING" />
  </logger>
  <logger name="com.<domain>" level="WARN" />
  <logger name="com.<domain>.cloud" level="DEBUG" />
  <logger name="com.<domain>.cloud.rtreporting" level="WARN" />
  <logger name="com.<domain>.salesforce.security" level="INFO" />

  <root level="WARN">
    <appender-ref ref="FILE" />
  </root>
</configuration>
```

- For MLM:

- Create a **RECORDING** appender if it does not exist. **[+]** Show example

```
<appender name="RECORDING" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <filter class="ch.qos.logback.classic.filter.LevelFilter">
    <level>INFO</level>
    <onMatch>ACCEPT</onMatch>
```

```

        <onMismatch>DENY</onMismatch><!-- ACCEPT for printing log above INFO, DENY
for printing only INFO-->
    </filter>
    <file>${jetty.logs}/recording.log</file>
    <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
        <fileNamePattern>${jetty.logs}/recording-%d{yyyy-MM-dd}.gz</fileNamePattern>
        <maxHistory>720</maxHistory><!-- 1 Month -->
    </rollingPolicy>
    <encoder>
        <pattern>%d{MM/dd/yyyy HH:mm:ss.SSS, UTC} [%X{principal.name}]
[%X{req.userAgent}] [%X{req.remoteHost}] %X{req.requestURI} %msg%n</pattern>
    </encoder>
</appender>

```

- Add the following loggers:

```

<logger name="com.genesyslab.cloud.v2.api.controllers.callrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.controllers.screenrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.callrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.interactionrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.screenrecording">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.api.tasks.settings">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.media.scheduler">
    <appender-ref ref="RECORDING" />
</logger>
<logger name="com.genesyslab.cloud.v2.media.task">
    <appender-ref ref="RECORDING" />
</logger>

```

For more information about Jetty Logback, see [Logback configuration](#).

3. Start Jetty using the following command:

```
sudo service jetty start
```

4. Review the audit log.**[+]** **Show example**

- Open the **/var/log/jetty/recording.log** file. The following example shows that two recordings are requested for playback and deletion:

```

10/28/2013 15:46:03.203 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/
537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

```

```

10/28/2013 15:46:03.341 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-

```

```
d615-4329-957a-f5341ed
fd5d7/recordings/recid0/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a659.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a659] of recording [recid0] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] failed

10/28/2013 15:46:10.946 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] requested

10/28/2013 15:46:11.033 [ops] [Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/ops/contact-centers/46284f2f-
d615-4329-957a-f5341ed
fd5d7/recordings/recid1/play/2cb4ea04-f81d-44e8-83b6-1f4a63a1a658.mp3 Play media
[2cb4ea04-f81d-44e8-83b6-1
f4a63a1a658] of recording [recid1] from contact center [46284f2f-d615-4329-957a-
f5341edfd5d7] succeed

10/28/2013 15:46:52.179 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid0 Delete recording [reci
d0] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] requested

10/28/2013 15:46:52.216 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid0 Delete recording
[recid0] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] failed

10/28/2013 15:46:56.253 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete recording [reci
d1] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] requested

10/28/2013 15:46:56.420 [admin@genesyslab.com] [Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_0) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/28.0.1500.71 Safari/537.36] [192.168.135.1] /api/v2/recordings/
recid1 Delete recording
[recid1] from contact center [46284f2f-d615-4329-957a-f5341edfd5d7] succeeded
```

Setting the Advanced Options for Web Services

API Thread Pool

Web Services provides properties for the Call Recording API thread pool via archaius.**[+]** **Show the properties**

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name].	N/A	The hystrix timeout. The default

Property/API Name	Thread Pool Name	Description
execution.isolation.thread.timeoutInMilliseconds		value is set to 6000.
hystrix.threadpool.[API Pool Name] .coreSize	N/A	The thread pool size. The default value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateCallRecordingApiTaskV2	ApiCreatePool	Create call recording
DeleteCallRecordingApiTaskV2	ApiDeletePool	Delete call recording
GetCallRecordingApiTaskV2	ApiGetPool	Get call recording meta data
GetCallRecordingCFInfoApiTaskV2	ApiGetPool	Get call recording CF Information
GetCallRecordingMediaApiTaskV2	ApiGetPool	Streaming call recording media
QueryCallRecordingApiTaskV2	ApiQueryPool	Query call recording Meta data

For more information about the Web Services Call Recording API, see the [Web Services API Reference](#).

For more information about how to use Workspace Web Edition for Voice Recording, see the [Workspace Web Edition Help](#).

For Screen Recordings

As with call recordings, Web Services requires a specific configuration for GIR **screen** recordings to work correctly. The following sections describe how to configure Web Services for screen recordings.

Configuring the Parameters

1. On all Web Services instances, modify the **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the **server-settings.yaml** instead), and add the following parameters:

```
crossOriginSettings:
  corsFilterCacheTimeToLive: 120
  allowedOrigins: <Web Services Servers>,<SpeechMiner Web Servers>
  allowedMethods: GET,POST,PUT,DELETE,OPTIONS
  allowedHeaders: "X-Requested-With,Content-Type,Accept,Origin,Cookie,authorization,ssid,surl,ContactCenterId,Range"
  allowCredentials: true
screenRecordingSettings:
  screenRecordingEServicesEnabled: true
  screenRecordingVoiceEnabled: true
screenRecordingConnectionReportingSettings:
  reportingEnabled: true
  createReportingCF: true
multiPartResolverMaxUploadSize: 536870912
multiPartResolverMaxInMemorySize: 536870912
```

Important

- Change <Web Services Servers> and <SpeechMiner Web Servers> to the HTTP/HTTPS addresses of the Web Services instances and SpeechMiner Web Servers.
- **multipartResolverMaxUploadSize** controls the maximum allowed size for the Screen Recording video file that can be uploaded to Web Services and Applications (in bytes). Setting the value too high (10MB+) for this parameter may cause performance and/or security issues for Web Services and Applications.

2. Add screen recording features to the Contact Center:

```
POST http://<htcc-host-prefix>/api/v2/ops/contact-centers/  
bea09df2-82c5-441a-9072-5f2fc15fadc4/features  
{  
  "uris":[  
    "/api/api-voice-screenrecording",  
    "/api/api-multimedia-screenrecording",  
    "/api/api-screenrecording-connection-reporting"  
  ]  
}
```

Important

- Use the **api-voice-screenrecording** parameter for voice interactions, and use the **api-multimedia-screenrecording** parameter for non-voice interactions.
- If you wish to direct the SpeechMiner UI to Web Services instead of Recording Crypto Server for decryption of screen recordings, add the **api-recordings-decryption-proxying** parameter to the list of features enabled for the contact center above. Note that this requires additional configuration and applies to the Web Services version 8.5.200.85 and later only.

Restarting Web Services

For more information on starting and stopping Web Services, see the [Web Services Deployment Guide](#).

Configuring the Storage Credentials for Web Services

1. Determine the contact center ID on Web Services using the following command with the ops username and password (ops:ops):

```
curl -u ops:ops http://<Web Services Server>:<Web Services port>/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://<Web Services Server>:<Web Services port>/api/v2/ops/  
contact-centers/<contact center ID (in hex format)>"]}
```

Important

Use the <contact center ID (in hex format)> in all subsequent commands.

2. In a text editor, create a new file called `create_table`, with the following content:

```
{
  "operationName": "createCRCF"
}
```

And then execute the following command:

```
curl -u ops:ops -X POST -d @create_table http:// <Web Services Server>:<Web Services Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/screen-recordings --header "Content-Type: application/json"; echo
```

3. Enable storage for a single or multiple locations:

- For a **single** location:

- a. In a text editor, create the `create_single_location` file. **[+] Show how**

```
{
  "name": "storage",
  "location": "/",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "<webdav user>",
        "password": "<webdav password>",
        "storagePath": "<webdav uri>"
      }
    }
  ]
}
```

Important

Replace <webdav user>, <webdav password>, <webdav uri> with the appropriate values.

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_single_location http:// <Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>/settings/screen-recording --header "Content-Type: application/json"; echo
```

- For **multiple** locations:

- a. In a text editor, create the `create_first_location` file. **[+] Show how**


```
{
  "name": "storage",
  "location": "<node_location>",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "<webdav user>",
        "password": "<webdav password>",
        "storagePath": "<webdav uri>"
      }
    }
  ]
}
```

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_first_location http://<Web Services
Server>:8080/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
```

Important

Replace <node_location>, <webdav user>, <webdav password>, <webdav uri> with the appropriate values. The values for the <node_location> are similar to the nodePath settings in the Web Services **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the server-settings.yaml instead), but allow a hierarchical representation. For example, a Web Services node uses a storage setting with a location of "/US" in the nodePath set to "/US/AK" or "/US/HI".

For more information on hierarchical location setting, see https://docs.genesys.com/Documentation/CR/8.5.2/Solution/GWSSettings#Hierarchical_Location_Matching.

- c. Repeat steps a and b for each location required.

For more information on the properties of this settings group, see [Interaction Recording Web Services Settings Groups](#).

Setting the Advanced Options for Web Services

API Thread Pool

Web Services provides properties for the Screen Recording API thread pool via archaius.**[+]** **Show the properties**

The following table describes the parameters required to set the API thread pool.

Property/API Name	Thread Pool Name	Description
hystrix.command.[API Name]. execution.isolation.thread. timeoutInMilliseconds	N/A	The hystrix timeout. The default value is set to 6000.

Property/API Name	Thread Pool Name	Description
hystrix.threadpool.[API Pool Name] .coreSize	N/A	The thread pool size. The default value is set to 10.
RecordingOperationApiTaskV2	ApiOperationPool	The call or screen recording operation.
CreateScreenRecordingApiTaskV2	ApiUploadPool	Create screen recording
DeleteScreenRecordingMediaApiTaskV2	ApiDeletePool	Delete screen recording
GetScreenRecordingApiTaskV2	ApiGetPool	Get screen recording meta data
GetScreenRecordingMediaApiTaskV2	ApiStreamPool	Stream screen recording media
QueryScreenRecordingApiTaskV2	ApiQueryPool	Query screen recording meta data

For more information about the Web Services Call Recording API, see the [Genesys Interaction Recording API Reference](#).

Deploying SIP Server for GIR

Genesys Interaction Recording (GIR) needs SIP Server for routing, call control and to initiate the recordings. The following steps describe how to deploy and configure SIP Server for GIR, and how to configure the DNSs for GIR.

You can also use these configuration settings with SIP Cluster, but certain limitations might apply. Any limitations for SIP Cluster are noted in each section, where applicable.

For more information about the SIP Server configuration settings described on this page, see the [SIP Server Deployment Guide](#).

SIP Server

1. Install and configure SIP Server as described in the [SIP Server Deployment Guide](#).
2. In addition to the configuration described in the deployment guide, set the following SIP Server options:

Section Name	Parameter Name	Description
TServer	msml-support	Set to <code>true</code> to enable support of the call recording solution.
	resource-management-by-rm	Set to <code>true</code> to enable support of the call recording solution. Resource monitoring and notification will be done by the Resource Manager. SIP Server will contact Media Server through Resource Manager.
	msml-record-support	Set to <code>true</code> to enable SIP Server to engage GVP as a Media Server through the msml protocol for call recording.
	msml-record-metadata-support	Set to <code>true</code> to send additional metadata in the INFO message of Genesys Media Server when starting call recording.
	record-consult-calls	Specifies whether to record consult calls: <ul style="list-style-type: none"> • <code>true</code>—record consult calls. • <code>false</code>—do not record consult calls.
	recording-filename	Must be set to <code>\$UUID\$_\$DATE\$_\$TIME\$</code>
	wrap-up-time	(Optional) Duration of time (in seconds) to record the agent's

Section Name	Parameter Name	Description
		screen while they are in the After Call Work (ACW) state. For more information, see Agent Login .

VoIP Service DN

1. Create a new MSML DN object and add the following parameters to the **General** tab:
 - **Number** = The name of the MSML Server
 - **Type**= Voice over IP Service
2. Add the following parameters to the **Annex** tab of the new DN:

Section Name	Parameter Name	Description
TServer	contact	Set this to the Resource Manager IP address and port. Use the following format: sip: <Resource Manager_IP_address:Resource Manager_SIP_port> Specifies the contact URI that SIP Server uses for communication with the treatment server.
	service-type	Set to msml
	prefix	Set to msml=
	subscription-id	Set to the name of the tenant to which this SIP Server belongs, using the following syntax <TenantName>
	refer-enabled	Set to false
	make-call-rfc3725-flow	Set to 1
	ring-tone-on-make-call	Set to false
	sip-hold-rfc3264	Set to true
	oos-check	Set to 5
	oos-force	Set to 4

Agent DN

On the Agent's DN, in the **[TServer]** section, set the following parameters:

- If you want to start recording based on static DN-level settings, set the **record** parameter to true.

Important

This parameter can be set in either the **Agent DN** or **Agent Login** object, but not both. If setting it in **Agent DN**, make sure that the **record** parameter is not set to true in **Agent Login**.

- If you are using WDE or WWE, set the **enable-agentlogin-presence** parameter to false as the required information is provided by WDE or WWE.
- If you are not using WDE or WWE, set **enable-agentlogin-presence** to true. This option is required to provide agent hierarchy and name to SpeechMiner to ensure correct access limitations.

Agent Login

On the **Annex** of the Agent Login object, in the **[TServer]** section, set the following parameters:

- To start recording based on static DN-level settings, set the **record** parameter to true.

Important

This parameter can be set in either the **Agent Login** or **Agent DN** object, but not both. If setting it in **Agent Login**, make sure that the **record** parameter is not set to true in **Agent DN**.

- If you want to record the agent's screen while they are in the After Call Work (ACW) state, set the **wrap-up-time** in seconds; for example, set **wrap-up-time=10**. For more information, see the **isACWEnabled** parameter on the [Deploying the Screen Recording Service - Advanced Configuration](#) page.

Important

Agent Login objects are not supported if you are using SIP Cluster.

Deploying Interaction Concentrator for GIR

Important

The ICON deployment procedure is not required when using the Voice Processor instead of the Recording Processor Script (RPS).

Genesys Interaction Recording needs Interaction Concentrator (ICON) to store detailed reporting data from various sources in a contact center empowered with Genesys software.

Installing ICON

Install and configure ICON as described in the [ICON Deployment Guide](#). You can read more about ICON [here](#).

Important

Genesys Interaction Recording requires that the ICON database be case insensitive. Genesys also recommends that you use a separate ICON database for GIR (for example, do not use the same ICON database for GIR that is being used by Genesys Info Mart reporting).

If you want to deploy a single instance of the ICON database across multiple sites, see the [Supported Deployment Scenarios](#) in the ICON Deployment Guide.

Configuring ICON

In addition to the configuration described in the deployment guide, configure your ICON application as follows:

1. To collect all metadata, in the **[callconcentrator]** section, set the following parameters:
 - **adata-reasons-history** = none
 - **adata-extensions-history** = none
 - **adata-userdata-history** = all
 - **role** = all
2. To collect attached data, in the **[custom-states]** section, set the following parameters:

- **EventData** = <type1>,<key1>,<type2>,<key2>... where <typeN> is the data type (for example, char or int) and <keyN> is the attached data key name.
- **store-event-data** = conf

To improve ICON performance for Genesys Interaction Recording, Genesys recommends updating the ICON database schema with the following new indexes:

- Index G_PARTY:
 - NONCLUSTERED/NONUNIQUE INDEX G_PARTY.CALLID
- Index G_USERDATA_HISTORY:
 - NONCLUSTERED/NONUNIQUE INDEX G_USERDATA_HISTORY.CALLID
- Index G_IS_LINK:
 - NONCLUSTERED/NONUNIQUE INDEX G_IS_LINK.CALLID
- Index G_CUSTOM_DATA_S:
 - NONCLUSTERED/NONUNIQUE INDEX G_CUSTOM_DATA_S.CALLID

For optimal performance, it is recommended that the ICON's gsysPurge81 stored procedure (or similar) be used regularly to purge call data from the ICON database that is older than two days. See the [ICON User's Guide](#) for more information.

Important

Genesys Interaction Recording requires data from the following ICON tables:

- G_IS_LINK
- G_CALL
- G_PARTY
- G_PARTY_HISTORY
- G_AGENT_STATE_HISTORY
- G_CUSTOM_DATA_S
- G_USERDATA_HISTORY
- G_SECURE_USERDATA_HISTORY
- GC_AGENT

Make sure that you are populating these tables. For more information, see the [ICON Deployment Guide](#).

Deploying Recording Crypto Server

Genesys Interaction Recording (GIR) needs the Recording Crypto Server (RCS) to manage the certificates and the encryption/decryption process when retrieving and playing back the stored recording files.

Important

- RCS does not support on-the-fly configuration changes. Restart RCS to apply changes to the Genesys Advanced Disconnect Detection Protocol (ADDP) configuration.
- RCS will not start if Configuration Server is using UCS-2 encoding. In this scenario, use UTF-8 or set the Configuration Server option **[confserv] allow-mixed-encoding** to true.

Installing Recording Crypto Server

Preparing the Host

You must install the correct JRE version on the host machine where the Recording Crypto Server will be installed. For Recording Crypto Server 8.5.095.22 (or higher), JRE 17 is required. For Recording Crypto Server 8.5.095.17 (or lower), JRE 8 is required.

Important

For more detailed information about the supported versions for each operating system, see the [Genesys Supported Operating Environment Reference Guide](#).

To install JRE:

1. Perform one of the following:

- For Recording Crypto Server 8.5.095.22 (or higher), download and install Java Runtime Environment (JRE) 17 from your preferred provider. For example, you can download this from use an OpenJDK version of the software.
 - For Recording Crypto Server 8.5.095.17 (or lower), download and install Java Runtime Environment (JRE) 8 from your preferred provider. For example, you can download this from Oracle or use an OpenJDK version of the software.
- Set the following environment variables for your host, as follows:

- (Linux) Insert the following lines into the **/etc/profile** file:
export JAVA_HOME=/usr/lib/java/jre-<version of Java downloaded>/jre
Log out and log in again to activate the new environment variables in the current session.
- (Windows) Create a new System Variable named JAVA_HOME and use the path that was used during installation as the value. To do this, right-click your Computer icon. Select **Properties > Advanced System Settings > Environment Variables**, and then create the **JAVA_HOME** variable.

Installing Recording Crypto Server Using the Deployment Wizard

For instructions about installing Recording Crypto Server using the Genesys Administrator Extension, see the [Solution Deployment](#) section of the Genesys Administrator Extension User Guide.

When Recording Crypto Server (RCS) is started for the first time, and then terminated (either by using the Solution Control Interface or by killing the process) soon after, the RCS directory structure might be left in a partially initialized state. This can cause RCS to fail on subsequent attempts to start. To work around this, do not terminate RCS for at least 60 seconds starting it for the first time. If the directory structure is still invalid, delete all sub-directories in the RCS root directory, except for the conf and legal directories. When RCS is re-started, the required directories will be created.

Installing the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files

Important

If you are using Java 17, this step is not required.

In older versions of Java 8, the default installation limits key sizes to 128 bits. Larger key sizes can be enabled by installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

Important

- If you are using an OpenJDK version of Java, no additional cryptography configuration is required.
- If the Java 8 version you are using is older than Java 8u151, then follow the installation steps for JCE described below.
- If you are using Java 8u151 or 8u152, you do not need to download and install JCE. However, you must make a change to the unlimited policy option in the **JRE_HOME/lib/security/java.security** file. Find the **#crypto.policy=unlimited** line and remove the hash (#) character to uncomment it.
- If you are using Java 8u161 or newer, no additional cryptography configuration is required.

To install:

1. If you are using the Oracle version of Java 8, download the **Java 8** specific package from the **Oracle** website and follow the instructions provided with the package.
2. Copy the **Local_policy.jar** and **Us_export_policy.jar** files to the **JRE_HOME/lib/security** directory. If there are already copies of these files in that directory, make backup copies of these existing files in case you want to revert the installation.

Important

Make sure that the policy files are installed before starting the RCS for the first time. RCS will not start without these files.

Upgrading Recording Crypto Server

1. Make a backup copy of the **rcs.properties** file.
2. Make a backup copy of the **keystore** file.
3. Uninstall the Recording Crypto Server component.
4. Install the new Recording Crypto Server component.
5. Copy the settings from the backup copy of the **rcs.properties** file to the new **rcs.properties** file.
6. Copy the backup **keystore** file to the desired **keystore** file location and update the **rcs.properties** configuration file's **keystorepath** parameter to point to this file.

Configuring Recording Crypto Server

This section describes how to configure the Recording Crypto Server in your environment using Genesys Administrator Extension.

For more information about using Genesys Administrator Extension, see the [Genesys Administrator Extension Help](#).

Configuring the KeyStore and Certificate Authority

For information on how Genesys supports TLS for secure data exchange, refer to [Securing Connections Using TLS](#) in the [Genesys Security Deployment Guide](#).

The Recording Crypto Server stores certificate and key data files based keystores. Certificates uploaded to the server can be optionally validated against a Certificate Authority (CA).

Important

The CA configuration is used for recording certificates and not for TLS network

connections. This section describes the keystore and CA related configuration parameters.

To limit access, all recording encryption key related parameters are stored in a local **<Recording Crypto Server Install Directory>/conf/rcs.properties** configuration file.

The following table lists the parameters used in the **rcs.properties** configuration file.

Parameter Name	Default Value	Description
keystorepath	keystore.bin	Specifies the path to the keystore file. If HA is enabled, the keystore file should be accessed through a network share (see Configure HA).
keystorepassword	genesys	Specifies the password that accesses the keystore file. Note: The keystorepassword parameter can be overridden by the RCS_KEYSTORE_PASSWORD environment variable. In this case the same password is used for both keystorepassword and keypassword.
keypassword	genesys	Specifies the password used for each private key that is added to the keystore. Note: <ul style="list-style-type: none">The same password is used for each private key.The keypassword parameter can be overridden by the RCS_KEYSTORE_PASSWORD environment variable. In this case, the same password is used for both keystorepassword and keypassword.
cacertstorepath	Java-R00T	Specifies the CA certificate keystore. Possible values are: <ul style="list-style-type: none">Java-R00T—The path to the default Java JRE CA certificate file.Windows-R00T—The path to the Windows system keystore. This is not valid for Linux systems.

Parameter Name	Default Value	Description
		<ul style="list-style-type: none">File Path—The path to use the CA keystore. This file must be a Java JKS keystore file.None—Disables validation of certificates.
cacertstorepassword	changeit	Specifies the password for the CA certificate keystore.

The following shows an example **rcs.properties** configuration file:

```
keystorepath=keystore.bin
keystorepassword=keystorepassword
keypassword=keypassword
cacertstorepath=Java-R00T
cacertstorepassword=capassword
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

The Recording Crypto Server uses API calls to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for recording playback and archival operations. To configure the Interaction Recording Web Services (Web Services) connection, set the following parameters in the **[htcc]** section of the Recording Crypto Server application:

Parameter Name	Default Value	Description
baseurl	https://htcchost:8080	Specifies the base URL for the Interaction Recording Web Services (Web Services) connection. This parameter is dependent on the Interaction Recording Web Services (Web Services) server protocol (http or https), port, and URL suffix.
internalUrlPrefix	/api/v2	Controls the prefix added to requests sent to Interaction Recording Web Services to retrieve recording files. By default, or if a value other than disable is specified, RCS will concatenate the baseurl , internalUrlPrefix , and the mediaPath returned by RWS as the request URL. If the internalUrlPrefix value is set to disable , RCS will use the mediaUri from the metadata instead when fetching the recordings from RWS.
domain	Empty string	Specifies the domain of the Interaction Recording Web Services (Web Services) contact

Parameter Name	Default Value	Description
		center. This is the domain ID set for the contact center within Interaction Recording Web Services (Web Services).
user	ops	Specifies the name of the operations user for the Interaction Recording Web Services (Web Services) connection.
password	opspassword	Specifies the password of the operations user for the Interaction Recording Web Services (Web Services) connection.
max-sr-playback-connections	50	Specifies the maximum number of HTTP connections between Recording Crypto Server and Interaction Recording Web Services (Web Services) for screen recording playback.
contactcenterid	Empty string	Specifies the contact center ID value in the RCS requests sent to Interaction Recording Web Services (RWS). If this value is not specified, the contact center ID information is derived from the /api/v2/ops/contact-centers request sent to RWS. Important: If you are a Recording Crypto Server API user and you specify an empty Contact Center ID (CCID) when using the /rcs/contact-centers/<ccid>/recordings/... path, you will receive a misleading HTTP 403 Access is denied message.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. RCS will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web

Parameter Name	Default Value	Description
		Services (Web Services) in the Configuring Transport Layer Security (TLS) Connections (Optional) section.

Configuring Cross Origin Resource Sharing (CORS)

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has Configuring Cross-Site Request Forgery (CSRF) protection enabled, CORS must be configured.

To configure CORS, set the following options in the **[cors]** section of the Recording Crypto Server application:

Parameter Name	Default Value	Description
allowed-origins	empty	Specifies the allowed origins list that is attached in the HTTP response Access-Control-Allow-Origins header, sent to a cross-origin request. In this case, it must be a list of all base URLs used by the users to connect to SpeechMiner Web.
allowed-headers	X-Requested-With, Content-Type, Accept, Origin, Cookie, authorization, Access-Control-Allow-Headers, X-Response-Access-Control-CenterId, X-Response-Access-Control-CenterId	Specifies the allowed headers list that is attached in the HTTP response Access-Control-Allow-Headers header, sent to a cross-origin request.
allowed-methods	GET, POST, PUT, DELETE, OPTIONS	Specifies the allowed methods list that is attached in the HTTP response Access-Control-Allow-Methods header, sent to a cross-origin request.
allow-credentials	true	Specifies the value sent in Access-Control-Allow-Credentials header of the HTTP response to cross-origin request.
enable-cors-filter	true	Enables handling cross-origin requests originating from other domains like SpeechMiner Web. The value should always be true for CORS security to be active.

Configuring SameSite Cookie for Screen Recording Playback (Optional)

The Recording Crypto Server provides the ability to enable the **SameSite=None** and **Secure** cookie attributes for the cookie used for screen recording playback in the SpeechMiner browser application. These attributes are not set by default.

To configure the **SameSite** and **Secure** cookie attributes, set the following option within the **[general]** section of the Recording Crypto Server application:

Important

Before enabling this option, ensure that the connection between the SpeechMiner browser application and Recording Crypto Server is configured to use HTTPS. If you set the value of this option to `true` and are using HTTP, the cookie will not be returned by the browser.

Parameter Name	Default Value	Description
<code>samesite.enable</code>	<code>false</code>	Specifies whether the SameSite=None and Secure cookie attributes are set during screen recording playback from the SpeechMiner browser application.

Configure Passwords

Important

- In a Linux or Windows environment, RCS supports reading the RCS keystore password from an environment variable instead of from the configuration file. When both are available, the environment variable takes precedence.
- **RCS_KEYSTORE_PASSWORD** - maps to the existing configuration parameters `keystorepassword` and `keypassword` in the RCS properties file. When specified the same password is used for both parameters.

In a Windows environment only, the Recording Crypto Server (RCS) can store the password in the Windows Vault instead of in the `rcs.properties` file.

For example, run the following commands for the Recording Crypto Server located at `<Recording Crypto Server Directory>\scripts\powershell`:

Command to store: `encryptPassword.bat [-store <path to credentials store>] -password <password>`

Command to start RCS: `startRCS.bat [-store <path to credentials store>] -rcs <command to start RCS>`

For example:

```
startRCS.bat -store C:\GCTI\RecordingCryptoServer\rcs.secret -rcs java %JAVA_OPTS%  
-jar rcs.war -host host1.example.com -port 8888 -app RCS_Application
```

where:

- **host1.example.com** is the host for the Configuration Server.

- **8888** is the port for the Configuration Server.
- **RCS_Application** is the RCS application object.

Important

If the command `<path to credentials store>` contains a space, the path must be enclosed with quotation marks ("").

Configuring Archiving

The Recording Crypto Server provides support for automatic archiving of recordings that are older than a predefined time.

Important

Genesys recommends that the Media Lifecycle Management (MLM) functionality, which provides more flexible backup and purging rules, be used instead (see [Media Lifecycle Management](#)). New features, such as protecting recordings from deletion, are not supported with the Recording Crypto Server archiving mechanism.

To configure archiving, set the following options:

1. In the **[general]** section, set the **archive.block-size** option to the number of recordings RCS will fetch for archiving. The valid value ranges from 100 to 10000 and the default value is 5000. This option is used to verify that RCS does not run out of memory when it fetches all of the recordings at one time for archiving.

Important

Genesys recommends setting the RCS maximum Java heap size to no less than 1024 MB when **archive.block-size** is 5000. This setting enables you to avoid RCS running out of memory. Increase the maximum Java heap size accordingly when you increase the **archive.block-size**. To set the maximum Java heap size for RCS, add the **JVM** option (`-xmx1024m`), to the RCS start script.

2. On the Annex tab of each Tenant (including the Environment Tenant), in the **[recording.archive]** section, set the following parameters:

Parameter Name	Default Value	Description
interval	1	Specifies how often, in days, the archiving process runs.
retentiontime	60	Specifies how long, in days, to keep the recordings before archiving them.

Parameter Name	Default Value	Description
speechminerurl	https://host/interactionreceiver	Specifies the SpeechMiner URL where the recording metadata is stored.
user	archiveuser	Specifies the SpeechMiner username used to authenticate the SpeechMiner database.
password	changeit	Specifies the SpeechMiner password that is used to authenticate the SpeechMiner database.
outputfolder	archive	Specifies the destination folder where the archived recordings are stored.
speechminer-trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to SpeechMiner Interaction Receiver. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. RCS will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to SpeechMiner Interaction Receiver in the Configuring Transport Layer Security (TLS) Connections (Optional) section.

Important

Genesys does not recommend using a Network driver for Recording Crypto Server archive output. Therefore, set output to be a physical hard drive on the same machine.

Configuring High Availability

The Recording Crypto Server provides High Availability (HA) support to multiple Recording Crypto Server instances accessed through a load balancer. In this mode, all Recording Crypto Server instances use the same keystore file accessed through a network share, and are accessed through a single URL that utilizes the load balancer. To configure HA:

1. Set the Redundancy Type to Hot Standby on each Recording Crypto Server application instance. This setting enables logic for coordinated access to a shared keystore file.

2. Create a network share for the keystore file and set the **keystorepath** parameter in the Recording Crypto Server local configuration file to point to this file. Ensure that each Recording Crypto Server instance has read and write access to the keystore file.
3. Set the Recording Crypto Server URL parameter of the SpeechMiner application to the load balancer URL of Recording Crypto Server. If Genesys Administrator Extension is to be configured with a tenant specific URL for Recording Crypto Server, set this to the URL of the load balancer.
4. Create a Recording Crypto Server Cluster application using the `recording_crypto_850` application template, and set the following parameters:
 - On the General tab:
 - Application Name—The name of the cluster (for example, `RCS_Cluster`).
 - Working Directory—A period ".".
 - Command Line—A period ".".
 - Command Line Arguments—A period ".".
 - Host—The name of the host that the load balancer is installed on. This host must be in the configuration database.
 - On the Ports tab:
 - Configure the port by following the instructions provided in the [Configure HTTP / HTTPS Port](#) section.
5. Add a connection in the Genesys Administrator Extension application to the Recording Crypto Cluster application.

Important

For RCS HA configuration, each RCS instance operates in primary mode. The Backup Server setting on the Server Info tab of each RCS application should be set to None.

Example Load Balancer Configuration

The following is example configuration for the Apache load balancer. The details of setting up the required Apache modules are not shown. The load balancer setup must include "session sticky" so that a session that starts on a particular balancer member continues to be directed to the same member. This is achieved in the example below using the **route** and **stickysession** parameters. The **route** value must be set to the application name of the Recording Crypto Server instance, where " " characters in the name are replaced with the `_` character. For example, if the application name is **RCS 1**, set the **route** value to `RCS_1`.

```
<Proxy balancer://rcscluster>
BalancerMember https://rcshost1:port/rcs disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember https://rcshost2:port/rcs disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcscluster
```

If High Availability mode is not to be used, set the Recording Crypto Server's application Redundancy

Type to Not Specified. For this mode, the keystore file can be located on the local file system, a network share is optional.

Configuring an HTTP / HTTPS Port

To configure a port, follow these steps:

1. Log onto Genesys Administrator Extension (GAX).
2. In the GAX **Configuration** tab, choose **Environment**. Then, click **Applications** and select Recording Crypto Server application.
3. Go to the **Ports** tab in the Recording Crypto Server application.
4. Add a port or edit the existing one by entering values in the fields, **Port ID** and **Communication Port**. Note that there must be only one port.

You can configure either an unsecured port or a secured port based on your requirement.

Configuring an Unsecured (HTTP) Port

1. Enter the value `http` in the **Connection Protocol** field.
2. Enter the value `unsecured` in the **Listening Mode** field.
3. Leave the other fields empty and click **Save**.

Configuring a Secured (HTTPS) Port

1. Leave the **Connection Protocol** field empty.
2. Enter the value `secured` in the **Listening Mode** field. This sets the value `tls=1` in the **Transport Parameters** field automatically.
3. If you are setting up Mutual TLS, add `tls-mutual=1` to the **Transport Parameters** field.
4. Configure the secure port parameters at the appropriate level, as follows:

Important

If the protocol is set to `https` or left blank, a TLS server certificate and private key must be configured. This is done using the common method for Genesys applications as documented in the [Genesys Security Deployment Guide](#). The certificate and private key can be configured in the host object, the application object, and the application port entry for HTTPS.

- Host Level
 - a. In the GAX **Configuration** tab, choose **Environment** and click **Hosts**.
 - b. Click on the host object on which the server is running and enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
 - c. Restart the Recording Crypto Server.

- Application Level
 - a. In the GAX **Configuration** tab, choose **Environment** and click **Applications**.
 - b. In the **General** tab of the Recording Crypto Server application object, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
 - c. Restart the Recording Crypto Server.
- Port Level
 - a. In the GAX **Configuration** tab, choose **Environment** and click **Applications**.
 - b. In the **Ports** tab of the Recording Crypto Server application object, click on the port that you created earlier and enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
 - c. Restart the Recording Crypto Server.

Configuration of certificates at the port level has precedence over the application level, which has precedence over the host level. The private key PEM file must be in PKCS8 format. This can be achieved using the following openssl command:
openssl pkcs8 -topk8 -nocrypt -in private_keyfile.pem -inform PEM -out private_keyfile_pkcs8.pem

For more information on securing connections, refer to the [Genesys Security Deployment Guide](#).

Configuring the Connection to the Primary Configuration Server

To work with Configuration Server High Availability, the Recording Crypto Server (RCS) requires a connection to the primary Configuration Server application. For information on how to set this connection, see the [Framework 8.5.1 Management Layer User's Guide](#).

RCS supports an Advanced Disconnect Detection Protocol (ADDP) connection to the Configuration Server. To enable ADDP, perform the following:

- Add the Configuration Server to the RCS Connections tab.
- Specify the connection protocol as ADDP.
- Configure remote and local timeouts, valid values are 0-3600, where 0 means no timeout.
- Specify the required trace mode, either Local, Remote, or both.

For additional details, see the Advanced Disconnect Detection Protocol page in the [Framework 8.5.1 Deployment Guide](#).

Important

- You will see log messages about ADDP activity in the RCS logs despite switching ADDP Trace Mode to **Trace Is Turned Off** or **Trace On Server Side**. This is due to the underlying libraries handling ADDP protocol functionality.
- ADDP debug logging can be suppressed by modifying the **suppress-debug-loggers** value in the **[log]** section of the RCS configuration to contain:

```
com.genesyslab.platform.commons.connection.interceptor.AddpInterceptor  
, com.genesyslab.platform.commons.timer.impl.SchedulerImpl
```

- Genesys Advanced Disconnect Detection Protocol (ADDP) will appear in the **[log]** section of the Configuration Server log files when **verbose=all**.

Configuring Log Output

The Recording Crypto Server supports the Genesys Management Framework log configuration. For information on how to set up log output appropriate for your Recording Crypto Server application, see the Common Log Options section of the [Framework 8.5.1 Configuration Options Reference Manual](#).

Configuring the Connection to Message Server

The Recording Crypto Server must have a connection to the Message Server application to enable central auditing and alarming. For information on how to set this connection, see the [Framework 8.5.1 Management Layer User's Guide](#).

Configuring Transport Layer Security (TLS) Connections (Optional)

Configuring TLS connection to Interaction Recording Web Services (Web Services)

1. Set up TLS on Interaction Recording Web Services. For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[htcc]** section of the Recording Crypto Server configuration file, set the `baseurl` parameter to use https.
3. In the **[htcc]** section of the Recording Crypto Server configuration file, configure the **trusted-ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted-ca** to true.
 - If the TLS certificate being used by RWS is a self-signed certificate, set **trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted-ca** to false. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime

Environment 8 versions disallow MD5 signatures for certificates.

Important

The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to SpeechMiner Interaction Receiver

1. Set up TLS on SpeechMiner Interaction Receiver. For more information, see [SpeechMiner Server-Side Configuration](#).
2. On the **Annex** tab of each Tenant (including the Environment Tenant), in the **[recording.archive]** section, set the `speechminerurl` parameter to use `https`.
3. In the **[recording.archive]** section, configure the `speechminer-trusted-ca` parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **speechminer-trusted-ca** to `true`.
 - If the TLS certificate is a self-signed certificate, set **speechminer-trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and TLS is used only for encrypted transmission, set **speechminer-trusted-ca** to `false`. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 8 versions disallow MD5 signatures for certificates.

Important

The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to Message Server

1. Set up TLS on Message Server. For more information, see [Securing Core Framework Connections](#) section in the *Genesys Security Deployment Guide*. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. To connect to the secure TLS port, see [Configuring a Secure Client Connection to Other Genesys Servers](#)

section in the *Genesys Security Deployment Guide*.

3. In the properties of the **Connection** table, configure the **trusted-ca** parameter as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted-ca** to true.
- If the TLS certificate is a self-signed certificate, set **trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and TLS is used only for encrypted transmission, remove the **trusted-ca** parameter from the configuration. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 8 versions disallow MD5 signatures for certificates.

Important

The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to Configuration Server

1. Set up TLS on the Configuration Server. For more information, see [Configuring TLS on Configuration Server](#) in the *Genesys Security Deployment Guide*. Refer to [Genesys Security Deployment Guide](#) to acquire TLS certificates and private keys.
2. In the command line arguments of start information in the RCS application properties, change the port to use the Configuration Server Auto-Detect port.

For more information about the Recording Crypto Server options, see the [Genesys Interaction Recording Options Reference](#).

Deploying the Recording Plug-in for GAX

Installing Recording Plug-in for GAX

Prerequisites

- For Recording Plug-in for GAX version 8.5.500.05 (or higher):
 - Required software: Genesys Administrator Extension 9.0.107.x or later.
 - Required software: **Interaction Recording Web Services** 8.5.205.69 (or higher) instance where the Recording Lifecycle Scheduler settings and Screen Recording Certificates are updated.
 - Required software: **Recording Crypto Server** 8.5.095.22 (or higher) instance to store the Recording Certificates.
- For Windows, keep the full path of your Recording Plug-in installation directory ready.
- For Linux, keep the full path of your Genesys Administrator Extension installation directory ready.
Keep the full path of your Recording Plug-in installation directory ready.

Important

The Recording Plug-in template XML file includes the role privilege data that must be imported into Genesys Administrator Extension.

The full path of the Recording Plug-in installation directory and the full path of the GAX installation directory should be different.

Installing the Plug-in

To install the Plug-in:

1. Install the Recording Plug-in IP.
 - For Windows, the file is located at <IP plugin directory>/setup.exe.
 - For Linux, the file is located at <IP plugin directory>/install.sh.
2. Perform one of the following as required:
 - For Recording Plug-in for GAX version 8.5.500.05 (or higher), copy the gax-rcs-8.5.5xx.xx.jar file from the installed Recording Plug-in folder to GAX_ROOT/webapp/WEB-INF/lib directory.
For example, copy gax-rcs-8.5.5xx.xx.jar from C:\Program Files\GCTI\RecPluginGAX64 to C:\Program Files\GCTI\Genesys Administrator Extension\webapp\WEB-INF\lib.
 - For Recording Plug-in for GAX version 8.5.097.61 (or lower), copy the gax-rcs-8.5.0xx.xx.jar file from the installed Recording Plug-in folder to GAX_ROOT/plugin-ins directory.

For example, copy `gax-rcs-8.5.0xx.xx.jar` from `C:\Program Files\GCTI\RecPluginGAX64` to `C:\Program Files\GCTI\Genesys Administrator Extension\plug-ins`.

3. Restart Genesys Administrator Extension.
4. Import the metadata to Genesys Administrator Extension:
 - Log in to GAX.
 - From the top menu, choose **Configuration > Application Templates**.
 - Click **New**.
 - Click **Import Metadata**.
 - Click **Choose File** and choose the `recording_plugin_855.xml` file located in the **Templates** folder of Recording Plugin IP.
 - Click **OK**.
 - Provide a unique name for the template.
 - Enter the Recording Plug-in version that is defined in `recording_plugin_855.xml`.
 - Choose **Type** as **Genesys Administrator Server**.
 - Click **Save**.
5. Upload and deploy the new **SPD**.

Installing the Plug-in via GAX Installation Package UI (Discontinued after GIR 8.5.224.00)

Important

The Recording Plug-in template XML file includes the role privilege data that must be imported into Genesys Administrator Extension. This data is imported when the IP and template files are imported into Genesys Administrator Extension.

Installing the Plug-in from within Genesys Administrator Extension

Prerequisites

- Be prepared to enter the directory path to an installation directory, or to a zipped file.
- Required software: Genesys Administrator Extension 8.1.4 or later.

To install the Plug-in:

1. Select **Installation Packages** from the **Administration** menu.
2. Click the "plus" icon (+) at the upper right of the **Installation Packages** window. The **Software Installation Wizard** dialog appears to the right of the current window, offering these **Import Type**

Selection choices as radio buttons:

- Installation Package Upload (includes templates)
- Installation Package Upload (template uploaded separately)
- UNC Path to Mounted CD or Directory
- UNC Path to an Existing Administrator Repository
- UNC Path to Zipped IPs from Support

3. Select the radio button that matches your installation source and click the Next button.

4. The next dialog will request input according to your choice in the previous step:

- Installation Package Upload (includes templates) requires you to choose a zipped IP file.
- Installation Package Upload (template uploaded separately) requires you to select a zipped IP file, an XML template and an APD template (all three).
- Each of the three choices that begin with UNC Path requires a directory path that you may type or paste into the entry field. You may see a request to correct an error; type or paste your correction. When GAX is ready to install, the Finish button will be enabled.

5. Click the Finish button and wait for the upload to complete. When you see the message, Import has started. You may now close this wizard, close the Software Installation Wizard dialog by clicking the Close button at the bottom right or the X icon at the top right. The Plug-in is ready to install.

6. Select the item that you imported from the Installation Packages window. A dialog with that title appears to the right.

7. The Genesys Interaction Recording Plug-in for GAX dialog offers these actions:

- Download—Downloads the installation package to your computer.
- Delete—Erases the IP.
- Copy to Tenants—Copies the IP to the tenant(s) that you specify. Select a tenant and click Finish.
- Deploy Profile: install—Displays the IP Deployment Wizard start dialog. All following steps in this procedure are the result of this choice.

8. Click Next to display a list of host computers for possible installation. Select one or more hosts for installation using the check box to the left of each host name, then click Next.

9. At the Application Parameters dialog, complete these fields:

- Application name for host
- Tenant Name
- App port
- Primary Configuration Server
- Backup Configuration Server
- Skip IP Re-install

Notes:

- Click the Information (i) icon to the right of each field title, for tool tip help.
- A red * indicates a mandatory entry.

- Click Next when you have completed all mandatory fields.
10. Perform this step depending on the OS:
 - For Windows, at the Installation Parameters (`silent.ini`) dialog, complete the IPCommon: InstallPath field. This is the path where the IP binaries will be extracted to, and the directory must exist on the machine. The default answer offered is `C:\genesys\GCTI\`.
 - For Linux, at the Installation Parameters (`silent.ini`) dialog, complete the following steps:
 - a. The IPCommon: InstallPath field. This is the path where the IP binaries will be extracted to, and the directory must exist on the machine. The default answer offered is `/home/genesys/GCTI\`.
 - b. The RecPluginGAX: GAX_Directory field, which corresponds to the installation root folder for the GAX installation. The Recording Plugin for GAX jar file should be placed in `<GAX_ROOT>/webapp/WEB-INF/lib` directory.
 11. At the Deployment dialog, verify that the answers you gave are all correct. If they are correct, click Finish and wait for the installation to complete.
 12. Restart Genesys Administrator Extension.

Upgrading the Plug-in

Using GAX 8.1.4

If you are using Genesys Administrator Extension version 8.1.4, perform the following steps:

Prerequisites

- The previous version of the Plug-in must be uninstalled.
- Be prepared for these information requests and choices:
 - You will need the full path to your Genesys Administrator Extension installation.
 - You will either confirm the default installation directory, or enter a new one.
 - If the target installation directory is populated, you will choose an action:
 - Back up all files in the directory.
 - Overwrite only the files contained in this package.
 - Wipe the directory clean.
- Perform Step 4 of the **Installing the Plug-in** section. This is required to import the Plug-in metadata to Genesys Administrator Extension.

1. Stop Genesys Administrator Extension.

2. Run the installation executable:

- For Windows, this file is `<IP plugin directory>/setup.exe`.
- For Linux, this file is `<IP plugin directory>/install.sh`.

- Copy gax-rcs-8.5.xxx.xx.jar from the installed Recording Plug-in folder to GAX_ROOT/plugin-ins directory.
For example, copy gax-rcs-8.5.097.57.jar from C:\Program Files\GCTI\RecPluginGAX64 to C:\Program Files\GCTI\Genesys Administrator Extension\plugin-ins.
- Start Genesys Administrator Extension.
- Upload and deploy the new **SPD** file.

Using GAX 9.0.107.x (or higher)

If you are using Genesys Administrator Extension version 9.0.107.x or later, perform the following steps:

Prerequisites

- The previous version of the Plug-in must be uninstalled:
 - For Windows, use the **Control Panel** to uninstall the Plug-in. You must manually remove the Plug-in gax-rcs-8.5.5xx.xx.jar file from the GAX_ROOT\webapp\WEB-INF\lib directory.
 - For Linux, manually remove the Plug-in gax-rcs-8.5.5xx.xx.jar file from the GAX_ROOT/webapp/WEB-INF/lib/ directory.
- Be prepared for these information requests and choices:
 - You will need the full path to your Genesys Administrator Extension installation.
 - You will either confirm the default installation directory, or enter a new one.
 - If the target installation directory is populated, you will choose an action:
 - Back up all files in the directory.
 - Overwrite only the files contained in this package.
 - Wipe the directory clean.
- Perform Step 4 of the **Installing the Plug-in** section. This is required to import the Plug-in metadata to Genesys Administrator Extension.

1. Stop Genesys Administrator Extension.
2. Delete the previous version of the plug-in .jar file (for example, gax-rcs-8.5.5xx.xx.jar) from the GAX_ROOT/plugin-ins/ directory.
3. Copy the gax-rcs-8.5.5xx.xx.jar file from the installed Recording Plug-in folder to GAX_ROOT/webapp/WEB-INF/lib directory.
For example, copy gax-rcs-8.5.5xx.xx.jar from C:\Program Files\GCTI\RecPluginGAX64 to C:\Program Files\GCTI\Genesys Administrator Extension\webapp\WEB-INF\lib.
4. Start Genesys Administrator Extension.
5. Upload and deploy the new **SPD** file.

Configuring the Plug-in

For the Recording Plug-in connection to the Recording Crypto Server in the Genesys Administrator Extension application connections, add a connection to the Recording Crypto Server application. If the Recording Crypto Server is setup in HA mode where there are multiple Recording Crypto Server instances behind a load balancer, set this connection to the **Recording Crypto Server cluster application**.

Configure for Screen Recording

Single-Tenant Environment

1. Using Genesys Administrator Extension, navigate to **Configuration > Applications > <Genesys Administrator Extension Application Object> > Application Options**.
2. Under the **rccs** section, set the **htcc_base_url** parameter to the Interaction Recording Web Services server URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; `http://<Interaction Recording Web Services IP Address>:8081`

Multi-Tenant Environment

1. Using Genesys Administrator Extension, navigate to **Configuration > Tenant> <Tenant Object> > Options**.
2. Under the recording section, set the **htcc_base_url** parameter to the Tenant-specific Interaction Recording Web Services load balancer URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; `http://<Interaction Recording Web Services IP Address>:8081`

Important

If this parameter does not exist or has an empty value, the Recording Plug-in retrieves the Interaction Recording Web Services (Web Services) base URL from the **htcc** section of the configured Recording Crypto Server parameters.

Configuring Transport Layer Security

1. Review the documentation for TLS support for GAX at **Transport Layer Security (TLS)**.
2. Configure TLS for Interaction Recording Web Services (see **Configuring TLS on the Server-Side for Interaction Recording Web Services**) and Recording Crypto Server (see **Configure HTTP**).
3. Add a CA certificate to the trust store by executing the following command line:

```
keytool -importcert -file <root_ca certificate of rcs/rws> -keystore keystore
```
4. Configure the trust store location for GAX by updating the Java environment. For example, on Linux or Windows you can configure this by adding the following lines to the **setenv.sh** or **setenv.bat** script, respectively:

```
set JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore="<path to keystore file which is generated using above command>"
set JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword="<password>"
```

5. Enable Transport Layer Security (TLS) certificate validation by using the following parameters:

- **trusted_ca_rcs** - Configures whether or not the certificate will be validated when making a secure outbound connection to Recording Crypto Server (RCS). This parameter can be configured in the **[rcs]** section of the GAX application object or in the **[recording]** section of a specific tenant object. If this parameter is configured in both sections, then the tenant level configuration takes higher priority than the application level. Valid values are `true` or `false`. This parameter is optional, and defaults to `true`.
- **trusted_ca_rws** - Configures whether or not the certificate will be validated when making a secure outbound connection to Interaction Recording Web Services (RWS). This parameter can be configured in the **[rcs]** section of the GAX application object or in the **[recording]** section of a specific tenant object. If this parameter is configured in both sections, then the tenant level configuration takes higher priority than the application level. Valid values are `true` or `false`. This parameter is optional, and defaults to `true`.

When validation is disabled, all the certificates are not validated, thereby ignoring the following:

- Expired certificates
- Certificates with missing chain
- No trusted CA installed in case of self signed certificate

Configure Roles and Privileges

To configure the roles and privileges required for Genesys Interaction Recording, see [Configuring Access Control for GIR Users](#).

For more information about the Plug-in options, see the [Genesys Interaction Recording Options Reference](#).

Deploying Recording Processor Script

Recording Processor Script (Python 3)

Important

Voice Processor, a multi-threaded microservice based on the Node.JS platform, is an alternative to the Recording Processor Script (RPS).

- For information on deploying Voice Processor, see [Deploying Voice Processor](#).
- To migrate from an existing RPS deployment to Voice Processor, see [Migrating from RPS to Voice Processor](#).
- RPS is not supported for deployments integrated with SIP Cluster. If your deployment uses SIP Cluster, you must use Voice Processor.

For new deployments, Genesys recommends using Voice Processor instead of RPS.

Prerequisites

Before installing and configuring the RPS, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.
- For Recording Processor Script 8.5.500.13 (or higher), Recording Muxer Script must be upgraded to 8.5.500.10 (or higher).

Installing Recording Processor Script

Installing on Windows

1. Install 64 bit Python 3.11.5 from the [Python](#) website. To make Python 3 to work with OpenSSL 3.0.13, follow the below steps:
 - Download `libcrypto-3.dll` and `libssl-3.dll` from the [Python Binary repository](#).

- In [python-source-folder]\DLLs, replace with the above downloaded DLL files.
2. Install the **RPS IP** with the installer.

Important

Offline installation is supported starting from RPS 8.5.500.19.

To install it in a fully offline environment, follow these steps:

1. Unzip the <RPS>\thirdparty\flit_core-3.10.1.zip file.
2. Run `py -m pip install . --no-build-isolation` from the <RPS>\thirdparty\flit_core-3.10.1 directory.
3. Unzip the <RPS>\thirdparty\wheel-0.45.0.zip file.
4. Run `py -m pip install . --no-build-isolation` from the <RPS>\thirdparty\wheel-0.45.0 directory.

Also, add the flag `--no-build-isolation` for the upcoming commands when installing in an offline environment. For example, `py -m pip install . --no-build-isolation`

Important

Install the following third-party libraries in the order they appear and unzip the files in Administrator mode.

3. Unzip the <RPS>\thirdparty\more-itertools-10.1.0.zip file.
4. Run `py -m pip install .` from the <RPS>\thirdparty\more-itertools-10.1.0 directory.
5. Unzip the <RPS>\thirdparty\jaraco.functools-4.0.0.zip file.
6. Run `py -m pip install .` from the <RPS>\thirdparty\jaraco.functools-4.0.0 directory.
7. Unzip the <RPS>\thirdparty\cheroot-10.0.0.zip file.
8. Run `py -m pip install .` from the <RPS>\thirdparty\cheroot-10.0.0 directory.
9. Unzip the <RPS>\thirdparty\web.py-0.62.zip file.
10. Run `py -m pip install .` from the <RPS>\thirdparty\web.py-0.62 directory.
11. Unzip the <RPS>\thirdparty\pyparsing-3.1.1.zip file.
12. Run `py -m pip install .` from the <RPS>\thirdparty\pyparsing-3.1.1 directory.
13. Unzip the <RPS>\thirdparty\httplib2-0.22.0.zip file.
14. Run `py -m pip install .` from the <RPS>\thirdparty\httplib2-0.22.0 directory.
15. Unzip the <RPS>\thirdparty\six-1.16.0.zip file.
16. Run `py -m pip install .` from the <RPS>\thirdparty\six-1.16.0 directory.
17. Unzip the <RPS>\thirdparty\python-dateutil-2.8.2.zip file.

18. Run `py -m pip install .` from the `<RPS>\thirdparty\python-dateutil-2.8.2` directory.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).
2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `libffi-devel` (`yum install libffi-devel`).
4. Install OpenSSL.
 - For 8.5.500.11 or lower versions, install OpenSSL version 1.1.1.
 - For 8.5.500.13 or higher versions, install OpenSSL 3.0.13. Download OpenSSL 3.0.13 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-3.0.13 --openssldir=/usr/home/openssl-3.0.13 --libdir=lib no-shared`
5. Install 64 bit Python 3.11.5.
 - For 8.5.500.11 or lower versions, compile with OpenSSL 1.1.1 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
 - For 8.5.500.13 or higher versions, compile with OpenSSL 3.0.13 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-3.0.13 --enable-optimizations`
6. Install the **RPS IP** with the installer.

Important

Offline installation is supported starting from RPS 8.5.500.19.

To install it in a fully offline environment, follow these steps:

1. Unzip the `<RPS>\thirdparty\flit_core-3.10.1.zip` file.
2. Run `py -m pip install . --no-build-isolation` from the `<RPS>\thirdparty\flit_core-3.10.1` directory.
3. Unzip the `<RPS>\thirdparty\wheel-0.45.0.zip` file.
4. Run `py -m pip install . --no-build-isolation` from the `<RPS>\thirdparty\wheel-0.45.0` directory.

Also, add the flag `--no-build-isolation` for the upcoming commands when installing in an offline environment. For example, - `py -m pip install . --no-build-isolation`

Important

Install the following third-party libraries in the order they appear.

7. Untar the <RPS>/thirdparty/more-itertools-10.1.0.tar.gz file.
8. Run `python3 -m pip install .` from the <RPS>/thirdparty/more-itertools-10.1.0 directory.
9. Untar the <RPS>/thirdparty/jaraco.functools-4.0.0.tar.gz file.
10. Run `python3 -m pip install .` from the <RPS>/thirdparty/jaraco.functools-4.0.0 directory.
11. Untar the <RPS>/thirdparty/cheroot-10.0.0.tar.gz file.
12. Run `python3 -m pip install .` from the <RPS>/thirdparty/cheroot-10.0.0 directory.
13. Untar the <RPS>/thirdparty/web.py-0.62.tar.gz file.
14. Run `python3 -m pip install .` from the <RPS>/thirdparty/web.py-0.62 directory.
15. Untar the <RPS>/thirdparty/pyparsing-3.1.1.tar.gz file.
16. Run `python3 -m pip install .` from the <RPS>/thirdparty/pyparsing-3.1.1 directory.
17. Untar the <RPS>/thirdparty/httplib2-0.22.0.tar.gz file.
18. Run `python3 -m pip install .` from the <RPS>/thirdparty/httplib2-0.22.0 directory.
19. Untar the <RPS>/thirdparty/six-1.16.0.tar.gz file.
20. Run `python3 -m pip install .` from the <RPS>/thirdparty/six-1.16.0 directory.
21. Untar the <RPS>/thirdparty/python-dateutil-2.8.2.tar.gz file.
22. Run `python3 -m pip install .` from the <RPS>/thirdparty/python-dateutil-2.8.2 directory.

Important

- GIR does not support direct upgrade of RPS from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip install` command as mentioned above.

Configuring Recording Processor Script

This section describes how to configure the Recording Processor Script for your environment.

Configuring High Availability

Recording Processor Cluster

RPS now provides High Availability support using multiple instances of RPS (all active). These active/active instances must be accessed through an HA proxy or load balancer. In this mode, each RPS is responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner based on the load it receives. Each Recording Processor is responsible for fetching metadata from all ICON DB Servers. Media Control Platform

(MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In each Recording Processor's **rpconfig.cfg** configuration file, in the **[processing]** section, set the following options:
 - **get_from_http_before_posting** = 1
 - **mode** = active
2. Ensure that all Recording Processor instances have the *same* network related configuration.

Important

Genesys recommends that multiple Recording Processor instances be deployed on a single host to optimize the available CPU and take advantage of parallel processing. Multiple Recording Processor instances can then be deployed on other hosts as needed.

3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the **Recording Processor URI** parameter to the load balancer's URL.
4. Configure the load balancer to balance traffic to the Recording Processor instances.

The following is an example configuration section that is needed for setting up an Apache load balancer for a three-instance Recording Processor cluster.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
  BalancerMember http://<IP address of the active1 Recording Processor server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active2 Recording Processor Server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active3 Recording Processor Server>:<active
Recording Processor port>
</Proxy>
```

Important

SpeechMiner version 8.5.2 or later is required for the Recording Processor cluster support to work properly.

Recording Processor Script Active/Backup HA

RPS can also provide High Availability support by using two RPS instances (active and backup) accessed through an HA proxy or load balancer in failover mode. In this mode, the active RPS is always responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner, and the backup instance is responsible for receiving and temporarily storing metadata if the active instance is unavailable. Once the active instance recovers, the balancer will direct clients to the active instance, and the backup instance will

send any stored data to the active instance for metadata processing. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In the active Recording Processor's **rpconfig.cfg** configuration file located in the **[processing]** section, set the **mode** parameter to active.
2. In the backup Recording Processor's **rpconfig.cfg** configuration file:
 - Set the **mode** parameter to backup.
 - In the **[processing]** section, set the **post_uri** parameter to `http://<active_rp_ip>:<active_rp_port>/api/contact-centers/%s/recordings/`.
3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the Recording Processor URI parameter to the failover load balancer's URL.
4. Configure the load balancer to direct traffic to the active Recording Processor instance first and to the backup instance if an error/failure occurs.

The following is an example configuration section that is needed for setting up an Apache load balancer in failover mode for Recording HA support.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
    BalancerMember http://<IP address of the active Recording Processor server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the backup Recording Processor Server>:<active
Recording Processor port> status=H
</Proxy>
```

For more information about how to use Genesys Administrator Extension to configure your Contact Center, see the [Genesys Administrator Extension Help](#).

Configure Passwords

Important

In a Linux or Windows environment, RPS supports reading the environment variables for password related configuration parameters in order to avoid storing the password in plain-text in the configuration file. When both are available, the environment variables take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

HTCC_PASSWORD - maps to the existing configuration parameter under the **htcc** section, password value.
AUTH_PASSWORD - maps to the existing configuration parameter under the **auth** section, password value.
CONFIG_SERVER_PASSWORD - maps to the existing configuration parameter under the **config_server** section, password value.
<ICON_ID>_DB_INFO_PASSWORD - maps to the existing configuration parameter under the

<ICON_ID>_db_info section, password value, where <ICON_ID> refers to the ICON instance listed in the icon_db_servers section.

For example, if you have VCCSIPSwitch: icon1 the environment variable that corresponds to the icon1_db_info password is icon1_DB_INFO_PASSWORD.

In a Windows environment only, the Recording Processor Script (RPS) can store passwords in the Windows Vault instead of in the **rpconfig.cfg** file or requiring the use of environment variables.

For example, run the following command for the Recording Processor Script credentials located at <Recording Processor Directory>\rp. This command will prompt the user to enter valid values for the password/key configuration parameters and stores the passwords in the encrypted file named **rp.secret**:

Command to store:

```
encryptPassword.bat -password <password_string>
```

Where <password_string> is a comma-delimited series of key/value pairs, use the format <environment variable name 1>=<environment variable value 1>,<environment variable name 2>=<environment variable value 2>,<environment variable name 3>=<environment variable value 3>, and so on. Note that space is not allowed in <password_string>.

For example:

```
encryptPassword.bat -password "HTCC_PASSWORD=somepassword1, AUTH_PASSWORD=somepassword2, CONFIG_SERVER_PASSWORD=somepassword3, ICON1_DB_INFO_PASSWORD=somepassword4, ICON2_DB_INFO_PASSWORD=somepassword5"
```

Important

Passwords used with this command cannot contain a comma or an equals sign.

Configure the Configuration Server Connection

To configure the Configuration Server connection, set the following parameters in the **[config_server]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
application_name	Empty	Specifies the name of the RPS application object in the Configuration Server, when using RPS as a third party server application.
hostname	<ip address>	Specifies the IP address of the primary Configuration Server.
port	2020	Specifies the port of the primary

Parameter Name	Default Value	Description
		Configuration Server.
username	default	Specifies the Configuration Server username.
password	password	Specifies the Configuration Server password. Note: The password can be overridden by the CONFIG_SERVER_PASSWORD environment variable.
backup_host	Empty	Specifies the IP address of the backup Configuration Server.
backup_port	Empty	Specifies the backup port of the backup Configuration Server.

Important

Recording Processor Script does not support a secure connection to the Configuration Server.

Configuring the Server Port

In the **[rp_server]** section of the **rpconfig.cfg** file, set the **port** parameter.

Important

You can also set the "port" parameter using the command line with the --port command line argument. The command line argument takes precedence over the configuration file value.

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri	http://<Web Services IP>:<Web Services Port>	Specifies the Base URI for accessing the Interaction Recording Web Services (Web Services) API.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services)

Parameter Name	Default Value	Description
		account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note: The password can be overridden by the HTCC_PASSWORD environment variable.</p>

Each Interaction Recording Web Services (Web Services) instance must have a region associated with it. Set the region parameter in the [metadata] section of the `rpconfig.cfg` file to match the region associated with Interaction Recording Web Services (Web Services) instance set to receive the Recording Processor's metadata.

Configuring Cross-Site Request Forgery (CSRF) Protection

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled, set the following parameter in the **[htcc]** section of the **rpconfig.cfg** file:

- **csrfp** = 1

Configuring the Connection to SpeechMiner

To configure the SpeechMiner Connection:

1. In the IVR Profile, set the recording destinations to point to the SpeechMiner interaction receiver:
 - a. Login to Genesys Administrator Extension, and navigate to **Configuration > System > Configuration Manager**.
 - b. Under **Voice Platform**, select **Voice Platform Profiles**.
 - c. Click on the IVR Profile for which you want to set the recording destination.
 - d. Select the **Recording** tab.
 - e. In the **SpeechMiner Interaction Receiver** field, enter the URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile. For example, `https://<SpeechMiner IP>/interactionreceiver`.
 - f. In the **SpeechMiner Interaction Receiver Authorization Header** field, enter the authorization information (username:password) required to connect to the SpeechMiner service used by the RPS. For example, `user:password`.

Important

The values of these options must match the corresponding configuration options in the SpeechMiner system.

Configuring Failed Message Files

The Recording processor can backup messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner. These files are located in the **<recording processor dir>\failed** folder.

In the **rpconfig.cfg** configuration file, add the following parameter:

```
[processing]
backup_failed_metadata = 1
```

Configuring the Agent Hierarchies

Recording Processor Script uses the agent hierarchy information to set the access control information for recordings within the recording metadata. Refer to [Access Control for Genesys Interaction Recording Users](#) to configure this appropriately.

Configuring Basic Authorization

In the **rpconfig.cfg** configuration file, set the following parameters:

```
[auth]
# Basic Authentication username and password. Set username blank to disable.
username = rp_username
password = rp_password
```

Important

- The username and password must match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.
- The password can be overridden by the **AUTH_PASSWORD** environment variable.

Configuring After Call Work

Recording Processor can collect After Call Work (ACW) customized data from ICON.

In the **rpconfig.cfg** file, in the **[processing]** section, add the following parameters:

- **enable_acw**—Set it to 1.
- **acw_threshold_minutes**—Set it to the maximum time to wait for the customized attached data.

Important

- If Call Customized Attached Data is still not available in the ICON database after *acw_threshold_minutes*, the RPS will stop collecting customized data for this recording and write it to the database.
- If *enable_acw* is set to 0, ACW customized data will not be included.
- If disposition code is required in the metadata, you must set **enable_acw** and **acw_threshold_minutes** using Recording Processor configuration. The disposition code is part of the user data collected during ACW. For this reason, **enable_acw** must be enabled in the Recording Processor. If it is not enabled, the data will not be collected. If the disposition code must be collected from the Recording Processor, configure the following to include the disposition code for recording:

```
[processing] enable_acw=1
[metadata] acw_threshold_minutes=5.
```

Where 5 is the maximum time (in minutes) to wait for the disposition code. In the ICON configuration, the **EventData** parameter in the **custom-states** section, must include `char,DispositionCode` and **store-event-data** must be set to `conf` to collect the attached data:

```
[custom-states] store-event-data=conf      EventData=char,DispositionCode
```

For additional information, refer to the [ICON Deployment Guide](#).

Configuring ICON for Recording Processor

Important

When configuring Recording Processor to connect to a primary and backup ICON Database in HA mode, two separate DB Servers must be used. The DB Servers must run in an active/active pair mode.

To configure ICON, edit the **rpconfig.cfg** configuration file as follows:

1. Configure the switches:

Add a configuration option for each switch name under the **[icon_db_servers]** section. You can specify more than two ICON databases per SIP Switch configuration. For example:

```
[icon_db_servers]
SIP_Switch1: icon1
SIP_Switch2: icon2, icon2Backup
SIP_Switch3: icon3, icon4, icon5, icon6
```

Important

In the above example, **SIP_Switch3** has 4 ICON databases. The Recording Processor Script (RPS) keeps track of the ICON database instance currently used. If the current database instance becomes unavailable, RPS will

attempt the operation in the next database.

The configuration option name must match the exact name of the switch as configured in the Genesys configuration. The primary and backup ICON names must be unique, but do not have to match anything in the Genesys configuration.

2. Configure the ICON Connection Settings:

- For each unique ICON specified in the first step, create a new section using the following syntax: `<ICON_ID>_db_info`, where `<ICON_ID>` corresponds to the values defined in the **[icon_db_servers]** section above.
- **dbengine** must be `mssql`, `oracle`, `db2`, or `postgres`.
- **dbserver_host** and **dbserver_port** specify the host and port information for the Genesys DB Server.
- **dbms** specifies the host where the database resides.

The following is an example using the values for **SIP_Switch1** and **SIP_Switch2** from step 1:

```
[icon1_db_info]
dbserver_host = vm221.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2_db_info]
dbserver_host = vm222.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2Backup_db_info]
dbserver_host = vm223.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon_oracle_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host/Oracle SID>
```

```
dbengine = oracle

[icon_postgres_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname = <database name>
dbms = <database host>
dbengine = postgres

[icon_db2_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host>
dbengine = db2
```

Important

- For Oracle or DB2 implementations, the **dbname** parameter must be left blank or empty.
- The password can be overridden by the **<ICON_ID>_DB_INFO_PASSWORD** environment variable.

In the example above, the RPS will use the connection properties in section **[icon1_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch1. The RPS will use the connection properties in section **[icon2_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch2. In the case of SIP_Switch2, the RPS will use the connection settings in **[icon2Backup_db_info]** if the primary ICON (icon2) is unavailable when recording metadata is being processed.

Configure how to Filter Metadata from ICON

The Recording Processor supports the ability to filter specific attached data fields (based on the key name), such as attached data and After Call Work (ACW) customized data retrieved from the ICON database. This support prevents specific metadata from reaching additional GIR related components (for example, SpeechMiner).

The following two sections describe how to:

- **Filter attached data.**
- **Filter ACW.**

Important

- Verify that the following items are not removed from the filter. Removing these items may cause errors in GIR:
 - GRECORD_PARTITIONS
 - GRECORD_PROGRAM
 - GSIP_REC_FN
- When running SpeechMiner, you must include Workspace Web Edition (WWE) in the attached_data_filter and acw_custom_data_filter Recording Processor configuration values. For example:


```
[filter]
attached_data_filter=^ORSI:|^WWE
acw_custom_data_filter=^ORSI:|^WWE
```

Filter Attached Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **attached_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out attached data whose key name matches the pattern.

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **attached_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude key names that should not be filtered out (for example, like GRECORD_PARTITIONS).

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
attached_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the default
value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Filter ACW Related Custom Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **acw_custom_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out ACW whose key name matches the pattern.

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **acw_custom_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude ACW that should not be filtered out (for example, like **GRECORD_PARTITIONS**).

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
acw_custom_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the
default value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Configuring SSL for Recording Processor

To configure SSL:

Configure HTTPS on the Primary Recording Processor Server

1. Create a self-signed certificate and private key for the Recording Processor host. For example, on RHEL run: `openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem`
2. In the **rp_server** section of the Recording Processor's configuration file, set the following parameters:
 - **ssl_certificate**—To point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - **ssl_private_key**—To point to the private key file. For example, `cert228.pem`.
3. Give the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section [Genesys Voice Platform 8.5 User's Guide](#).

4. Restart Recording Processor.

Configure the HTTPS connection to Interaction Recording Web Services (Web Services)

1. Set up HTTPS on Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). See the [Genesys Security Deployment Guide](#).
2. Get the corresponding certificate for the Interaction Recording Web Services (Web Services) server. Set the **caCertificate** option in your Interaction Recording Web Services application (see **caCertificate** if you're using a Web Services application).
3. In the **[htcc]** section of the Recording Processor configuration file, set **base_uri** parameter to use https.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection to SpeechMiner

1. Set up HTTPS on SpeechMiner. See the [Genesys Security Deployment Guide](#).
2. Set the **disable_ssl_certificate_validation** parameter in the **[speechminer]** section of the Recording Processor configuration to a value of 1.
3. Using Genesys Administrator Extension on the Recording tab of the IVR Profile, modify the SpeechMiner Interaction Receiver field use https as the protocol in the URL.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA cert file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection from the backup Recording Processor to the primary Recording Processor

1. Configure HTTPS on the primary Recording Processor.
2. Get the corresponding PEM certificate for the Web Services server.
3. In the **[processing]** section of the Recording Processor configuration file, set the **post_uri** parameter

to use https as the protocol in the URL.

4. In the **client** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS on the backup Recording Processor Server

Follow the same procedure used for the Primary Recording Processor Server using a new certificate and private key for the Backup Recording Processor's server.

Configuring the IVR Profile

Using Genesys Administrator Extension, configure the following parameters on the **Recording** tab of the IVR Profile:

1. **Recording Processor URI**—The URI that the Media Control Platform (MCP) uses to post the metadata of the audio recording after the recording is complete. For example, `http:// <Recording Processor Host>/api/contact-centers/<Contact Cente Domain Name>/recordings/`.

Important

The value for the URI must always end with a forward slash (/).

2. **SpeechMiner Interaction Receiver**—The URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile.
3. **SpeechMiner Interaction Receiver Authorization Header**—The authorization information required to connect to the SpeechMiner service used by the RPS. For example, `<SpeechMiner Webserver Username>:<SpeechMiner Webserver Password>`.

For more information, see the [Configuring GVP](#).

Configuring the Recording Processor Using Genesys Administrator Extension (Optional)

The Recording Processor uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Processor as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Processor as a "third party server" application in

Genesys Administrator Extension. For more information, see the "Using the Management Layer" section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring RPS to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest RPS.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Processor Script of type Third Party Server.
4. Create a new application (for example, myRPS) in Genesys Administrator Extension using this new application template.
5. Set the Command Line parameter (for example, C:\Python311\python.exe).
6. Set the Host parameter in the application's server info to the correct Host object.
7. Set the Working Directory parameter to the <Recording Processor Install Directory>\rp directory. For example, /opt/genesys/Recording_Processor_Script_8.5/rp/.
8. Set the Command Line Arguments parameter to the appropriate values. For example, recording_process.py --config-file=/opt/genesys/Recording_Processor_Script_8.5/rp/rpconfig.cfg.
Refer to the [Starting the Recording Processor Script](#) section for additional command line parameters
9. Make sure that LCA has permission to read and write to the Recording Processor installation directory and Recording Processor log directory.
10. Save the configuration changes.
11. Ensure that the Configuration Server parameters in the Recording Processor configuration file are set appropriately. Refer to **Configure the Configuration Server Connection** tab on this page.

Important

The Recording Processor does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

For more information about the RPS options, see [Genesys Interaction Recording Options Reference](#).

Starting the Recording Processor Script

To launch the RPS, run the following command from the <Recording Processor Install Directory>:

```
<python executable> recording_process.py --config-file=rpconfig.cfg
```

Use the following command line when you want to run multiple instances of RPS on the same machine:


```
<python executable> recording_process.py --config-file=rpconfig.cfg --id=1 --port=8889
```

For each RPS instance, assign a unique id (--id parameter) and port number (--port).

Important

- --port defines the server port opened by the RPS process.
- --id represents the suffix of the:
 - application_name in the configuration file. For example, if application_name is defined in the configuration file as **RecordingProcessorScript** and --id 2 is specified in the command line, then the application object named **RecordingProcessorScript_2** will be used to start the program.
 - log files
 - metadata json files created in the failed folder
 - database file created by the process

By default the RPS log file is stored in the working directory. This can be changed by specifying a preexisting folder in the logfile_path parameter in the log file section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingProcessor
```

Recording Processor Script (Python 3) RHEL 7

Important

Voice Processor, a multi-threaded microservice based on the Node.js platform, is an alternative to the Recording Processor Script (RPS).

- For information on deploying Voice Processor, see [Deploying Voice Processor](#).
- To migrate from an existing RPS deployment to Voice Processor, see [Migrating from RPS to Voice Processor](#).
- RPS is not supported for deployments integrated with SIP Cluster. If your deployment uses SIP Cluster, you must use Voice Processor.

For new deployments, Genesys recommends using Voice Processor instead of RPS.

Prerequisites

Before installing and configuring the RPS, you must have the following prerequisites:

- An **Interaction Recording Web Services** 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.
- A **Recording Crypto Server** 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.

Installing Recording Processor Script

Installing on Windows

1. Install 64 bit Python 3.11.5 from the **Python** website.
2. Install the **RPS IP** with the installer. **Note:** Install the following third-party libraries in the order they appear and unzip the files in Administrator mode.
3. Unzip the <RPS>\thirdparty\more-itertools-10.1.0.zip file.
4. Run `py -m pip install .` from the <RPS>\thirdparty\more-itertools-10.1.0 directory.
5. Unzip the <RPS>\thirdparty\jaraco.functools-4.0.0.zip file.
6. Run `py -m pip install .` from the <RPS>\thirdparty\jaraco.functools-4.0.0 directory.
7. Unzip the <RPS>\thirdparty\cheroot-10.0.0.zip file.
8. Run `py -m pip install .` from the <RPS>\thirdparty\cheroot-10.0.0 directory.
9. Unzip the <RPS>\thirdparty\web.py-0.62.zip file.
10. Run `py -m pip install .` from the <RPS>\thirdparty\web.py-0.62 directory.
11. Unzip the <RPS>\thirdparty\pyparsing-3.1.1.zip file.
12. Run `py -m pip install .` from the <RPS>\thirdparty\pyparsing-3.1.1 directory.
13. Unzip the <RPS>\thirdparty\httplib2-0.22.0.zip file.
14. Run `py -m pip install .` from the <RPS>\thirdparty\httplib2-0.22.0 directory.
15. Unzip the <RPS>\thirdparty\six-1.16.0.zip file.
16. Run `py -m pip install .` from the <RPS>\thirdparty\six-1.16.0 directory.
17. Unzip the <RPS>\thirdparty\python-dateutil-2.8.2.zip file.
18. Run `py -m pip install .` from the <RPS>\thirdparty\python-dateutil-2.8.2 directory.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).

2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `libffi-devel` (`yum install libffi-devel`).
4. Install OpenSSL 1.1.1.
 - For RHEL 7:
 1. Download OpenSSL 1.1.1 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-1.1.1 --openssldir=/usr/home/openssl-1.1.1`
 2. Add OpenSSL lib path in `LD_LIBRARY_PATH`. Example command - `export LD_LIBRARY_PATH=/usr/home/openssl-1.1.1/lib:$LD_LIBRARY_PATH`
5. Install 64 bit Python 3.11.5 compiled with OpenSSL 1.1.1 from the [Python](#) website.
 - While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
6. Install the **RPS IP** with the installer. **Note:** Install the following third-party libraries in the order they appear.
7. Untar the `<RPS>/thirdparty/more-itertools-10.1.0.tar.gz` file.
8. Run `python3 -m pip install .` from the `<RPS>/thirdparty/more-itertools-10.1.0` directory.
9. Untar the `<RPS>/thirdparty/jaraco.functools-4.0.0.tar.gz` file.
10. Run `python3 -m pip install .` from the `<RPS>/thirdparty/jaraco.functools-4.0.0` directory.
11. Untar the `<RPS>/thirdparty/cheroot-10.0.0.tar.gz` file.
12. Run `python3 -m pip install .` from the `<RPS>/thirdparty/cheroot-10.0.0` directory.
13. Untar the `<RPS>/thirdparty/web.py-0.62.tar.gz` file.
14. Run `python3 -m pip install .` from the `<RPS>/thirdparty/web.py-0.62` directory.
15. Untar the `<RPS>/thirdparty/pyparsing-3.1.1.tar.gz` file.
16. Run `python3 -m pip install .` from the `<RPS>/thirdparty/pyparsing-3.1.1` directory.
17. Untar the `<RPS>/thirdparty/httplib2-0.22.0.tar.gz` file.
18. Run `python3 -m pip install .` from the `<RPS>/thirdparty/httplib2-0.22.0` directory.
19. Untar the `<RPS>/thirdparty/six-1.16.0.tar.gz` file.
20. Run `python3 -m pip install .` from the `<RPS>/thirdparty/six-1.16.0` directory.
21. Untar the `<RPS>/thirdparty/python-dateutil-2.8.2.tar.gz` file.
22. Run `python3 -m pip install .` from the `<RPS>/thirdparty/python-dateutil-2.8.2` directory.

Important

- GIR does not support direct upgrade of RPS from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip install` command as mentioned above.

Configuring Recording Processor Script

This section describes how to configure the Recording Processor Script for your environment.

Configuring High Availability

Recording Processor Cluster

RPS now provides High Availability support using multiple instances of RPS (all active). These active/active instances must be accessed through an HA proxy or load balancer. In this mode, each RPS is responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner based on the load it receives. Each Recording Processor is responsible for fetching metadata from all ICON DB Servers. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In each Recording Processor's **rpconfig.cfg** configuration file, in the **[processing]** section, set the following options:
 - **get_from_httpc_before_posting** = 1
 - **mode** = active
2. Ensure that all Recording Processor instances have the *same* network related configuration.

Important

Genesys recommends that multiple Recording Processor instances be deployed on a single host to optimize the available CPU and take advantage of parallel processing. Multiple Recording Processor instances can then be deployed on other hosts as needed.

3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the **Recording Processor URI** parameter to the load balancer's URL.
4. Configure the load balancer to balance traffic to the Recording Processor instances.

The following is an example configuration section that is needed for setting up an Apache load balancer for a three-instance Recording Processor cluster.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
  BalancerMember http://<IP address of the active1 Recording Processor server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active2 Recording Processor Server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the active3 Recording Processor Server>:<active
Recording Processor port>
```

```
</Proxy>
```

Important

SpeechMiner version 8.5.2 or later is required for the Recording Processor cluster support to work properly.

Recording Processor Script Active/Backup HA

RPS can also provide High Availability support by using two RPS instances (active and backup) accessed through an HA proxy or load balancer in failover mode. In this mode, the active RPS is always responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner, and the backup instance is responsible for receiving and temporarily storing metadata if the active instance is unavailable. Once the active instance recovers, the balancer will direct clients to the active instance, and the backup instance will send any stored data to the active instance for metadata processing. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In the active Recording Processor's **rpconfig.cfg** configuration file located in the **[processing]** section, set the **mode** parameter to active.
2. In the backup Recording Processor's **rpconfig.cfg** configuration file:
 - Set the **mode** parameter to backup.
 - In the **[processing]** section, set the **post_uri** parameter to `http://<active_rp_ip>:<active_rp_port>/api/contact-centers/%s/recordings/`.
3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the Recording Processor URI parameter to the failover load balancer's URL.
4. Configure the load balancer to direct traffic to the active Recording Processor instance first and to the backup instance if an error/failure occurs.

The following is an example configuration section that is needed for setting up an Apache load balancer in failover mode for Recording HA support.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
  BalancerMember http://<IP address of the active Recording Processor server>:<active
Recording Processor port>
  BalancerMember http://<IP address of the backup Recording Processor Server>:<active
Recording Processor port> status=H
</Proxy>
```

For more information about how to use Genesys Administrator Extension to configure your Contact Center, see the [Genesys Administrator Extension Help](#).

Configure Passwords

Important

In a Linux or Windows environment, RPS supports reading the environment variables for password related configuration parameters in order to avoid storing the password in plain-text in the configuration file. When both are available, the environment variables take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

HTCC_PASSWORD - maps to the existing configuration parameter under the **htcc** section, password value.

AUTH_PASSWORD - maps to the existing configuration parameter under the **auth** section, password value.

CONFIG_SERVER_PASSWORD - maps to the existing configuration parameter under the **config_server** section, password value.

<ICON_ID>_DB_INFO_PASSWORD - maps to the existing configuration parameter under the **<ICON_ID>_db_info** section, password value, where **<ICON_ID>** refers to the ICON instance listed in the **icon_db_servers** section.

For example, if you have VCCSIPSwitch: icon1 the environment variable that corresponds to the icon1_db_info password is icon1_DB_INFO_PASSWORD.

In a Windows environment only, the Recording Processor Script (RPS) can store passwords in the Windows Vault instead of in the **rpconfig.cfg** file or requiring the use of environment variables.

For example, run the following command for the Recording Processor Script credentials located at **<Recording Processor Directory>\rp**. This command will prompt the user to enter valid values for the password/key configuration parameters and stores the passwords in the encrypted file named **rp.secret**:

Command to store:

```
encryptPassword.bat -password <password_string>
```

Where **<password_string>** is a comma-delimited series of key/value pairs, use the format **<environment variable name 1>=<environment variable value 1>,<environment variable name 2>=<environment variable value 2>,<environment variable name 3>=<environment variable value 3>**, and so on. Note that space is not allowed in **<password_string>**.

For example:

```
encryptPassword.bat -password "HTCC_PASSWORD=somepassword1, AUTH_PASSWORD=somepassword2, CONFIG_SERVER_PASSWORD=somepassword3, ICON1_DB_INFO_PASSWORD=somepassword4, ICON2_DB_INFO_PASSWORD=somepassword5"
```

Important

Passwords used with this command cannot contain a comma or an equals sign.

Configure the Configuration Server Connection

To configure the Configuration Server connection, set the following parameters in the **[config_server]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
application_name	Empty	Specifies the name of the RPS application object in the Configuration Server, when using RPS as a third party server application.
hostname	<ip address>	Specifies the IP address of the primary Configuration Server.
port	2020	Specifies the port of the primary Configuration Server.
username	default	Specifies the Configuration Server username.
password	password	Specifies the Configuration Server password. Note: The password can be overridden by the CONFIG_SERVER_PASSWORD environment variable.
backup_host	Empty	Specifies the IP address of the backup Configuration Server.
backup_port	Empty	Specifies the backup port of the backup Configuration Server.

Important

Recording Processor Script does not support a secure connection to the Configuration Server.

Configuring the Server Port

In the **[rp_server]** section of the **rpconfig.cfg** file, set the **port** parameter.

Important

You can also set the "port" parameter using the command line with the --port command line argument. The command line argument takes precedence over the configuration file value.

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri	http://<Web Services IP>:<Web Services Port>	Specifies the Base URI for accessing the Interaction Recording Web Services (Web Services) API.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	Specifies the password used to access the Interaction Recording Web Services (Web Services) account. Note: The password can be overridden by the HTCC_PASSWORD environment variable.

Each Interaction Recording Web Services (Web Services) instance must have a region associated with it. Set the region parameter in the [metadata] section of the rpconfig.cfg file to match the region associated with Interaction Recording Web Services (Web Services) instance set to receive the Recording Processor's metadata.

Configuring Cross-Site Request Forgery (CSRF) Protection

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled, set the following parameter in the **[htcc]** section of the **rpconfig.cfg** file:

- **csrfp = 1**

Configuring the Connection to SpeechMiner

To configure the SpeechMiner Connection:

1. In the IVR Profile, set the recording destinations to point to the SpeechMiner interaction receiver:
 - a. Login to Genesys Administrator Extension, and navigate to **Configuration > System > Configuration Manager**.
 - b. Under **Voice Platform**, select **Voice Platform Profiles**.
 - c. Click on the IVR Profile for which you want to set the recording destination.
 - d. Select the **Recording** tab.
 - e. In the **SpeechMiner Interaction Receiver** field, enter the URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile. For example, `https://<SpeechMiner IP>/interactionreceiver`.

- f. In the **SpeechMiner Interaction Receiver Authorization Header** field, enter the authorization information (username:password) required to connect to the SpeechMiner service used by the RPS. For example, user:password.

Important

The values of these options must match the corresponding configuration options in the SpeechMiner system.

Configuring Failed Message Files

The Recording processor can backup messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner. These files are located in the **<recording processor dir>\failed** folder.

In the **rpconfig.cfg** configuration file, add the following parameter:

```
[processing]
backup_failed_metadata = 1
```

Configuring the Agent Hierarchies

Recording Processor Script uses the agent hierarchy information to set the access control information for recordings within the recording metadata. Refer to [Access Control for Genesys Interaction Recording Users](#) to configure this appropriately.

Configuring Basic Authorization

In the **rpconfig.cfg** configuration file, set the following parameters:

```
[auth]
# Basic Authentication username and password. Set username blank to disable.
username = rp_username
password = rp_password
```

Important

- The username and password must match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.
- The password can be overridden by the **AUTH_PASSWORD** environment variable.

Configuring After Call Work

Recording Processor can collect After Call Work (ACW) customized data from ICON.

In the **rpconfig.cfg** file, in the **[processing]** section, add the following parameters:

- **enable_acw**—Set it to 1.
- **acw_threshold_minutes**—Set it to the maximum time to wait for the customized attached data.

Important

- If Call Customized Attached Data is still not available in the ICON database after *acw_threshold_minutes*, the RPS will stop collecting customized data for this recording and write it to the database.
- If *enable_acw* is set to 0, ACW customized data will not be included.
- If disposition code is required in the metadata, you must set **enable_acw** and **acw_threshold_minutes** using Recording Processor configuration. The disposition code is part of the user data collected during ACW. For this reason, **enable_acw** must be enabled in the Recording Processor. If it is not enabled, the data will not be collected. If the disposition code must be collected from the Recording Processor, configure the following to include the disposition code for recording:
[processing] enable_acw=1
[metadata] acw_threshold_minutes=5. Where 5 is the maximum time (in minutes) to wait for the disposition code. In the ICON configuration, the **EventData** parameter in the **custom-states** section, must include char,DispositionCode and **store-event-data** must be set to conf to collect the attached data:
[custom-states] store-event-data=conf EventData=char,DispositionCode
For additional information, refer to the [ICON Deployment Guide](#).

Configuring ICON for Recording Processor

Important

When configuring Recording Processor to connect to a primary and backup ICON Database in HA mode, two separate DB Servers must be used. The DB Servers must run in an active/active pair mode.

To configure ICON, edit the **rpconfig.cfg** configuration file as follows:

1. Configure the switches:

Add a configuration option for each switch name under the **[icon_db_servers]** section. You can specify more than two ICON databases per SIP Switch configuration. For example:

```
[icon_db_servers]
SIP_Switch1: icon1
SIP_Switch2: icon2, icon2Backup
SIP_Switch3: icon3, icon4, icon5, icon6
```

Important

In the above example, **SIP_Switch3** has 4 ICON databases. The Recording Processor Script (RPS) keeps track of the ICON database instance currently used. If the current database instance becomes unavailable, RPS will attempt the operation in the next database.

The configuration option name must match the exact name of the switch as configured in the Genesys configuration. The primary and backup ICON names must be unique, but do not have to match anything in the Genesys configuration.

2. Configure the ICON Connection Settings:

- For each unique ICON specified in the first step, create a new section using the following syntax: `<ICON_ID>_db_info`, where `<ICON_ID>` corresponds to the values defined in the **[icon_db_servers]** section above.
- **dbengine** must be `mssql`, `oracle`, `db2`, or `postgres`.
- **dbserver_host** and **dbserver_port** specify the host and port information for the Genesys DB Server.
- **dbms** specifies the host where the database resides.

The following is an example using the values for **SIP_Switch1** and **SIP_Switch2** from step 1:

```
[icon1_db_info]
dbserver_host = vm221.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname     = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql

[icon2_db_info]
dbserver_host = vm222.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname     = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql

[icon2Backup_db_info]
dbserver_host = vm223.us.int.genesyslab.com
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname     = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql

[icon_oracle_db_info]
dbserver_host = <host>
dbserver_port = <port>
```

```
username = <username>
password = <password>
dbname =
dbms = <database host/Oracle SID>
dbengine = oracle

[icon_postgres_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname = <database name>
dbms = <database host>
dbengine = postgres

[icon_db2_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host>
dbengine = db2
```

Important

- For Oracle or DB2 implementations, the **dbname** parameter must be left blank or empty.
- The password can be overridden by the **<ICON_ID>_DB_INFO_PASSWORD** environment variable.

In the example above, the RPS will use the connection properties in section **[icon1_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch1. The RPS will use the connection properties in section **[icon2_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch2. In the case of SIP_Switch2, the RPS will use the connection settings in **[icon2Backup_db_info]** if the primary ICON (icon2) is unavailable when recording metadata is being processed.

Configure how to Filter Metadata from ICON

The Recording Processor supports the ability to filter specific attached data fields (based on the key name), such as attached data and After Call Work (ACW) customized data retrieved from the ICON database. This support prevents specific metadata from reaching additional GIR related components (for example, SpeechMiner).

The following two sections describe how to:

- **Filter attached data.**
- **Filter ACW.**

Important

- Verify that the following items are not removed from the filter. Removing these items may cause errors in GIR:
 - GRECORD_PARTITIONS
 - GRECORD_PROGRAM
 - GSIP_REC_FN
- When running SpeechMiner, you must include Workspace Web Edition (WWE) in the attached_data_filter and acw_custom_data_filter Recording Processor configuration values. For example:


```
[filter]
attached_data_filter=^ORSI|^WWE
acw_custom_data_filter=^ORSI|^WWE
```

Filter Attached Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **attached_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out attached data whose key name matches the pattern.

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **attached_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude key names that should not be filtered out (for example, like GRECORD_PARTITIONS).

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
attached_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the default
value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Filter ACW Related Custom Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **acw_custom_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out ACW whose key name matches the pattern.

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

4. Add a new option called **acw_custom_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude ACW that should not be filtered out (for example, like GRECORD_PARTITIONS).

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
acw_custom_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the
default value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/3.11/library/re.html>.

5. Restart the Recording Processor.

Configuring SSL for Recording Processor

To configure SSL:

Configure HTTPS on the Primary Recording Processor Server

1. Create a self-signed certificate and private key for the Recording Processor host. For example, on RHEL run: `openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem`
2. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—To point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228key.pem`.
3. Give the self-signed certificate PEM file to any MCP client that needs to validate the certificate during

the SSL handshake. See the "Enable Secure Communication" section [Genesys Voice Platform 8.5 User's Guide](#).

4. Restart Recording Processor.

Configure the HTTPS connection to Interaction Recording Web Services (Web Services)

1. Set up HTTPS on Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). See the [Genesys Security Deployment Guide](#).
2. Get the corresponding certificate for the Interaction Recording Web Services (Web Services) server. Set the **caCertificate** option in your Interaction Recording Web Services application (see **caCertificate** if you're using a Web Services application).
3. In the **[htcc]** section of the Recording Processor configuration file, set **base_uri** parameter to use https.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection to SpeechMiner

1. Set up HTTPS on SpeechMiner. See the [Genesys Security Deployment Guide](#).
2. Set the **disable_ssl_certificate_validation** parameter in the **[speechminer]** section of the Recording Processor configuration to a value of 1.
3. Using Genesys Administrator Extension on the Recording tab of the IVR Profile, modify the SpeechMiner Interaction Receiver field use https as the protocol in the URL.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA cert file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection from the backup Recording Processor to the primary Recording Processor

1. Configure HTTPS on the primary Recording Processor.

2. Get the corresponding PEM certificate for the Web Services server.
3. In the **[processing]** section of the Recording Processor configuration file, set the **post_uri** parameter to use https as the protocol in the URL.
4. In the **client** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS on the backup Recording Processor Server

Follow the same procedure used for the Primary Recording Processor Server using a new certificate and private key for the Backup Recording Processor's server.

Configuring the IVR Profile

Using Genesys Administrator Extension, configure the following parameters on the **Recording** tab of the IVR Profile:

1. **Recording Processor URI**—The URI that the Media Control Platform (MCP) uses to post the metadata of the audio recording after the recording is complete. For example, `http:// <Recording Processor Host>/api/contact-centers/<Contact Cente Domain Name>/recordings/`.

Important

The value for the URI must always end with a forward slash (/).

2. **SpeechMiner Interaction Receiver**—The URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile.
3. **SpeechMiner Interaction Receiver Authorization Header**—The authorization information required to connect to the SpeechMiner service used by the RPS. For example, `<SpeechMiner Webserver Username>:<SpeechMiner Webserver Password>`.

For more information, see the [Configuring GVP](#).

Configuring the Recording Processor Using Genesys Administrator Extension (Optional)

The Recording Processor uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Processor as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Processor as a "third party server" application in Genesys Administrator Extension. For more information, see the "Using the Management Layer" section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring RPS to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest RPS.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Processor Script of type Third Party Server.
4. Create a new application (for example, myRPS) in Genesys Administrator Extension using this new application template.
5. Set the Command Line parameter (for example, C:\Python311\python.exe).
6. Set the Host parameter in the application's server info to the correct Host object.
7. Set the Working Directory parameter to the <Recording Processor Install Directory>\rp directory. For example, /opt/genesys/Recording_Processor_Script_8.5/rp/.
8. Set the Command Line Arguments parameter to the appropriate values. For example, recording_process.py --config-file=/opt/genesys/Recording_Processor_Script_8.5/rp/rpconfig.cfg.
Refer to the [Starting the Recording Processor Script](#) section for additional command line parameters
9. Make sure that LCA has permission to read and write to the Recording Processor installation directory and Recording Processor log directory.
10. Save the configuration changes.
11. Ensure that the Configuration Server parameters in the Recording Processor configuration file are set appropriately. Refer to **Configure the Configuration Server Connection** tab on this page.

Important

The Recording Processor does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

For more information about the RPS options, see [Genesys Interaction Recording Options Reference](#).

Starting the Recording Processor Script

To launch the RPS, run the following command from the <Recording Processor Install Directory>:

```
<python executable> recording_process.py --config-file=rpconfig.cfg
```

Use the following command line when you want to run multiple instances of RPS on the same machine:

```
<python executable> recording_process.py --config-file=rpconfig.cfg --id=1 --port=8889
```

For each RPS instance, assign a unique id (--id parameter) and port number (--port).

Important

- --port defines the server port opened by the RPS process.
- --id represents the suffix of the:
 - application_name in the configuration file. For example, if application_name is defined in the configuration file as **RecordingProcessorScript** and --id 2 is specified in the command line, then the application object named **RecordingProcessorScript_2** will be used to start the program.
 - log files
 - metadata json files created in the failed folder
 - database file created by the process

By default the RPS log file is stored in the working directory. This can be changed by specifying a preexisting folder in the logfile_path parameter in the log file section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingProcessor
```

Recording Processor Script Legacy (Python 2) Deprecated

Important

Recording Processor Script Legacy (based on Python 2) has been discontinued as of March 31, 2024.

Important

Voice Processor, a multi-threaded microservice based on the Node.js platform, is an alternative to the Recording Processor Script (RPS).

- For information on deploying Voice Processor, see [Deploying Voice Processor](#).
- To migrate from an existing RPS deployment to Voice Processor, see [Migrating from RPS to Voice Processor](#).
- RPS is not supported for deployments integrated with SIP Cluster. If your deployment

uses SIP Cluster, you must use Voice Processor.

For new deployments, Genesys recommends using Voice Processor instead of RPS.

Genesys Interaction Recording (GIR) needs the Recording Processor Script (RPS) to manage the recording metadata between Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner.

Prerequisites

Before installing and configuring the RPS, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) (or [Web Services](#) if you're using version 8.5.210.02 or earlier) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.

Installing Recording Processor Script

Installing on Windows

1. Install 32 bit Python 2.7.5 or latest 2.7.x release from the [Python](#) website.
2. Install the RPS IP.
3. Unzip the <RPS>\thirdparty\httplib2-0.8.zip file.
4. From the newly created directory, run: `python setup.py install`.
5. Unzip the <RPS>\thirdparty\setuptools-1.3.2.zip file.
6. From the newly created directory, run: `python setup.py install`.
7. Unzip the <RPS>\thirdparty\python-dateutil-1.5.zip file.
8. From the newly created directory, run: `python setup.py install`.
9. Unzip the <RPS>\thirdparty\web.py-0.37.zip file.
10. From the newly created directory, run: `python setup.py install`.
11. Download pyOpenSSL for Windows (32 bit) from the [Python pyOpenSSL](#) site.
12. Install pyOpenSSL by running `pyOpenSSL-0.12.1.win32-py2.7.exe`.

Important

Installing pyOpenSSL (the previous steps) is optional, and is only required if an HTTPS server is needed for Recording Processor Script to receive metadata. Without it, only an HTTP server is supported.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).
2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `openssl-devel` (`yum install openssl-devel.x86_64`).
4. Install Python 2.7.5 or latest 2.7.x release from the [Python](#) website:
 - Genesys recommends that newer versions of Python are installed separately from existing versions (do not update).
 - See the [Example](#) below for an example of how to install CPython 2.7.6 on RHEL5.
5. Install/deploy the RPS IP.
6. Install `httplib2-0.8`:
 - a. Untar `httplib2-0.8.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `setuptools-1.3.2`:
 - a. Untar `setuptools-1.3.2.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `python-dateutil-1.5`:
 - a. Untar `python-dateutil-1.5.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `web.py-0.37`:
 - a. Untar `web.py-0.37.tar.gz` from the `thirdparty` directory in the RPS installation directory.
 - b. From the newly created directory, run: `python setup.py install`.
3. Install `pyOpenSSL-0.12`:
 - a. Download `pyOpenSSL-0.12.tar.gz` from the [pyOpenSSL 0.12](#) site.
 - b. Untar the downloaded file, `pyOpenSSL-0.12.tar.gz` to the RPS installation directory.
 - c. From the newly created directory, run the following command to build the library: `python setup.py build`.
 - d. Install the library, run: `python setup.py install`.

Important

Installing pyOpenSSL (the previous steps) is optional, and is only required if an HTTPS server is needed for Recording Processor Script to receive metadata. Without it, only an HTTP server is supported.

RPS on RHEL8 cannot build pyOpenSSL-0.12. You must download **openssl-1.0.1e.tar.gz** from <https://ftp.openssl.org/source/old/1.0.1>, build openssl-1.0.1e, proceed to build Python 2.7.18 normally, and then build pyOpenSSL-0.12 while also updating LD_LIBRARY_PATH to point to the openssl-1.0.1e libraries.

Example: Installing CPython 2.7.6 on RHEL5 (64bit)

The following instructions are intended as an example only. A specific system or environment may require different steps when installing CPython 2.7.6 on RHEL5 (64bit):

1. Verify that `zlib-devel` is installed on the OS (`yum install zlib-devel`).
2. Verify that `sqlite dev` is installed on the OS (`yum install sqlite-devel.x86_64`).
3. Verify that `openssl devel` is installed on the OS (`yum install openssl-devel.x86_64`).
4. Download CPython 2.7.6 source from the [Python](#) site.
5. Untar compressed source.
6. Run `./configure --enable-ipv6`.
7. Run `"make altinstall"` (this should prevent the overwriting of any existing versions).

Upgrading Recording Processor Script

1. Stop the RPS process.
2. Stop the RPS application.
3. Back up the RPS configuration file (`rpconfig.cfg`) and the `sqlite` file from the `\rp` directory (`rpqueue.db`).
4. Rename the existing installation folder name to `<folder name>.<old.current_date>` or something similar.
5. Uninstall the RPS component.
6. Install the new RPS component.
7. Copy the `rpconfig.cfg` and `rpqueue.db` files from the previous version into the `\rp` folder inside the new installation directory.
8. Start the new RPS application.
9. Repeat the above steps for additional RPS instances.

Configuring Recording Processor Script

This section describes how to configure the Recording Processor Script for your environment.

Configuring High Availability

Recording Processor Cluster

RPS now provides High Availability support using multiple instances of RPS (all active). These active/active instances must be accessed through an HA proxy or load balancer. In this mode, each RPS is responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner based on the load it receives. Each Recording Processor is responsible for fetching metadata from all ICON DB Servers. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In each Recording Processor's **rpconfig.cfg** configuration file, in the **[processing]** section, set the following options:
 - **get_from_httpc_before_posting** = 1
 - **mode** = active
2. Ensure that all Recording Processor instances have the *same* network related configuration.

Important

Genesys recommends that multiple Recording Processor instances be deployed on a single host to optimize the available CPU and take advantage of parallel processing. Multiple Recording Processor instances can then be deployed on other hosts as needed.

3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the **Recording Processor URI** parameter to the load balancer's URL.
4. Configure the load balancer to balance traffic to the Recording Processor instances.

The following is an example configuration section that is needed for setting up an Apache load balancer for a three-instance Recording Processor cluster.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
    BalancerMember http://<IP address of the active1 Recording Processor server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the active2 Recording Processor Server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the active3 Recording Processor Server>:<active
Recording Processor port>
</Proxy>
```

Important

SpeechMiner version 8.5.2 or later is required for the Recording Processor cluster support to work properly.

Recording Processor Script Active/Backup HA

RPS can also provide High Availability support by using two RPS instances (active and backup) accessed through an HA proxy or load balancer in failover mode. In this mode, the active RPS is always responsible for sending metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner, and the backup instance is responsible for receiving and temporarily storing metadata if the active instance is unavailable. Once the active instance recovers, the balancer will direct clients to the active instance, and the backup instance will send any stored data to the active instance for metadata processing. Media Control Platform (MCP) instances must be configured to access the Recording Processor instances by specifying a single URL that points to the load balancer.

To configure HA:

1. In the active Recording Processor's **rpconfig.cfg** configuration file located in the **[processing]** section, set the **mode** parameter to active.
2. In the backup Recording Processor's **rpconfig.cfg** configuration file:
 - Set the **mode** parameter to backup.
 - In the **[processing]** section, set the **post_uri** parameter to `http://<active_rp_ip>:<active_rp_port>/api/contact-centers/%s/recordings/`.
3. Using Genesys Administrator Extension, under the **Recording** tab in the IVR Profile, set the Recording Processor URI parameter to the failover load balancer's URL.
4. Configure the load balancer to direct traffic to the active Recording Processor instance first and to the backup instance if an error/failure occurs.

The following is an example configuration section that is needed for setting up an Apache load balancer in failover mode for Recording HA support.

```
ProxyPass /cluster balancer://nodecluster
<Proxy balancer://nodecluster>
    BalancerMember http://<IP address of the active Recording Processor server>:<active
Recording Processor port>
    BalancerMember http://<IP address of the backup Recording Processor Server>:<active
Recording Processor port> status=H
</Proxy>
```

For more information about how to use Genesys Administrator Extension to configure your Contact Center, see the [Genesys Administrator Extension Help](#).

Configure Passwords

Important

In a Linux or Windows environment, RPS supports reading the environment variables for password related configuration parameters in order to avoid storing the password in plain-text in the configuration file. When both are available, the environment variables take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

HTCC_PASSWORD - maps to the existing configuration parameter under the **htcc** section, password value.

AUTH_PASSWORD - maps to the existing configuration parameter under the **auth** section, password value.

CONFIG_SERVER_PASSWORD - maps to the existing configuration parameter under the **config_server** section, password value.

<ICON_ID>_DB_INFO_PASSWORD - maps to the existing configuration parameter under the **<ICON_ID>_db_info** section, password value, where **<ICON_ID>** refers to the ICON instance listed in the **icon_db_servers** section.

For example, if you have VCCSIPSwitch: icon1 the environment variable that corresponds to the icon1_db_info password is icon1_DB_INFO_PASSWORD.

In a Windows environment only, the Recording Processor Script (RPS) can store passwords in the Windows Vault instead of in the **rpconfig.cfg** file or requiring the use of environment variables.

For example, run the following command for the Recording Processor Script credentials located at **<Recording Processor Directory>\rp**. This command will prompt the user to enter valid values for the password/key configuration parameters and stores the passwords in the encrypted file named **rp.secret**:

Command to store:

```
encryptPassword.bat -password <password_string>
```

Where **<password_string>** is a comma-delimited series of key/value pairs, use the format **<environment variable name 1>=<environment variable value 1>,<environment variable name 2>=<environment variable value 2>,<environment variable name 3>=<environment variable value 3>**, and so on. Note that space is not allowed in **<password_string>**.

For example:

```
encryptPassword.bat -password "HTCC_PASSWORD=somepassword1, AUTH_PASSWORD=somepassword2, CONFIG_SERVER_PASSWORD=somepassword3, ICON1_DB_INFO_PASSWORD=somepassword4, ICON2_DB_INFO_PASSWORD=somepassword5"
```

Important

Passwords used with this command cannot contain a comma or an equals sign.

Configure the Configuration Server Connection

To configure the Configuration Server connection, set the following parameters in the **[config_server]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
application_name	Empty	Specifies the name of the RPS application object in the Configuration Server, when using RPS as a third party server application.
hostname	<ip address>	Specifies the IP address of the primary Configuration Server.
port	2020	Specifies the port of the primary Configuration Server.
username	default	Specifies the Configuration Server username.
password	password	Specifies the Configuration Server password. Note: The password can be overridden by the CONFIG_SERVER_PASSWORD environment variable.
backup_host	Empty	Specifies the IP address of the backup Configuration Server.
backup_port	Empty	Specifies the backup port of the backup Configuration Server.

Important

Recording Processor Script does not support a secure connection to the Configuration Server.

Configuring the Server Port

In the **[rp_server]** section of the **rpconfig.cfg** file, set the **port** parameter.

Important

You can also set the "port" parameter using the command line with the --port command line argument. The command line argument takes precedence over the configuration file value.

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Processor **rpconfig.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri	http://<Web Services IP>:<Web Services Port>	Specifies the Base URI for accessing the Interaction Recording Web Services (Web Services) API.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	Specifies the password used to access the Interaction Recording Web Services (Web Services) account. Note: The password can be overridden by the HTCC_PASSWORD environment variable.

Each Interaction Recording Web Services (Web Services) instance must have a region associated with it. Set the region parameter in the [metadata] section of the rpconfig.cfg file to match the region associated with Interaction Recording Web Services (Web Services) instance set to receive the Recording Processor's metadata.

Configuring Cross-Site Request Forgery (CSRF) Protection

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled, set the following parameter in the **[htcc]** section of the **rpconfig.cfg** file:

- **csrfp = 1**

Configuring the Connection to SpeechMiner

To configure the SpeechMiner Connection:

1. In the IVR Profile, set the recording destinations to point to the SpeechMiner interaction receiver:
 - a. Login to Genesys Administrator Extension, and navigate to **Configuration > System > Configuration Manager**.
 - b. Under **Voice Platform**, select **Voice Platform Profiles**.
 - c. Click on the IVR Profile for which you want to set the recording destination.
 - d. Select the **Recording** tab.
 - e. In the **SpeechMiner Interaction Receiver** field, enter the URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile. For example, `https://<SpeechMiner IP>/interactionreceiver`.

- f. In the **SpeechMiner Interaction Receiver Authorization Header** field, enter the authorization information (username:password) required to connect to the SpeechMiner service used by the RPS. For example, user:password.

Important

The values of these options must match the corresponding configuration options in the SpeechMiner system.

Configuring Failed Message Files

The Recording processor can backup messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner. These files are located in the **<recording processor dir>\failed** folder.

In the **rpconfig.cfg** configuration file, add the following parameter:

```
[processing]
backup_failed_metadata = 1
```

Configuring the Agent Hierarchies

Recording Processor Script uses the agent hierarchy information to set the access control information for recordings within the recording metadata. Refer to [Access Control for Genesys Interaction Recording Users](#) to configure this appropriately.

Configuring Basic Authorization

In the **rpconfig.cfg** configuration file, set the following parameters:

```
[auth]
# Basic Authentication username and password. Set username blank to disable.
username = rp_username
password = rp_password
```

Important

- The username and password must match the username and password entered in the IVR Profile. For more information about configuring the IVR Profile, see the [IVR Profile](#) section.
- The password can be overridden by the **AUTH_PASSWORD** environment variable.

Configuring After Call Work

Recording Processor can collect After Call Work (ACW) customized data from ICON.

In the **rpconfig.cfg** file, in the **[processing]** section, add the following parameters:

- **enable_acw**—Set it to 1.
- **acw_threshold_minutes**—Set it to the maximum time to wait for the customized attached data.

Important

- If Call Customized Attached Data is still not available in the ICON database after *acw_threshold_minutes*, the RPS will stop collecting customized data for this recording and write it to the database.
- If *enable_acw* is set to 0, ACW customized data will not be included.
- If disposition code is required in the metadata, you must set **enable_acw** and **acw_threshold_minutes** using Recording Processor configuration. The disposition code is part of the user data collected during ACW. For this reason, **enable_acw** must be enabled in the Recording Processor. If it is not enabled, the data will not be collected. If the disposition code must be collected from the Recording Processor, configure the following to include the disposition code for recording:
[processing] enable_acw=1
[metadata] acw_threshold_minutes=5. Where 5 is the maximum time (in minutes) to wait for the disposition code. In the ICON configuration, the **EventData** parameter in the **custom-states** section, must include char,DispositionCode and **store-event-data** must be set to conf to collect the attached data:
[custom-states] store-event-data=conf EventData=char,DispositionCode
For additional information, refer to the [ICON Deployment Guide](#).

Configuring ICON for Recording Processor

Important

When configuring Recording Processor to connect to a primary and backup ICON Database in HA mode, two separate DB Servers must be used. The DB Servers must run in an active/active pair mode.

To configure ICON, edit the **rpconfig.cfg** configuration file as follows:

1. Configure the switches:

Add a configuration option for each switch name under the **[icon_db_servers]** section. You can specify more than two ICON databases per SIP Switch configuration. For example:

```
[icon_db_servers]
SIP_Switch1: icon1
SIP_Switch2: icon2, icon2Backup
SIP_Switch3: icon3, icon4, icon5, icon6
```

Important

In the above example, **SIP_Switch3** has 4 ICON databases. The Recording Processor Script (RPS) keeps track of the ICON database instance currently used. If the current database instance becomes unavailable, RPS will attempt the operation in the next database.

The configuration option name must match the exact name of the switch as configured in the Genesys configuration. The primary and backup ICON names must be unique, but do not have to match anything in the Genesys configuration.

2. Configure the ICON Connection Settings:

- For each unique ICON specified in the first step, create a new section using the following syntax: `<ICON_ID>_db_info`, where `<ICON_ID>` corresponds to the values defined in the **[icon_db_servers]** section above.
- **dbengine** must be `mssql`, `oracle`, `db2`, or `postgres`.
- **dbserver_host** and **dbserver_port** specify the host and port information for the Genesys DB Server.
- **dbms** specifies the host where the database resides.

The following is an example using the values for **SIP_Switch1** and **SIP_Switch2** from step 1:

```
[icon1_db_info]
dbserver_host = 10.0.0.221
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2_db_info]
dbserver_host = 10.0.0.222
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon2Backup_db_info]
dbserver_host = 10.0.0.223
dbserver_port = 12201
username = iconuser_1
password = genesys
dbname = ICON_LRM_DB_1
dbms = 10.0.0.228,1433
dbengine = mssql
```

```
[icon_oracle_db_info]
dbserver_host = <host>
dbserver_port = <port>
```

```
username = <username>
password = <password>
dbname =
dbms = <database host/Oracle SID>
dbengine = oracle

[icon_postgres_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname = <database name>
dbms = <database host>
dbengine = postgres

[icon_db2_db_info]
dbserver_host = <host>
dbserver_port = <port>
username = <username>
password = <password>
dbname =
dbms = <database host>
dbengine = db2
```

Important

- For Oracle or DB2 implementations, the **dbname** parameter must be left blank or empty.
- The password can be overridden by the **<ICON_ID>_DB_INFO_PASSWORD** environment variable.

In the example above, the RPS will use the connection properties in section **[icon1_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch1. The RPS will use the connection properties in section **[icon2_db_info]** when processing recording metadata from an MCP provisioned to SIP_Switch2. In the case of SIP_Switch2, the RPS will use the connection settings in **[icon2Backup_db_info]** if the primary ICON (icon2) is unavailable when recording metadata is being processed.

Configure how to Filter Metadata from ICON

The Recording Processor supports the ability to filter specific attached data fields (based on the key name), such as attached data and After Call Work (ACW) customized data retrieved from the ICON database. This support prevents specific metadata from reaching additional GIR related components (for example, SpeechMiner).

The following two sections describe how to:

- **Filter attached data.**
- **Filter ACW.**

Important

- Verify that the following items are not removed from the filter. Removing these items may cause errors in GIR:
 - RECORD_PARTITIONS
 - RECORD_PROGRAM
 - GSIP_REC_FN
- When running SpeechMiner, you must include Workspace Web Edition (WWE) in the attached_data_filter and acw_custom_data_filter Recording Processor configuration values. For example:


```
[filter]
attached_data_filter=^ORSI|^WWE
acw_custom_data_filter=^ORSI|^WWE
```

Filter Attached Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **attached_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out attached data whose key name matches the pattern.

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

4. Add a new option called **attached_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude key names that should not be filtered out (for example, like RECORD_PARTITIONS).

```
...
[filter]
attached_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
attached_data_filter_exception = ^RECORD_PARTITIONS$      ; (Note: this is the default
value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

5. Restart the Recording Processor.

Filter ACW Related Custom Data

1. Edit the **rpconfig.cfg** file.
2. Locate the **Filter** section. If the **Filter** section does not exist, add it as follows:

```
...
[filter]
...
```

3. Add a new option called **acw_custom_data_filter** to the **Filter** section as follows. The value must be a Regex pattern used to filter out ACW whose key name matches the pattern.

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

4. Add a new option called **acw_custom_data_filter_exception** to this section as follows. The value must be a Regex pattern used to exclude ACW that should not be filtered out (for example, like **GRECORD_PARTITIONS**).

```
...
[filter]
acw_custom_data_filter = ^ORSI:      ; (Note: this is the default value when the option
is not specified.)
acw_custom_data_filter_exception = ^GRECORD_PARTITIONS$      ; (Note: this is the
default value when the option is not specified.)
...
```

For information about Regex patterns, refer to the Python's documentation found here: <https://docs.python.org/2/library/re.html>.

5. Restart the Recording Processor.

Configuring SSL for Recording Processor

To configure SSL:

Configure HTTPS on the Primary Recording Processor Server

1. Make sure pyOpenSSL is installed.
2. Create a self-signed certificate and private key for the Recording Processor host. For example, on RHEL run: `openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem`
3. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—To point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228.pem`.
4. Give the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section [Genesys Voice Platform 8.5 User's Guide](#).

5. Restart Recording Processor.

Configure the HTTPS connection to Interaction Recording Web Services (Web Services)

1. Set up HTTPS on Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). See the [Genesys Security Deployment Guide](#).
2. Get the corresponding certificate for the Interaction Recording Web Services (Web Services) server. Set the **caCertificate** option in your Interaction Recording Web Services application (see **caCertificate** if you're using a Web Services application).
3. In the **[htcc]** section of the Recording Processor configuration file, set **base_uri** parameter to use https.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file. See the <Python27 install directory>\Lib\site-packages\httplib2\cacerts.txt file for an example.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection to SpeechMiner

1. Set up HTTPS on SpeechMiner. See the [Genesys Security Deployment Guide](#).
2. Set the **disable_ssl_certificate_validation** parameter in the **[speechminer]** section of the Recording Processor configuration to a value of 1.
3. Using Genesys Administrator Extension on the Recording tab of the IVR Profile, modify the SpeechMiner Interaction Receiver field use https as the protocol in the URL.
4. In the **[client]** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA cert file. See the <Python27 install directory>\Lib\site-packages\httplib2\cacerts.txt file for an example.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS connection from the backup Recording Processor to the primary Recording Processor

1. Configure HTTPS on the primary Recording Processor.
2. Get the corresponding PEM certificate for the Web Services server.

3. In the **[processing]** section of the Recording Processor configuration file, set the **post_uri** parameter to use https as the protocol in the URL.
4. In the **client** section, set the **certs** parameter to point to the file that contains the certificate (see previous step). Or, point to an existing CA certificate file after copying the content of the new certificate into the existing CA certificate file. See the <Python27 install directory>\Lib\site-packages\httplib2\cacerts.txt file for an example.

Important

If there are multiple client connections using HTTPS, use a single CA cert file with all the certificates listed.

Configure the HTTPS on the backup Recording Processor Server

Follow the same procedure used for the Primary Recording Processor Server using a new certificate and private key for the Backup Recording Processor's server.

Configuring the IVR Profile

Using Genesys Administrator Extension, configure the following parameters on the **Recording** tab of the IVR Profile:

1. **Recording Processor URI**—The URI that the Media Control Platform (MCP) uses to post the metadata of the audio recording after the recording is complete. For example, http:// <Recording Processor Host>/api/contact-centers/<Contact Center Domain Name>/recordings/.

Important

The value for the URI must always end with a forward slash (/).

2. **SpeechMiner Interaction Receiver**—The URL that points to the SpeechMiner service responsible for accepting metadata from the RPS for this profile.
3. **SpeechMiner Interaction Receiver Authorization Header**—The authorization information required to connect to the SpeechMiner service used by the RPS. For example, <SpeechMiner Webserver Username>:<SpeechMiner Webserver Password>.

For more information, see the [Configuring GVP](#).

Configuring the Recording Processor Using Genesys Administrator Extension (Optional)

The Recording Processor uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Processor as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Processor as a "third party server" application in Genesys Administrator Extension. For more information, see the "Using the Management Layer" section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring RPS to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest RPS.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Processor Script of type Third Party Server.
4. Create a new application (for example, myRPS) in Genesys Administrator Extension using this new application template.
5. Set the Command Line parameter (for example, C:\Python27\python.exe).
6. Set the Host parameter in the application's server info to the correct Host object.
7. Set the Working Directory parameter to the <Recording Processor Install Directory>\rp directory. For example, /opt/genesys/Recording_Processor_Script_8.5/rp/.
8. Set the Command Line Arguments parameter to the appropriate values. For example, recording_process.py --config-file=/opt/genesys/Recording_Processor_Script_8.5/rp/rpconfig.cfg.
Refer to the [Starting the Recording Processor Script](#) section for additional command line parameters
9. Make sure that LCA has permission to read and write to the Recording Processor installation directory and Recording Processor log directory.
10. Save the configuration changes.
11. Ensure that the Configuration Server parameters in the Recording Processor configuration file are set appropriately. Refer to **Configure the Configuration Server Connection** tab on this page.

Important

The Recording Processor does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

For more information about the RPS options, see [Genesys Interaction Recording Options Reference](#).

Starting the Recording Processor Script

To launch the RPS, run the following command from the <Recording Processor Install Directory>:

```
<python executable> recording_process.py --config-file=rpconfig.cfg
```

Use the following command line when you want to run multiple instances of RPS on the same machine:

```
<python executable> recording_process.py --config-file=rpconfig.cfg --id=1 --port=8889
```

For each RPS instance, assign a unique id (--id parameter) and port number (--port).

Important

- --port defines the server port opened by the RPS process.
- --id represents the suffix of the:
 - application_name in the configuration file. For example, if application_name is defined in the configuration file as **RecordingProcessorScript** and --id 2 is specified in the command line, then the application object named **RecordingProcessorScript_2** will be used to start the program.
 - log files
 - metadata json files created in the failed folder
 - database file created by the process

By default the RPS log file is stored in the working directory. This can be changed by specifying a preexisting folder in the logfile_path parameter in the log file section of the configuration file. For example:

```
logfile_path = C:\logs\recordingProcessor
```

Deploying Voice Processor

Genesys Interaction Recording (GIR) needs Voice Processor to process recording metadata from Media Control Platform (MCP), combine this metadata with data collected from Genesys Info Mart (GIM), and forward the result to Interaction Recording Web Services (RWS) and SpeechMiner Interaction Receiver (SM IR). During this process, recordings from multiple call legs are combined into a single interaction.

Important

If you are not using Voice Processor, Recording Processor Script (RPS) can be used. However, for new deployments, Genesys recommends using Voice Processor instead of RPS.

This topic contains the following sub-topics:

- [Prerequisites](#)
- [Preparing your Docker environment](#)
- [Configuring Voice Processor](#)
- [Deploying Voice Processor to Docker](#)
- [Miscellaneous Docker tips](#)
- [Migrating from RPS to Voice Processor](#)
- [Monitoring and troubleshooting information](#)

Comparison of Voice Processor and RPS

Voice Processor is a multi-threaded microservice based on the Node.js platform and it replaces the Python-based Recording Processor Script (RPS). The key advantage of Voice Processor is that since Node.js is a multi-threaded platform, a single Voice Processor instance can handle an incoming recording post load equivalent to 30-40 instances of RPS (20+ recordings per second). Therefore, a single instance should be sufficient for most customers. For customers with extremely high volumes, or who require redundancy, the Voice Processor can be run behind a load balancer similar to the existing RPS deployments.

Another benefit is that, in the event of outages that prevent posting of recordings to RWS and SpeechMiner Interaction Receiver, the Voice Processor automatically retries these posts for up to 40 days. As a result, you do not have to manually recover recordings if you resolve the downstream outage within that period.

The Voice Processor retrieves additional metadata from Genesys Info Mart (GIM) instead of Interaction Concentrator Database (ICON). The format and contents of the metadata posted by the Voice Processor to RWS and SM IR do not differ significantly from the format and contents posted by

RPS. However, there are some differences that may impact third-party integrations that download recording metadata from RWS, Recording Backup Service (RCBS), or from SpeechMiner. You must consider the following differences in the format and contents of the metadata:

- Name of the **eventID** property in the **eventData** list is different.
- Metadata that is meant to be internal-use only is not posted by the Voice Processor.
- As Genesys Info Mart is a data warehouse that is updated on a periodic basis through ICON data, the arrival of recordings in SpeechMiner is slightly delayed when compared to RPS.

Prerequisites

- Docker version 17.12.1-ce or higher running on a x86_64 Linux host.
- Ansible 2.6 or higher installed on the Docker host and the deployment is using Red Hat Enterprise Linux 7.
- Ansible 2.13 or higher installed on the Docker host and the deployment is using Red Hat Enterprise Linux 8.
- PostgreSQL 12.11 or higher.
- Genesys Info Mart 8.5 or higher installed on Microsoft SQL Server or PostgreSQL. For information on the system requirements for GIM, see [Genesys Info Mart Requirements](#).
- Interaction Recording Web Services (RWS) 8.5.201.90 or higher.
- SpeechMiner 8.5 or higher if you are using SpeechMiner. We recommend that you install Speechminer before deploying Voice Processor.

We recommend that you have the following details before proceeding with deployment:

- Host name, port, database name for Genesys Info Mart, and user (read-only) credentials.
- Host name and port for the Interaction Recording Web Services (RWS), and Operation Admin (ops) credentials.
- Configuration Manager credentials for an account with access to the IVR Profile.
- Host name, port, and credentials needed to post to SpeechMiner Interaction Receiver. These details are required only for new installations.

Preparing your Docker environment

Extracting installation files

You can download the Docker image from the Genesys customer portal. The Docker image is a .tar file that contains the installation and configuration files required to set up and run the Voice Processor. Extract and copy the files from the image using the following steps:

1. Load the Docker image.

```
zcat <.tar file> | docker load
```

2. View the list of Docker images and make a note of the newly loaded image.

```
docker image ls
```

3. Add a custom tag to the docker image for your reference.

```
docker tag <image ID> <tag>
```

4. Copy the files from the image.

```
id=$(docker run --rm -dt <image> cat) && docker cp $id:/rps/compose . && docker stop $id
```

Now you have the sample configuration files in **./compose/defaults**, an Ansible playbook in **./compose** to help you set up and run the Voice Processor, and an SQL file for database setup. You will need these files for installing and configuring the Voice Processor.

Tip

You can refer to the custom tag of your docker image when setting parameters in the configuration files such as **settings-override.yml**, **secrets.yml**, and **docker-config.yml**.

Setting up Docker

Docker on the host must be running in swarm mode, but a multi-host swarm is not required. A new network inside Docker is used for the deployment. Docker swarm and its network can be set up this way:

```
docker swarm init
docker network create gir_vp --driver overlay --scope swarm
```

If the network that you created collides with an existing network in your environment, you can define the network's IP range like this:

```
docker network create --subnet 10.99.99.0/24 --gateway 10.99.99.1 --scope swarm gir_vp
```

Docker configs and secrets

Configs and secrets are Docker objects available in Docker swarm mode for storing run-time container configuration files and are mounted inside the container at run-time. The key difference between the two is that secrets are encrypted in Docker when at rest.

Docker logs

Docker container logs are typically stored under **/var/lib/docker/containers**, but container logs can be accessed simply by `docker logs <container name>`. Logs are rotated based on run-time configurations.

Configuring Voice Processor

This section contains the following sub-sections:

- [PostgreSQL database configuration](#)
- [Service level configuration](#)
- [Genesys Voice Platform profile configuration](#)
- [Tenant level configuration](#)
- [GIM DB ETL configuration](#)

PostgreSQL database configuration

The Voice Processor requires a service-specific database that tracks work in progress items. This database runs on a PostgreSQL server. Set up the database using the following steps:

1. Create a database in your PostgreSQL server for the Voice Processor.
2. Create a PostgreSQL user and grant all privileges to the database that you created in the previous step.
3. Assign a password to the user that does not contain a backward slash (\) or quotation marks, as they might cause issues later.
4. Make a note of the database name, user name, and password — they are needed when configuring the Voice Processor in later steps.
5. Confirm that the **standard_conforming_strings** parameter of the PostgreSQL server is set to on (default).

Important

- GIR Voice Processor must have a separate PostgreSQL DB from Config Server since the Voice Processor PostgreSQL DB requires the `standard_conforming_strings` setting to be on and the Config Server PostgreSQL DB requires the `standard_conforming_strings` setting to be off.
- Voice Processor supports connections to PostgreSQL DB when `password_encryption` is set to `md5` or `scram-sha-256` in the **postgresql.conf** file.
- When using a PostgreSQL DB for Genesys Info Mart, if `password_encryption` is set to `scram-sha-256` in the **postgresql.conf** file, the Genesys Info Mart version must be 8.5.016.04 or higher.

6. Run the provided script, `create_node_rps_tables_v2.sql`, against this new database to provision it.

Important

To avoid possible conflicts with their settings requirements, Genesys recommends not hosting the Voice Processor and Configuration Server databases on the same PostgreSQL instance.

Service level configuration

You can follow the instructions provided with the configuration files available in the default directory. You can copy the provided configuration files and make changes to your copies. We recommend that you use a version control repository to store your configurations. Add the PostgreSQL database, user name, and password to the **nodeRpsDb** setting in your copies of **settings-override.yml** and **secrets.yml**.

Voice Processor database settings

To enable TLS connection to the Voice Processor database, set the **ssl** parameter to `true` and configure the **trustedCA** parameter under **nodeRpsDb** in **settings-override.yml**.

```
nodeRpsDb:
  database: <database name>
  host: <db server hostname>
  port: <db port>
  user: <db user>
  ssl: < true / false >
  trustedCA: false / true / "<path to root certificate>"
```

The **ssl** parameter is optional and its default value is `false`. When you set it to `true`, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to `true`, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is `false`.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is `true`.
- Authenticate the server certificate against the specified root authorities. Set **vpdb_ca_cert** in your copy of **docker-config.yml** with the `<path to root certificate>` value.

Voice Processor HTTPS settings

The **rwsBaseUri** setting in **settings-override.yml** supports HTTPS. For example:

```
https://<RWS hostname>:<RWS port>
```

To use HTTPS on the Voice Processor service API, set **https** to `true` in your copy of **docker-config.yml**. You must provide the server private key, public key, and path to the files.

```
https: true
tls:
  privkey: <path to the private key file>
  pubkey: <path to the public key file>
```

MCP post basic authentication

Add the following lines to the **settings-override.yml** file to enable basic authentication for the endpoint used by the MCP to post recording metadata:

```
authUsername: "<basic auth username>"
authPassword: "<basic auth password>"
```

If you add these options, you must also configure the Voice Platform profile option, **recording client.callrec_authorization**, in the **[gvp.service-parameters]** section to match these credentials. As basic authentication involves sending the credentials in plain text format, we strongly recommend that you use TLS for maximum security. Note that the other Voice Processor endpoints are not authenticated. Therefore, you must install the Voice Processor behind a firewall or API gateway to restrict access. You can obtain a summary of endpoints exposed by the Voice Processor service by accessing:

```
http://<GIR VP hostname>:<port>/apidoc
```

Setting the Voice Processor Docker image

Add the following line to the **docker-config.yml** file to specify the Voice Processor docker image that was imported:

```
image: <image ID>:<tag>
```

To find the values for <image ID>:<tag>, use the `docker images` command. An example is given below:

```
[root@CENTOS7-CloudVM defaults]# docker images
REPOSITORY              TAG                IMAGE ID
CREATED                 SIZE
voice-processor_v9000025 latest            7320945e8b25
21 months ago          979MB
pureengage-docker-production.jfrog.io/gir_rp_nodejs 9.0.000.04.023    7320945e8b25
21 months ago          979MB
[root@CENTOS7-CloudVM defaults]#
```

Genesys Voice Platform profile configuration

Use HTTPS protocol in the Voice Processor URL when HTTPS is enabled in the Voice Processor service API.

```
recordingclient.callrec_dest = fixed,https://<VP hostname>:<VP port>/api/contact-
centers/<CCID>/recordings/
```

Use HTTPS protocol in the SpeechMiner Interaction Receiver URL when HTTPS is enabled on the SpeechMiner Interaction Receiver. When using HTTPS for the SpeechMiner URL, by default, the Voice Processor does not validate SpeechMiner server certificate. You can set **sm_ca_cert** in your copy of **docker-config.yml** with the <path to root certificate> value to authenticate the server certificate against the specified root authorities.

```
recordingclient.rp.speechminer_uri: fixed,https://<Speechminer backend
hostname>/interactionreceiver/
```

Tenant level configuration

As the Voice Processor is designed to support Genesys cloud multi-tenancy model, settings that may vary from tenant to tenant are stored in an RWS group settings called **rps-provisioning**:

Important

You need an Ops Admin user account to access these settings. For more information on how to update settings in RWS, see [Settings API](#).

You must specify the Ops Admin user name and password in your copy of **secrets.yml**. The tenant level configuration values are set to the RWS group settings **rps-provisioning** using HTTP POST. For example:

```
curl -u <Ops admin user>:<password> -X POST -H "Content-Type: application/json"
<rwsBaseUri>/api/v2/ops/contact-centers/<ccid>/settings/rps-provisioning -d @rps-
settings.json
```

Where **rps-settings.json** contains settings like: eventDataFilters, gimDb, rwsPostRecBaseUri and others.

To confirm the Voice Processor per tenant settings, use HTTP GET. For example:

```
curl -u <Ops admin user>:<password> -X GET "<rwsBaseUri>/api/v2/ops/contact-
centers/<ccid>/settings/rps-provisioning?location=*&ignoreParentLocations=false"
```

GIM database

You must provide information needed to access the tenant's GIM database. To enable TLS connection to the GIM database, set the **ssl** parameter to true and configure the **trustedCA** parameter under GIM database settings in tenant level configuration.

```
{
  "name": "gimDb",
  "value": {
    "primary": {
      "host": "<GIM server hostname>",
      "port": "<GIM server port (default 5432 for Postgres, 1433 for MS SQL)>",
      "user": "< DB user name >",
      "database": "<database name>",
      "password": "<DB user password>",
      "dbType": "<postgres or mssql, default postgres>",
      "ssl": < true / false >,
      "trustedCA": false / true / "<path to root certificate>",
    },
    "backup": {
      < same settings as for primary >
    }
  }
}
```

The **ssl** parameter is optional and its default value is false. When you set it to true, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to true, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is false.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is true
- Authenticate the server certificate against the specified root authorities by performing the following steps:
 1. Set **gim_ca_cert** in your copy of **docker-config.yml** with the <path to root certificate> value.
 2. Set **trustedCA** to /rps/rpsdata/gimCA in GIM database settings to be posted to tenant level configuration.

The **backup** parameter is optional. You can omit it if there is only one GIM database available.

RWS posting

You must specify the RWS instance to which recordings are posted. As this is a region-based setting, multi-regional deployments can ensure that recording data stays within the jurisdictional boundaries. The Voice Processor instance selects the location identified through the **nodePath** of the RWS server from which the setting is retrieved or the nearest matching parent. The **backup** parameter is optional. The URL supports HTTPS. When using HTTPS for the RWS URL, by default, the Voice Processor does not validate RWS server certificate. You can set **rws_ca_cert** in your copy of **docker-config.yml** with the <path to root certificate> value to authenticate the server certificate against the specified root authorities.

For example, the following setting applies to all Voice Processor instances:

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

The following setting would override the above global setting for Voice Processor instances that retrieved the setting from an RWS node with **nodePath** /US or /US/* :

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/US",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

Event filtering

You can use filters to remove unwanted data from the recording metadata. The event filtering settings are similar to RPS except the mechanism of how the default filters are disabled.

```
{
  "name": "eventDataFilters",
  "value": {
    "attachedDataFilter": "regex for new attached data filter",
    "attachedDataFilterException": "regex for new attached data filter exception",
  }
}
```

```

    "acwCustomDataFilter": "regexp for new ACW data filter",
    "acwCustomDataFilterException": "regexp for new ACW data filter exception"

    -- or, to disable the default filters or filter exceptions --

    "disableAttachedDataFilter": true,
    "disableAttachedDataFilterException": true,
    "disableAcwCustomDataFilter": true,
    "disableAcwCustomDataFilterException": true
  }
}

```

The default filters are:

- **attachedDataFilter:** ^ORSI:|^WWE|^PegAG
- **attachedDataFilterException:** ^(GRECORDER_(PARTITIONS|PROGRAM)|GSR_STATE|GSIP_REC_FN)\$
- **acwCustomDataFilter:** ^ORSI:|^WWE|^PegAG
- **acwCustomDataFilterException:** ^(GRECORDER_(PARTITIONS|PROGRAM)|GSR_STATE|GSIP_REC_FN)\$

Complete after-call work (ACW) threshold

The ACW threshold indicates how long the Voice Processor waits, in minutes, following the end of an interaction to update custom data. Custom data entered by agents after this interval is not added to recording metadata. The default value is zero.

```

{
  "name": "acwThresholdMinutes",
  "value": <ACW wait interval in minutes>
}

```

GIM DB ETL configuration

You must configure the GIM ETL application properly to ensure recording metadata is posted from the Voice Processor to SpeechMiner in a timely manner.

The **etl-start-time**, **etl-end-time**, and **etl-timezone** options in the **[schedule]** section are used to configure a daily maintenance period during which population of GIM data is paused for maintenance purpose. New recordings posted to the Voice Processor during this period are not processed and they are held temporarily in a database until the maintenance period finishes and the relevant GIM data becomes available. You must configure the **maintain-start-time** option such that the GIM ETL maintenance job begins and completes during the maintenance period.

The **etl-frequency** option in the **[schedule]** section is used to specify the cycle time of the GIM ETL jobs that populate the recording metadata used by the Voice Processor. We recommend that you use the default value of one minute. Note that any time longer than 3 minutes may cause subsequent delays in recording posts. If a longer **etl-frequency** setting is used, then the value of the Voice Processor service setting, **rpsInitialInteractionTimeout**, should be increased accordingly.

The **user-event-data-timeout** option in the **[gim-etl]** section is used to ensure that custom attached data entered during after-call work is captured. You can increase the default value of one hour if your agents will spend more than a few minutes in after-call work.

Important

Consult Genesys before setting non-default values for the following options.

The **max-call-duration**, **merge-failed-is-link-timeout**, and **extract-data-stuck-threshold** options in the **[gim-etl]** section must be configured properly to ensure completeness of the call metadata recorded in GIM. For more information on these options, see [Operations-Related Options for Genesys Info Mart](#).

Deploying and Starting Voice Processor

Deploy Voice Processor to the newly configured Docker swarm using Ansible, referencing your copies of the default configuration files. This step also starts Voice Processor.

Before deploying, the **settings-override.yml** and **secrets.yml** files (or the yaml files you have designated to provide these settings) have several mandatory parameters that must be configured, as described below.

In the **settings-override.yml** file, the following parameters are required:

- **rwsBaseUri** – Specifies the address of the RWS cluster that will provide Voice Processor with contact center settings, including tenant-specific configurations such as Genesys Info Mart (GIM) database information and ACW Wait Time. Example: `http://some-rws-host.com:8090`
- **region** – Controls the region section in the metadata POSTed to RWS. This must match the **crRegion** setting of the RWS cluster to which recordings are posted. Example: `usa`
- **nodeRpsDb** – This section specifies the Voice Processor Persistence Database, which is required for storing recording metadata while Voice Processor is processing them.
 - **database** – Database on the host that will hold recording metadata. Example: `noderrpsdb`
 - **host** – Host of the Persistence Database. Example: `noderrpsdb.com`
 - **port** – Port that the Persistence Database is listening on. Example: `5432`

In the **secrets.yml** file, the following parameters are required:

- **nodeRpsDb** – This section specifies the credentials to the Voice Processor Persistence Database.
 - **user** – The user name to connect to the Persistence Database.
 - **password** – The password to connect to the Persistence Database.
- **rwsUserName** – The user name for authenticating with RWS.
- **rwsPassword** – The password for authenticating with RWS.

For an example of how the yaml files should be structured, you can refer to the default yaml files that were included with Voice Processor. These files are located at **<INSTALL_DIR>/defaults/**, where **<INSTALL_DIR>** is the location where you extracted the installation files to during the [Preparing](#)

your Docker environment step.

After configuring the default configuration files, deploy and start Voice Processor:

```
ansible-playbook \
  -e docker_config=<defaults directory of Your Docker Configuration Yaml (e.g. docker-
  config.yaml)> \
  -e logger_config=<defaults directory of Your Logger Configuration Yaml (e.g. logger-
  config.yaml)> \
  -e settings_override=<defaults directory of Your Voice Processor Configuration Yaml
  (e.g. settings-override.yaml)> \
  -e secrets=<defaults directory of Your Voice Processor Secrets Yaml (e.g.
  secrets.yaml)> \
  gir-vp-playbook.yaml
```

Important

If the above options are not specified, then the .yaml files in **./compose/defaults** will be used.

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }'
girvp.company.com/api/active-version
```

Validating

1. Place a call to an agent or a test agent that is configured for recording.
2. Verify that the call arrives at the SpeechMiner UI. It should take 5 to 15 minutes depending on your configured ACW wait setting.
3. Assuming that live traffic is not recorded, you can use the health check endpoint `<domain:port>/api/status?verbose=1`. The items `recordingsInProgress` or the MCP Post operational status can be helpful in determining whether or not the recording is arriving at the Voice Processor. This also helps you to isolate GVP configuration problems from problems with the Voice Processor service. If a load balancer is used, the node serving the health check may not be the one that handled the recording. Therefore, several health checks may be required to cover the whole cluster.

Upgrading

Docker object configurations and secrets cannot be upgraded. We recommend that you remove the stack, update the required configurations, and redeploy.

```
docker stack rm <gir_vp>
ansible-playbook \
```

```
-e docker_config=mydocker.yml \  
-e logger_config=mylogger.yml \  
-e settings_override=mysettings.yml \  
-e secrets=mysecrets.yml \  
gir-vp-playbook.yml
```

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }'  
girvp.company.com/api/active-version
```

Miscellaneous Docker tips

- To view network details:

```
docker network inspect <network_name>
```

- To view a list of your swarm stacks:

```
docker stack ls
```

- To view a list containers in your stack:

```
docker stack ps <gir_vp>
```

- To view the container logs:

```
docker logs <container name>
```

- To remove everything to start again:

```
docker stack rm gir_vp  
docker network rm <network_name>  
docker swarm leave --force
```

Migrating from RPS to Voice Processor

This section explains how to migrate from an existing RPS deployment to Voice Processor.

Prerequisites

- Voice Processor is fully deployed

- The following Voice Processor dependent components are working as expected:
 - Interaction Recording Web Services (RWS)
 - Genesys Info Mart Database
 - SpeechMiner Interaction Receiver
 - Voice Processor Database

Migrating procedure

You can migrate from RPS to Voice Processor by changing the IVR profile, even if a Load Balancer is being used for RPS.

You must configure the **Recording Processor URI** parameter in the **Recording** tab of the IVR profile using Genesys Administrator Extension (GAX). This URI is used by Media Control Platform (MCP) to post metadata of the audio recording after the recording is complete. You must change this parameter to ensure that MCP posts metadata to the Voice Processor instead of RPS. For example:

```
http://<Voice Processor Host>:<Voice Processor Port>/api/contact-centers/<Contact Center Domain Name>/recordings/
```

The value for the URI must always end with a forward slash (/). For more information, see [Deploying Genesys Voice Platform for GIR](#).

Important

We recommend that you save the original value that is needed if rollback becomes necessary.

Validation

1. Place a test call.
2. After 15 to 20 minutes, check the SpeechMiner UI for the test call recording that should appear for the test agent.
3. RPS should no longer receive any call data.

Rollback

Restore the original value of the **Recording Processor URI** parameter in the IVR profile.

Shutting down RPS

Before shutting down RPS, recover any lost recordings after the RPS has processed existing calls. For more information, see [Recovering Metadata for SpeechMiner](#). After ensuring that the Voice Processor is processing data as expected, shut RPS down. If there are any GIR ICONs used by RPS, shut them down as well.

Monitoring and troubleshooting information

The Voice Processor provides detailed health and performance information on the endpoint `<domain:port>/api/status`.

The following optional query parameters allow you to request specific health reports:

- `?verbose=1` provides a summary for each tenant and service.
- `?ccid=<HTCC ID>` provides a detailed report for a single tenant.
- `?service=<service name>` provides a detailed report for a single service. The following services are available for querying:
 - `persistence` provides information about the health and performance of the Voice Processor database.
 - `ccSettings` provides a status on the connection to RWS and the validity of the RWS settings.
 - `gim` provides health and performance information of the GIM database.
 - `rws` provides a status on RWS in the context of posting recording metadata.
 - `sm` provides health and performance information on data posted to SpeechMiner Interaction Receiver.
 - `schedRecovery` provides health and performance information on the internal scheduled recovery service which retries failed tasks periodically.
 - `mcpPosts` provides health and performance information on handling of incoming posts from MCP.

For example, if posts are not reaching SpeechMiner, a query to `/api/status?verbose=1` should provide sufficient information to isolate the problem. Additionally, you can provide a snapshot of the output from this query when you contact Genesys Customer Care for assistance.

GIR Voice Processor deployment using Podman

This deployment is applicable for Voice Processor 9.0.000.39 or higher for installing GIR VP using Podman.

Currently GIR Voice Processor is deployed using Docker swarm in Premise and through Elastic Container Service in AWS. Due to known issues with Docker Swarm and Docker in RHEL 8, GIR Voice Processor is moving towards Podman for deploying Voice Processor.

Limitations

- Currently there are no alternatives for Docker swarm features with Podman. We may have to use a load balancer for load balancing with Podman.
- Network mode *overlay* is not supported in Podman, we will use *host* network mode.

Prerequisites

- Confirm Podman 4.9 or higher.
- Confirm Podman-compose 1.2.0 or higher.
 - `podman --version`
- PostgreSQL - one of the following:
 - From Voice Processor 9.0.000.40 (or higher), install PostgreSQL 16.4.
 - From Voice Processor 9.0.000.39 (or lower), install PostgreSQL 12.11.
- Genesys Info Mart 8.5 or higher installed on Microsoft SQL Server or PostgreSQL. For information on the system requirements for GIM, see [Genesys Info Mart Requirements](#).
- Interaction Recording Web Services (RWS) 8.5.205.32 or higher.
- SpeechMiner 8.5 or higher if you are using SpeechMiner. We recommend that you install Speechminer before deploying Voice Processor.

We recommend that you have the following details before proceeding with deployment:

- Host name, port, database name for Genesys Info Mart, and user (read-only) credentials.
- Host name and port for the Interaction Recording Web Services (RWS), and Operation Admin (ops) credentials.
- Configuration Manager credentials for an account with access to the IVR Profile.

- Host name, port, and credentials needed to post to SpeechMiner Interaction Receiver. These details are required only for new installations.

Preparing Podman environment

Extracting installation files

You can download the GIR VP image from the Genesys customer portal. The GIR VP image is a .tar file that contains the installation and configuration files required to set up and run the Voice Processor. Extract and copy the files from the image using the following steps:

1. Load the GIR VP image.
`zcat <.tar file> | podman load`
2. View the list of container images and make a note of the newly loaded image.
`podman image ls`
3. Add a custom tag to the image for your reference.
`podman tag <image ID> <tag>`
4. Copy the files from the image.
`id=$(podman run --rm -dt <image> cat) && podman cp $id:/rps/compose . && podman stop $id`

In the above command, `--rm` option will delete the image file after creating the Podman container.

Now you have the sample configuration files in **./compose/defaults**, an Ansible playbook in **./compose** to help you set up and run the Voice Processor, and an SQL file for database setup. You will need these files for installing and configuring the Voice Processor.

Podman logs

Podman logs location can be find using:

```
podman inspect --format='{{.HostConfig.LogConfig.Path}}' <container-id>
```

Podman container logs can be accessed simply by `podman logs <container name>`.

Configuring Voice Processor

This section contains the following sub-sections:

- [PostgreSQL database configuration](#)
- [Service level configuration Documentation:CR:Solution:VP:8.5.2](#)
- [Genesys Voice Platform profile configuration Documentation:CR:Solution:VP:8.5.2](#)
- [Tenant level configuration Documentation:CR:Solution:VP:8.5.2](#)
- [GIM DB ETL configuration Documentation:CR:Solution:VP:8.5.2](#)

PostgreSQL database configuration

The Voice Processor requires a service-specific database that tracks work in progress items. This database runs on a PostgreSQL server. Set up the database using the following steps:

1. Create a database in your PostgreSQL server for the Voice Processor.
2. Create a PostgreSQL user and grant all privileges to the database that you created in the previous step.
3. Assign a password to the user that does not contain a backward slash (\) or quotation marks, as they might cause issues later.
4. Make a note of the database name, user name, and password — they are needed when configuring the Voice Processor in later steps.
5. Confirm that the **standard_conforming_strings** parameter of the PostgreSQL server is set to on (default).

Important

- GIR Voice Processor must have a separate PostgreSQL DB from Config Server since the Voice Processor PostgreSQL DB requires the `standard_conforming_strings` setting to be on and the Config Server PostgreSQL DB requires the `standard_conforming_strings` setting to be off.
- Voice Processor supports connections to PostgreSQL DB when `password_encryption` is set to md5 or scram-sha-256 in the **postgresql.conf** file.
- When using a PostgreSQL DB for Genesys Info Mart, if `password_encryption` is set to scram-sha-256 in the **postgresql.conf** file, the Genesys Info Mart version must be 8.5.016.04 or higher.

6. Run the provided script, `create_node_rps_tables_v2.sql`, against this new database to provision it.

Important

To avoid possible conflicts with their settings requirements, Genesys recommends not hosting the Voice Processor and Configuration Server databases on the same PostgreSQL instance.

Service level configuration

You can follow the instructions provided with the configuration files available in the default directory. You can copy the provided configuration files and make changes to your copies. We recommend that you use a version control repository to store your configurations. Add the PostgreSQL database, user name, and password to the **nodeRpsDb** setting in your copy of **settings-override.json**.

Voice Processor database settings

To enable TLS connection to the Voice Processor database, set the **ssl** parameter to **true** and configure the **trustedCA** parameter under **nodeRpsDb** in **settings-override.json**.

```
nodeRpsDb:
  database: <database name>
  host: <db server hostname>
  port: <db port>
  user: <db user>
  ssl: < true / false >
  trustedCA: false / true / "<path to root certificate>"
```

The **ssl** parameter is optional and its default value is **false**. When you set it to **true**, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to **true**, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is **false**.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is **true**.
- Authenticate the server certificate against the specified root authorities. Set **vpdb_ca_cert** in your copy of **settings-override.json** with the **<path to root certificate>** value.
- While enabling TLS on the Voice Processor service, please validate that Podman has access to certificate files.

Voice Processor HTTPS settings

The **rwsBaseUri** setting in **settings-override.json** supports HTTPS. For example:

```
https://<RWS hostname>:<RWS port>
```

To use HTTPS on the Voice Processor service API, set **https** to **true** in your copy of **settings-override.json**. You must provide the server private key, public key, and path to the files.

```
https: true
tls:
  privkey: <path to the private key file>
  pubkey: <path to the public key file>
```

MCP post basic authentication

Add the following lines to the **settings-override.json** file to enable basic authentication for the endpoint used by the MCP to post recording metadata:

```
authUsername: "<basic auth username>"
authPassword: "<basic auth password>"
```

If you add these options, you must also configure the Voice Platform profile option, **recording client.callrec_authorization**, in the **[gvp.service-parameters]** section to match these credentials. As basic authentication involves sending the credentials in plain text format, we strongly

recommend that you use TLS for maximum security. Note that the other Voice Processor endpoints are not authenticated. Therefore, you must install the Voice Processor behind a firewall or API gateway to restrict access. You can obtain a summary of endpoints exposed by the Voice Processor service by accessing: `http://<GIR VP hostname>:<port>/apidoc`

Setting the GR Voice Processor image

To find the values for `<image ID>:<tag>`, use the `podman image ls` command.

Genesys Voice Platform profile configuration

Use HTTPS protocol in the Voice Processor URL when HTTPS is enabled in the Voice Processor service API.

```
recordingclient.callrec_dest = fixed,https://<VP hostname>:<VP port>/api/contact-centers/<CCID>/recordings/
```

Use HTTPS protocol in the SpeechMiner Interaction Receiver URL when HTTPS is enabled on the SpeechMiner Interaction Receiver. When using HTTPS for the SpeechMiner URL, by default, the Voice Processor does not validate SpeechMiner server certificate. You can set **sm_ca_cert** in your copy of **settings-override.json** with the `<path to root certificate>` value to authenticate the server certificate against the specified root authorities.

```
recordingclient.rp.speechminer_uri: fixed,https://<Speechminer backend hostname>/interactionreceiver/
```

Tenant level configuration

As the Voice Processor is designed to support Genesys cloud multi-tenancy model, settings that may vary from tenant to tenant are stored in an RWS group settings called **rps-provisioning**:

Important

You need an Ops Admin user account to access these settings. For more information on how to update settings in RWS, see [Settings API](#).

You must specify the Ops Admin user name and password in your copy of **settings-override.json**. The tenant level configuration values are set to the RWS group settings **rps-provisioning** using HTTP POST. For example:

```
curl -u <Ops admin user>:<password> -X POST -H "Content-Type: application/json" <rwsBaseUri>/api/v2/ops/contact-centers/<ccid>/settings/rps-provisioning -d @rps-settings.json
```

Where **rps-settings.json** contains settings like: `eventDataFilters`, `gimDb`, `rwsPostRecBaseUri` and others.

To confirm the Voice Processor per tenant settings, use HTTP GET. For example:

```
curl -u <Ops admin user>:<password> -X GET "<rwsBaseUri>/api/v2/ops/contact-centers/<ccid>/settings/rps-provisioning?location=*&ignoreParentLocations=false"
```

GIM database

You must provide information needed to access the tenant's GIM database. To enable TLS connection to the GIM database, set the **ssl** parameter to true and configure the **trustedCA** parameter under GIM database settings in tenant level configuration.

```
{
  "name": "gimDb",
  "value": {
    "primary": {
      "host": "<GIM server hostname>",
      "port": "<GIM server port (default 5432 for Postgres, 1433 for MS SQL)>",
      "user": "<DB user name >",
      "database": "<database name>",
      "password": "<DB user password>",
      "dbType": "<postgres or mssql, default postgres>",
      "ssl": "< true / false >",
      "trustedCA": false / true / "<path to root certificate>",
    },
    "backup": {
      < same settings as for primary >
    }
  }
}
```

The **ssl** parameter is optional and its default value is false. When you set it to true, the Voice Processor establishes a secure connection to the GIM database using TLS 1.2. Additionally, when the **ssl** parameter is set to true, the **trustedCA** parameter can be interpreted as follows:

- Do not authenticate the server certificate when the **trustedCA** value is false.
- Authenticate the server certificate against the system's root authorities when the **trustedCA** value is true
- Authenticate the server certificate against the specified root authorities by performing the following steps:
 1. Set **gim_ca_cert** in your copy of **settings-override.json** with the <path to root certificate> value.
 2. Set **trustedCA** to /rps/rpsdata/gimCA in GIM database settings to be posted to tenant level configuration.

The **backup** parameter is optional. You can omit it if there is only one GIM database available.

RWS posting

You must specify the RWS instance to which recordings are posted. As this is a region-based setting, multi-regional deployments can ensure that recording data stays within the jurisdictional boundaries. The Voice Processor instance selects the location identified through the nodePath of the RWS server from which the setting is retrieved or the nearest matching parent. The **backup** parameter is optional. The URL supports HTTPS. When using HTTPS for the RWS URL, by default, the Voice Processor does not validate RWS server certificate. You can set **rws_ca_cert** in your copy of **settings-override.json** with the <path to root certificate> value to authenticate the server certificate against the specified root authorities.

For example, the following setting applies to all Voice Processor instances:

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

The following setting would override the above global setting for Voice Processor instances that retrieved the setting from an RWS node with nodePath /US or /US/* :

```
{
  "name": "rwsPostRecBaseUri",
  "location": "/US",
  "value": {
    "primary": "http://<hostname>:<port>{/<optional routing prefix>}",
    "backup": "http://<hostname>:<port>{/<optional routing prefix>}"
  }
}
```

Event filtering

You can use filters to remove unwanted data from the recording metadata. The event filtering settings are similar to RPS except the mechanism of how the default filters are disabled.

```
{
  "name": "eventDataFilters",
  "value": {
    "attachedDataFilter": "regex for new attached data filter",
    "attachedDataFilterException": "regex for new attached data filter exception",
    "acwCustomDataFilter": "regex for new ACW data filter",
    "acwCustomDataFilterException": "regex for new ACW data filter exception"

    -- or, to disable the default filters or filter exceptions --

    "disableAttachedDataFilter": true,
    "disableAttachedDataFilterException": true,
    "disableAcwCustomDataFilter": true,
    "disableAcwCustomDataFilterException": true
  }
}
```

The default filters are:

- **attachedDataFilter:** ^ORSI:|^WWE|^PegAG
- **attachedDataFilterException:** ^(GRECORDER_(PARTITIONS|PROGRAM)|GSRS_STATE|GSIP_REC_FN)\$
- **acwCustomDataFilter:** ^ORSI:|^WWE|^PegAG
- **acwCustomDataFilterException:** ^(GRECORDER_(PARTITIONS|PROGRAM)|GSRS_STATE|GSIP_REC_FN)\$

Complete after-call work (ACW) threshold

The ACW threshold indicates how long the Voice Processor waits, in minutes, following the end of an interaction to update custom data. Custom data entered by agents after this interval is not added to

recording metadata. The default value is zero.

```
{
  "name": "acwThresholdMinutes",
  "value": <ACW wait interval in minutes>
}
```

GIM DB ETL configuration

You must configure the GIM ETL application properly to ensure recording metadata is posted from the Voice Processor to SpeechMiner in a timely manner.

The **etl-start-time**, **etl-end-time**, and **etl-timezone** options in the **[schedule]** section are used to configure a daily maintenance period during which population of GIM data is paused for maintenance purpose. New recordings posted to the Voice Processor during this period are not processed and they are held temporarily in a database until the maintenance period finishes and the relevant GIM data becomes available. You must configure the **maintain-start-time** option such that the GIM ETL maintenance job begins and completes during the maintenance period.

The **etl-frequency** option in the **[schedule]** section is used to specify the cycle time of the GIM ETL jobs that populate the recording metadata used by the Voice Processor. We recommend that you use the default value of one minute. Note that any time longer than 3 minutes may cause subsequent delays in recording posts. If a longer **etl-frequency** setting is used, then the value of the Voice Processor service setting, **rpsInitialInteractionTimeout**, should be increased accordingly.

The **user-event-data-timeout** option in the **[gim-etl]** section is used to ensure that custom attached data entered during after-call work is captured. You can increase the default value of one hour if your agents will spend more than a few minutes in after-call work.

Important

Consult Genesys before setting non-default values for the following options.

The **max-call-duration**, **merge-failed-is-link-timeout**, and **extract-data-stuck-threshold** options in the **[gim-etl]** section must be configured properly to ensure completeness of the call metadata recorded in GIM. For more information on these options, see [Operations-Related Options for Genesys Info Mart](#).

Deploying and Starting Voice Processor

Deploy Voice Processor to the newly configured Podman image, referencing your copies of the default configuration files. This step also starts Voice Processor.

Before deploying, the **settings-override.json** file (or the yaml files you have designated to provide these settings) have several mandatory parameters that must be configured, as described below.

In the **settings-override.json** file, the following parameters are required:

- **rwsBaseUri** – Specifies the address of the RWS cluster that will provide Voice Processor with contact center settings, including tenant-specific configurations such as Genesys Info Mart (GIM) database information and ACW Wait Time. Example: `<code>http://some-rws-host.com:8090</code>`

- **region** – Controls the region section in the metadata POSTed to RWS. This must match the **crRegion** setting of the RWS cluster to which recordings are posted. Example: usa
- **nodeRpsDb** – This section specifies the Voice Processor Persistence Database, which is required for storing recording metadata while Voice Processor is processing them.
 - **database** – Database on the host that will hold recording metadata. Example: noderpsdb
 - **host** – Host of the Persistence Database. Example: noderpsdb.com
 - **port** – Port that the Persistence Database is listening on. Example: 5432
 - **user** – The user name to connect to the Persistence Database.
 - **password** – The password to connect to the Persistence Database.
- **rwsUsername** – The user name for authenticating with RWS.
- **rwsPassword** – The password for authenticating with RWS.

For an example of how the yaml files should be structured, you can refer to the default yaml files that were included with Voice Processor. These files are located at **<INSTALL_DIR>/defaults/**, where **<INSTALL_DIR>** is the location where you extracted the installation files to during the **Preparing your Podman environment** step.

After configuring the default configuration files, deploy and start Voice Processor:

```
sudo ansible-playbook \
  -e docker_config=<defaults directory of Your Docker Configuration Yaml (e.g. docker-
  config.yaml)> \
  -e logger_config=<defaults directory of Your Logger Configuration Yaml (e.g. logger-
  config.yaml)> \
  -e settings_override=<defaults directory of Your Voice Processor Configuration Yaml
  (e.g. settings-override.yaml)> \
  -e secrets=<defaults directory of Your Voice Processor Secrets Yaml (e.g.
  secrets.yaml)> \
  -e service_user=<user name under which gir vp service is running>
  -e service_group=<group name under which gir vp service is running>
  gir-vp-playbook.yml
```

Important

If the above options are not specified, then the .yaml files in **./compose/defaults** will be used.

Before starting the service, ensure proper permissions are set for the certificates and working directory:

1. Set ownership and permissions for the certificate directory:


```
sudo chown -R 1101:1101 <working_directory>/rpsdata
sudo chmod 755 <working_directory>/rpsdata
```
2. Set permissions for key files:


```
sudo chmod 644 <working_directory>/rpsdata/privkey
sudo chmod 644 <working_directory>/rpsdata/pubkey
```

3. Restart the service after permission changes:
`sudo systemctl restart girvp`

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }' girvp.company.com/api/active-version
```

GIR VP container will run as a Systemctl service: `girvp.service`.

When a container stopped, Systemctl will start a new Podman container:

```
sudo systemctl status girvp.service
```

Validating

1. Place a call to an agent or a test agent that is configured for recording.
2. Verify that the call arrives at the SpeechMiner UI. It should take 5 to 15 minutes depending on your configured ACW wait setting.
3. Assuming that live traffic is not recorded, you can use the health check endpoint `<domain:port>/api/status?verbose=1`. The items *recordingsInProgress* or the MCP Post operational status can be helpful in determining whether or not the recording is arriving at the Voice Processor. This also helps you to isolate GVP configuration problems from problems with the Voice Processor service. If a load balancer is used, the node serving the health check may not be the one that handled the recording. Therefore, several health checks may be required to cover the whole cluster.

Upgrading

Docker object configurations and secrets cannot be upgraded. We recommend that you remove the container, update the required configurations, and redeploy.

```
podman container rm <gir_vp>
```

After starting the Voice Processor, update the Voice Processor endpoint (**/api/active-version**) with the version of your Voice Processor instance. You do not require any credentials to do this.

The setting to post the active version:

```
{ "version": "<GIR VP Version>" }
```

Example

```
curl -X POST -H "Content-Type: application/json" -d '{ "version": "9.0.000.25" }' <hostname/Ip address>:8889/api/active-version
```

Miscellaneous Podman tips

- To view a list of running containers:

```
podman ps
```

- To view a list of all containers (started/stopped):

```
podman ps -a
```

- To view the container logs:

```
podman logs <container name>
```

- To remove everything to start again:

```
podman stop <gir_vp> podman container rm <gir_vp>
```

Load balancing with Podman

GIR VP supports load balancing with Nginx and httpd. To set up load balancing in a Premise environment, see [Setting up the Load Balancer in a Single-Tenant Environment](#).

Deploying Genesys Voice Platform for GIR

Genesys Voice Platform (GVP) provides the media services, including IVR, that GIR needs to record contact center interactions.

Installing GVP

Install and configure the GVP solution as described in the [GVP 8.5 Deployment Guide](#). You can learn more about GVP [here](#).

Configuring GVP

GVP uses four components and functions that require additional configuration to enable recording for GIR:

- [Resource Manager](#)
- [IVR Profile](#)
- [Logical Resource Group](#)
- [Media Control Platform \(MCP\)](#)

Resource Manager

1. In the GVP Resource Manager application, configure the following parameters:

Section Name	Parameter Name	Value
rm	conference-sip-error-respcode	Set to 503.
	resource-unavailable-respcode	Set to 603
monitor	sip.proxy.releaseconfonfailure	Set to false.

2. For each GVP shared tenant, a separate tenant is required by Resource Manager. Create a gateway resource for each tenant RM tenant using the SIP Server source address.

IVR Profile

Important

By default the profile named record is used for recording purposes. For IVR recording, the recording parameters associated with the record profile are combined with the

existing IVR profile that is used for the IVR functionality. For additional information, refer to [IVR Recording](#).

1. In Genesys Administrator Extension, navigate to **Configuration > Voice Platform**, select **Voice Platform Profiles**, and click **New**.
2. On the **General** tab, enter the following parameters:
 - **Name** (Genesys recommends naming it record)
 - **Display Name**
 - **Description**
3. On the **Options** tab,
 - Configure for basic authorization:
 - In the **[gvp.service-parameters]** section, set the **recordingclient.callrec_authorization** parameter to fixed, rp_username:rp_password.

Important

The rp_username:rp_password value must be the same username and password that are configured for authorization in the Recording Processor Script or Voice Processor. For more information, see [Recording Processor Script](#) or [Voice Processor](#).

- Configure the following in the **[gvp.service-parameters]** section to set the bitrate and to determine if MP3 recording is mono or stereo:
 - For 8 kbit/s mono:**
 - Set the recordingclient.gvp.config.mpc.mp3.bitrate parameter to fixed,8.
 - Set the recordingclient.channels parameter to fixed,1.
 - For 16 kbit/s stereo:**
 - Set the recordingclient.gvp.config.mpc.mp3.bitrate parameter to fixed,16.
 - Set the recordingclient.channels parameter to fixed,2.
 - For 32 kbit/s stereo:**
 - Set the recordingclient.gvp.config.mpc.mp3.bitrate parameter to fixed,32.
 - Set the recordingclient.channels parameter to fixed,2.
- 4. On the **Recording** tab, add the Recording Certificates, and set the parameters. **[+] Show the table describing the parameters.**

Section	Parameter Name	Description
Recording Destinations	Storage Destination	The path for recording storage on the WebDAV Server. For example, http://<webdav>/recordings.
	Storage HTTP Authorization Header	The credentials for the WebDAV Server. The format is username:password. This field is visible only if the Storage Destination begins with either http or https.

Section	Parameter Name	Description
	Recording Processor URI	The URI that MCP uses to post the metadata of the audio recording after the recording is complete. MCP uses HTTP POST to send the metadata to the Recording Processor or Voice Processor. The format for this parameter is: http://<Recording Processor Host>/api/contact-centers//recordings/. Note: The value for the URI must always end with a forward slash (/).
	SpeechMiner Interaction Receiver	Specifies the URL that points to the SpeechMiner Interaction Receiver responsible for accepting metadata from the Recording Processor Script or Voice Processor for this profile, for example, http://<SpeechMiner Host>/interactionreceiver.
	SpeechMiner Interaction Receiver Authorization Header	Specifies the credentials required to connect to the SpeechMiner Interaction Receiver used by the Recording Processor Script or Voice Processor associated with this profile. The format is username:password, where the username and password are the Interaction Receiver credentials. Note: The user and password value must be the same as the username and password configured in both of the following sections: <ul style="list-style-type: none"> • Configuring SpeechMiner settings in RWS. • Step 5 of Configuring SpeechMiner users.
Speech Analytics Parameters Note: Leave these parameters empty unless you have purchased and enabled speech analytics mode on SpeechMiner; otherwise, recording may not operate correctly.	SpeechMiner Destination	Specifies the URL that points to the SpeechMiner Interaction Receiver responsible for accepting analytics files for this profile, for example, http://<SpeechMiner Host>/interactionreceiver. This is an optional parameter and should be left empty if

Section	Parameter Name	Description
		speech analytics is not enabled.
	SpeechMiner HTTP Authorization Header	Specifies the credentials required to connect to the SpeechMiner Interaction Receiver used for accepting analytics files for this profile. The format is username:password, where the username and password are the Interaction Receiver credentials for analytics. This field is visible only if the SpeechMiner Destination begins with either http or https.
Additional Recording Parameters	Recording Storage MIME Type	The audio file type used for the storage recording. Set to audio/mp3.
	Recording Alert Tone Source (Optional)	The URI of the audio tone. For example, http://example.com/tone.wav.
Recording File Name Template	File Name Template	<p>Specifies the name of the template used for generating the MSML recording. When left blank, the default value is \$id\$. Choose any, or all of the following parameters:</p> <ul style="list-style-type: none"> • ID—The unique identifier of the template. • Date Time—The date and time of the call in which the recording is started. The date and time is sent in ISO format with UTC time. The ISO format is YYYY-MM-DDTHH:MM:SSZ • MCP Date Time—The local date and time of the call in which the recording is started. The local time follows the MCP instance where the recording is taking place. • SIP Server Application Name—The SIP Server application name in which the recording is started. • Call UUID—The call UUID of the call in which the

Section	Parameter Name	Description
		<p>recording is started.</p> <ul style="list-style-type: none"> • ANI—The ANI information of the call in which the recording is started. • Connection ID—The TLib Connection ID of the call in which the recording is started. • DNIS—The DNIS information of the call in which the recording is started. • Agent ID—The agent ID of the DN of the call in which the recording is started. If the recording has not started because the DN or Agent ID has not logged in, this parameter will not be present. <p>For example, if DNIS, ANI and Agent ID are selected, the File Name Template is set to \$dnis\$_\$ani\$_\$agentId\$.</p> <p>Note: Using too many parameters could exceed the 260 characters limit for a Windows file name.</p>

Using multiple locations

A Recording IVR profile enables you to set up a separate voice recording storage location, per data center location, based on the SIP Server geo-location. To use this functionality, create a separate IVR Profile for each geo-location, as follows:

1. Set the following parameters in the **[gvp.general]** section:
 - **service-type**=record
 - **geo-location** (that is, the geo-location that identifies SIP Server location).
2. For each new IVR Profile configure separate Recording Destinations:
 - Storage Destination: Set to the recording storage location for the corresponding data center.
 - Recording Processor URI: Set to the Recording Processor or Voice Processor address for the corresponding data center.
 - SpeechMiner Interaction Receiver: Set to the SpeechMiner Interaction Receiver in the primary data center.

For additional information about multiple data center locations, refer to the [Multiple data center locations](#) page.

Logical Resource Group

A single Media Control Platform (MCP) pool can be used to provide all types of media services including call recording. A dedicated Logical Resource Group can also be used for call recording.

1. Modify a Logical Resource Group to include call recording:
 - Set the **service-types** option to `voicexml;conference;announcement;cpd;media;recordingclient`.
2. Create a new Logical Resource Group. In the **[gvp.lrg]** section, set the following parameters:

Parameter Name	Value
service-types	recordingclient
load-balance-scheme	round-robin
monitor-method	option
port-usage-type	in-and-out
resource-confmaxsize	-1

Important

If using a dedicated Logical Resource Group, ensure that the `recordingclient` value is removed from the MCP pool's **service-types** parameter. For example, set the service type to `voicexml;conference;announcement;cpd;media`.

Media Control Platform

1. Ensure that the Media Control Platform (MCP) instances are included on the **Connections** tab of the Resource Manager Application object.
2. In the **[mpc]** section, set the **default_audio_format** parameter to ULAW, or ALAW, depending on the G711 settings.
3. In the **[mpc]** section, set the **mediamgr.recordwritetimeinterval** parameter to 10000 (10 seconds). The default value is 1000 milliseconds(1 second).
4. In the **[mpc]** section, set the **recordpostretrybackoff** parameter based on the time required to initialize the Recording Processor Script (RPS) or Voice Processor, which depends on the number of agents in the deployment, and how long it takes to retrieve agent information from the configuration environment through the Configuration Server. The initialization time can be determined by examining the RPS log or Voice Processor log and looking for an entry containing "INFO Recording processor is listening on" which indicates that the RPS or Voice Processor is fully initialized. Genesys recommends that the value be set to approximately half the time required for this initialization to complete. For example, if it takes 200 seconds for RPS or Voice Processor initialization to complete, **recordpostretrybackoff** should be set to 100 seconds. Note that this parameter is specified in milliseconds.

Important

- When assigning the MCP(s) for handling call recording, the IP address and Port must match the details of the MCP. Set the **max ports** option to double the number of calls that you want to handle with the MCP. One port is used per stream in the call, one for the customer leg and one for the caller leg. If **max ports** is set to 1000, the MCP can handle 500 calls.
- If screen recording is used, make sure the clock is synchronized to the same time as the agent desktop machines where the Screen Recording Service is installed.
- The **[mpc].recordnumparallelpost** parameter is set to 30 by default and it does not need to be changed during normal operation. However, in a scenario where MCP is posting high number of files to Recording Processor Script (RPS) or Voice Processor and WebDAV, it is recommended to set the value of this parameter based on the sizing calculation: $value2/value1$ where:
 - *value1*: The number of concurrent uploads the WebDAV is able to handle
 - *value2*: The number of recordings that MCP will be posting to WebDAV

For more information about the GVP and Media Server options, see the [Media Control Platform](#).

Encrypting and Provisioning Certificates

Before you configure encryption certificates for voice and screen recordings, you must generate the following keys and certificates:

- A certificate for the Certificate Authority (CA) in .pem format.
- A recording certificate (also known as public key) in .pem X.509 RSA format.
- A recording private key in .pem format.

Important

It is your responsibility to store your private keys and certificates, including the expired ones. You must also back up your keystore, keystore password, certificates and private keys in a secure location offsite to protect against site level disasters. When Genesys Interaction Recording encryption is enabled, loss of the keystore and private key would result in loss of recording files.

While renewing the certificates, keep your old certificates under Administration - Recording Certificates and provision the new certificates using the instructions provided in this section. This will ensure the playback of recordings encrypted with the older certificates without any issues.

Generating the Certificates and Keys

This certificate must meet the following requirements:

- 2048 bit RSA (or higher; please align encryption strength requirements with your IT Security)
- x509 certificate
- PEM format
- The certificate must be signed by a trusted third-party CA, self signed or signed by your own private CA
- If using a third-party CA, the certificate signing request provided to the third-party CA must contain the Subject Name, Serial Number, Subject DN, and Issuer DN. You might be contacted by the third-party CA who might ask for additional information
- The certificate validity period of the certificate determines when the next certificate needs to be generated for renewal

The following OpenSSL command to generate certificate signing request and private key is an example:

```
openssl req -nodes -newkey rsa:2048 -keyout private_key.pem -out cert.req -days <validity period>
```

The system prompts for DN fields to be filled in. You must fill in all of them. See the table below for

the details.

DN Field	Explanation	Example
Common Name	Name of your Recording Solution	Interaction Recording
Organization	The exact legal name of your organization. Do not abbreviate your organization name.	Monster & Sons, Inc.
Organization Unit	Section of the organization.	Robot Repairs
City or Locality	The city where your organization is legally located.	Pleasant Hill
State or Province	Full state or province where your organization is legally located.	California
Country	The two-letter ISO abbreviation for your country.	US

The files will have the following:

- `private_key.pem`— the private key that is used to decrypt the recordings. It must be kept safe and should not be shared.
- `cert.req`— the certificate signing request for the third-party CA that signs the request and provides the public key certificate to be used to encrypt the recordings.

Chained Certificates

Genesys recommends that the recording certificate that you want to use for Genesys Interaction Recording encryption be signed by a single trusted third-party CA.

Important

Chained certificates are certificates where the trusted third-party CA is used to sign the intermediate CA certificate, and the intermediate CA certificate is then used to sign the user certificate.

To set up a chained certificate:

1. Upload the certificate using Genesys Administrator Extension.
2. Obtain the CA file and place it in the MCP's local directory—for example, `/genesys/mcp/certificates/<tenant name>/<ca-file>`. Note that the CA file given here should be the bundle of all the intermediate CA's and the root CA in specific order—for example, `cat crt_inter3.pem crt_inter2.pem crt_inter1.pem root_ca.pem > ca.pem`. When you create a bundle from separate certificates, take note that these certificates might sometimes have additional information that should not be in the final bundle file. If this is the case, the above command (`cat`) will not work, and the information should be copied using an editor that opens the file using the Unix end of line. The information that should be taken starts from:
-----BEGIN CERTIFICATE-----
and finished with the line:
-----END CERTIFICATE-----

3. Configure the CA file path in IVR profile. In the `gvp.service-parameters` section, set the `recordingclient.gvp.config.mpc.mediamgr.CA_file` parameter to `fixed,/genesys/mcp/certificates/<tenant name>/<ca-file>`

For Call Recordings

A Recording Certificate binds a public encryption key to a particular recorded message identity.

Important

- When configuring encryption, backup of the private key is your responsibility. If the private key becomes lost or corrupt, any recording encrypted using that key will become unusable.
- If screen recording is also used in the deployment, it is required that a screen recording certificate is also provisioned. Otherwise, the Recording Muxer Script will not be able to mux the call recording and screen recording together, if the call recording is encrypted but the screen recording is not encrypted.

The following steps describe how to configure encryption for voice recordings:

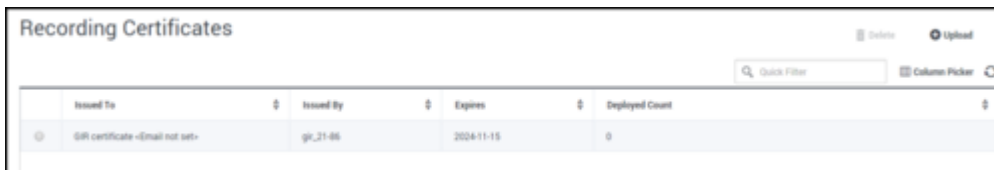
Prerequisites

- A certificate for the Certificate Authority (CA) in .pem format—for example, `ca_cert.pem`.
 - A recording certificate (also known as public key) in .pem format—for example, `02_gir_cert.pem`.
 - A recording private key in .pem format—for example, `02_gir_priv_key.pem`.
1. On the machine where the Recording Crypto Server is installed, place the Certificate Authority (`ca_cert.pem`) in the `<Recording Crypto Server Install Directory>\RCS` directory.
 2. Edit the **rcs.properties** file:
 - a. Change the value of the **cacertstorepath** parameter to `ca_cert.pem`.
 - b. Set the value of the **cacertstorepassword** parameter to the valid password.
 3. Restart the Recording Crypto Server.
 4. Using Recording Plug-in for GAX, edit all your Media Control Platforms (MCP):
 - On the **Options** tab of each MCP application object, in the **[mpc]** section, set the **mediamgr.CA_file** parameter to the location of the Certificate Authority file (for example, `c:\keystore\ca_cert.pem`).
 5. Restart all the MCP instances.

For an example of a certificate, see [Sample Certificate and Key File Generation](#). You are now ready to upload and deploy your certificates to complete the encryption process.

To upload a new certificate:

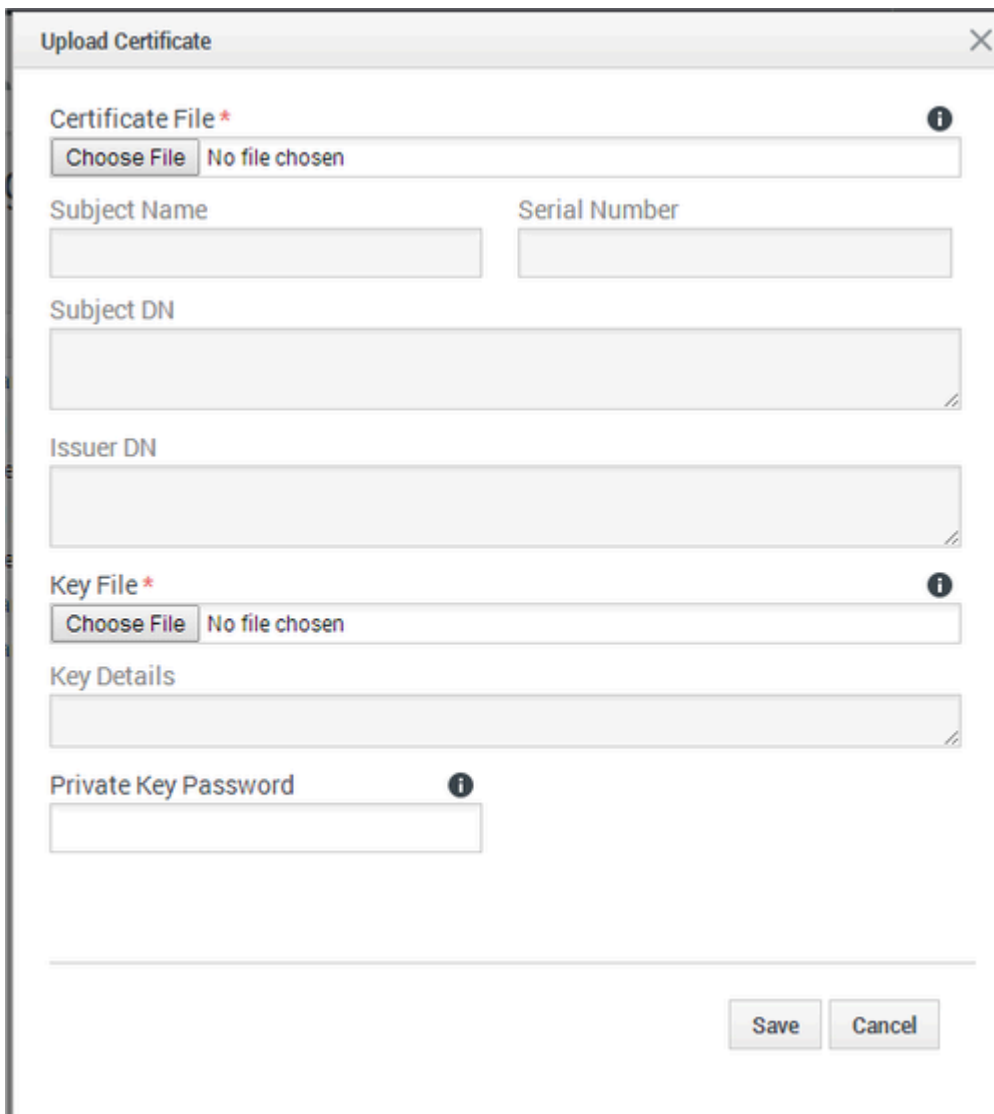
1. Log in to Genesys Administrator Extension, and navigate to **Administration > Certificates**.



The screenshot shows the 'Recording Certificates' panel in the Genesys Administrator Extension. It features a table with columns: Issued To, Issued By, Expires, and Deployed Count. A single row is visible with the following data: Issued To: GRI certificate (Email not set), Issued By: gr_21-06, Expires: 2024-11-15, and Deployed Count: 0. Above the table, there are buttons for 'Delete' and 'Upload', a 'Quick Filter' search bar, and a 'Column Picker' icon.

Issued To	Issued By	Expires	Deployed Count
GRI certificate (Email not set)	gr_21-06	2024-11-15	0

2. On the **Recording Certificates** panel, click **Upload**.



The 'Upload Certificate' dialog box contains the following fields and sections:

- Certificate File ***: Includes a 'Choose File' button and a 'No file chosen' status.
- Subject Name**: A text input field.
- Serial Number**: A text input field.
- Subject DN**: A large text area for the Subject Distinguished Name.
- Issuer DN**: A large text area for the Issuer Distinguished Name.
- Key File ***: Includes a 'Choose File' button and a 'No file chosen' status.
- Key Details**: A large text area for key details.
- Private Key Password**: A text input field.
- Buttons**: 'Save' and 'Cancel' buttons at the bottom right.

3. On the **Upload Certificate** panel, in the **Certificate File** section, click **Choose File**.
4. Select the appropriate file. This file must contain an X.509 RSA certificate in PEM format. The **Subject Name**, **Serial Number**, **Subject DN**, and **Issuer DN** fields automatically populate.
5. In the **Key File** section, click **Choose File**.

6. Select the appropriate file. The file must contain an RSA private key in PEM format. The encoding can be in either OpenSSL RSA private key or PKCS8 format. The **Key Details** field automatically populates.

Upload Certificate [X]

Certificate File * [i]
[Choose File] cert1.pem

Subject Name
Cert1 User <cert1.user@genesyslab.com>

Serial Number
1

Subject DN
C=CA,ST=Ontario,L=Markham,O=Genesys Telecommunications Laboratories,CN=Cert1 User,E=cert1.user@genesyslab.com

Issuer DN
C=CA,ST=Ontario,L=Markham,O=Genesys Telecommunications Laboratories,CN=Certificate Administrator,E=cert.admin@genesyslab.com

Key File * [i]
[Choose File] cert1.key

Key Details
Openssl format private key file

Private Key Password * [i]
[Empty Password Field]

[Save] [Cancel]

7. If the private key file is encrypted, enter the **Private Key Password**.
8. Click **Save**.

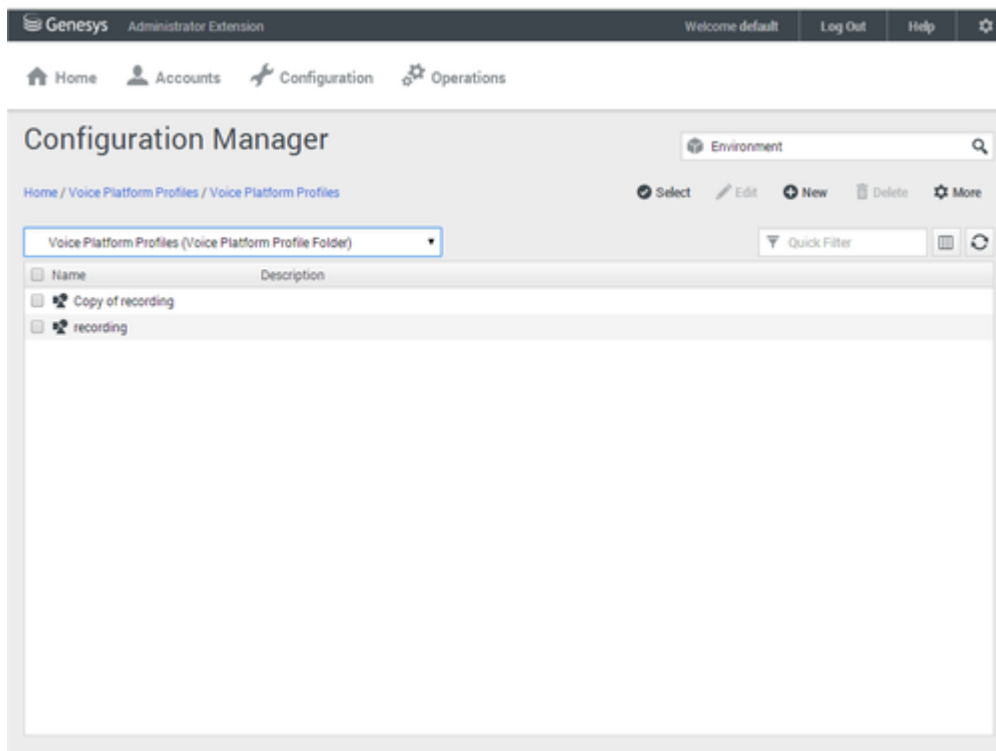
Important

- If you Upload and/or delete recording certificates in one Genesys Administrator Extension session, these changes are not reflected in another Genesys Administrator Extension session. You must log out and log in again to the second Genesys Administrator Extension session.

- If Recording Crypto Server (RCS) is restarted when a Genesys Administrator Extension user is logged in, the next Genesys Administrator Extension operation involving RCS fails because the RCS session saved by the Recording Plug-in for GAX does not exist. RCS will return a 401 "RCS is not available" error. The user must log out, and log in again when receiving the 401 "RCS is not available" error.

To deploy a new certificate:

1. Log in to Genesys Administrator Extension, and navigate to **Configuration > Configuration Manager > Voice Platform Profiles**.



2. From the **Voice Platform Profiles** screen, click the profile that you want to add the certificate to.

The screenshot displays the Genesys Administrator Extension interface. The top navigation bar includes the Genesys logo, 'Administrator Extension', and links for 'Welcome default', 'Log Out', 'Help', and a settings icon. Below this is a secondary navigation bar with icons and labels for 'Home', 'Accounts', 'Configuration', and 'Operations'. The main content area is titled 'Configuration Manager' and shows a breadcrumb trail: 'Home / Voice Platform Profiles / Voice Platform Profiles / recording Properties'. To the right of the breadcrumb are icons for 'Clone', 'Delete', and 'Move To'. On the left, a sidebar contains a list of tabs: 'General', 'Options', 'Permissions', and 'Recording'. The 'Recording' tab is selected. The main form area contains the following fields and controls:

- Name ***: A text input field containing the value 'recording'.
- Display Name ***: A text input field containing the value 'Recording'.
- Description**: A large text area that is currently empty.
- Tenant**: A dropdown menu showing 'Environment' with a search icon.
- State Enabled**: A checkbox that is checked.

At the bottom of the form are three buttons: 'Save', 'Apply', and 'Cancel'.

3. Select the **Recording** tab.

The screenshot shows the Genesys Administrator Extension Configuration Manager interface. The left sidebar contains a menu with 'General', 'Options', 'Permissions', and 'Recording'. The main content area is titled 'Configuration Manager' and shows the breadcrumb 'Home / Voice Platform Profiles / Voice Platform Profiles / recording Properties'. The 'Recording' tab is selected. The 'Recording Certificates' section has an 'Add' button and a table with columns 'Issued To', 'Issued By', and 'Expires'. Below this is the 'Recording Destinations' section with fields for 'Storage Destination', 'Recording Processor URI', 'SpeechMiner Destination', 'SpeechMiner Interaction Receiver', and 'SpeechMiner Interaction Receiver Authorization Header'. The 'Additional Recording Parameters' section has radio buttons for 'Recording Storage MIME Type' (audio/wav and audio/mp3) and a text field for 'Recording Alert Tone Source'. At the bottom are 'Save', 'Apply', and 'Cancel' buttons.

4. Click **Add**.
5. From the **Select Certificate** screen, select the certificate you want to add to the IVR Profile, and click **Add**.

Genesys Administrator Extension

Home Accounts Configuration Operations

Configuration Manager

Home / Voice Platform Profiles / Voice Platform Profiles / recording Properties

General Options Permissions **Recording**

Recording Certificates

Issued To	Issued By	Expires
Cert1 User (cert1.user@genesyslab.com)	Certificate Administrator	2024-04-22

Add Remove

Recording Destinations

Storage Destination

Recording Processor URI

SpeechMiner Destination

SpeechMiner Interaction Receiver

SpeechMiner Interaction Receiver Authorization H...

Additional Recording Parameters

Recording Storage MIME Type

☐ audio/wav

☒ audio/mp3

Recording Alert Tone Source

Recording File Name Template

File Name Template

☐ ID ☐ SIP Server Application Name ☐ ANI ☐ DNIS

☐ Date Time ☐ Call UUID ☐ Connection ID ☐ Agent ID

☐ MCP Date Time

Save Apply Cancel

6. Click **Save**.

Important

In Genesys Administrator Extension, do not open **certificates-n** (where **n** is 1, 2, 3, and so on) options using the **Options** tab of the IVR Profile for editing. If opened for editing and saved without making any changes, the certificate will be corrupted. Instead, always use the **Recording** tab of the IVR Profile for certificate administration. To fix this issue, remove the certificate using the **Recording** tab of the IVR profile, add it again, and then save.

For Screen Recordings

Assigning Certificates

To assign a new certificate:

1. Using Genesys Administrator Extension, in the header, go to **Administration > Screen Recording Certificates**.
2. On the **Screen Recording Certificates** panel, click **Add**.
3. From the **Select Certificate** window, perform one of the following actions:
 - Select the check box next to the appropriate certificate, and click **Add**.
 - Click **Cancel** to discard any changes.
4. Perform one of the following actions:
 - Click the **Save** button to accept the changes.
 - Click the **Cancel** button to discard the changes.

Setting up the Decryption Proxy

1. Configure the Recording Crypto Server (RCS) locations that Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) uses for encrypted screen recordings:
 - For a single location:
 - a. Using a text editor, create the **create_single_location** file using the following command:

```
{
  "name": "decrypt-uri-prefix",
  "location": "/",
  "value": "<rcs uri>/rcs"
}
```

Important

Replace `<rcs uri>` with the appropriate value.

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_single_location http://<Web Services
Server>:8080/api/v2/ops
/contact-centers/<contact center ID (in hex format)>/settings/screen-recording
--header "Content-Type: application/json"; echo
```

- For multiple locations:
 - a. Using a text editor, create the **create_first_location** file using the following command:

```
{
  "name": "decrypt-uri-prefix",
  "location": "<node_location>",
  "value": "<rcs uri>/rcs"
}
```

```
}
```

- b. Execute the following command:

```
curl -u ops:ops -X POST -d @create_first_location http://<Web Services Server>:8080/api/v2/ops/contact-centers/<contact center ID (in hex format)>/settings/screen-recording --header "Content-Type: application/json"; echo
```

Important

Replace <node_location> with the appropriate value. The values for the <node_location> are similar to the **nodePath** settings in the Interaction Recording Web Services (Web Services) **application.yaml** file (if you are using Web Services and Application version 8.5.201.09 or earlier, refer to the **nodePath** setting in the **server-settings.yaml** file instead), but allow a hierarchical representation. For example, an Interaction Recording Web Services (Web Services) node uses a **decrypt-uri-prefix** setting with a location of "/US" if the **nodePath** set to "/US/AK" or "/US/HI".

- c. Repeat steps a and b for each location required.

For more information on the properties of these group settings, see [Interaction Recording Web Services Group Settings](#).

Important

If you upload and/or delete recording certificates in one Genesys Administrator Extension session, these changes are not reflected in another Genesys Administrator Extension session. You must log out and log in again to the second Genesys Administrator Extension session.

Deploying the Screen Recording Service

Genesys Interaction Recording (GIR) requires that a Screen Recording Service (SRS) be installed on each Agent's desktop to enable the Agent to capture what is happening on the screen at the time of an active interaction.

The procedures on this page show how to download, install, configure and test the Screen Recording Service.

Important

- For blended agents that are configured to support the handling of both voice and non-voice interactions, GIR will perform screen recording of voice interactions only.
- If the Screen Recording Service is restarted while a recording is in progress or when trying to close a recording, an extra **vlc.exe** process might be left running in the system. If this happens, use Task Manager to stop any remaining **vlc.exe** processes.

Prerequisites

The following list provides you with the requirements you need to successfully deploy the Screen Recording Service (SRS):

- Before you can install and use the SR Service on your desktop, you must have the following information ready at hand. Your IT department or Genesys Professional can help you get this information.
 - Access to Workspace Web Edition (WWE) or Workspace Desktop Edition (WDE)
 - The software (minimum version 8.5.302.10)
- When the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) server is deployed with a GoDaddy, without including the G2-to-G1 cross certificate (that is, the intermediate certificate), you must perform one of the following manual workarounds:
 - Download the G2-to-G1 certificate from https://certs.godaddy.com/repository/gd_bundle-g2-g1.crt.
 - Include the G2-to-G1 cross certificate in the server side certificate (concatenated with the server certificate).
 - Import the G2-to-G1 cross certificate into the SYSTEM (Local Machine) Root CA certificate store.
- Verify that the client machine meets the following minimum specifications:
 - Pentium Dual Core CPU
 - 2 GB RAM (800 MB available for the SR Service)
 - A minimum of 5 GB of available space (in total) for the SR Service installation and working space.

- When the Interaction Recording Web Services server (or Web Services if you are using version 8.5.210.02 or earlier) is deployed with a self-signed certificate, you must import the certificate to **Trusted Root Certification Authorities** for the current user (for example, My User account) and local agent's workstation (for example, Compute Account), from the Certificates Microsoft Management Console (MMC) snap-in .
- If you are running Bria 4 on Windows 7, you must enable Windows Aero. If you do not enable Windows Aero, the Screen Recording Service may fail to capture the Bria 4 application.
- Verify the client machine is synchronized to the same time as the machine on which Media Control Platform (MCP) is installed.
- Starting from version 8.5.500.19, the Microsoft Visual C++ 2015-2022 Redistributable (x64) must be installed on the machine.

Installation considerations

After verifying that your system meets the basic prerequisites, you should consider the following:

- The recommended installation procedure will install the Screen Recording Service's self-signed PFX certificates to the root certificates store. For more information, see [Creating Self-Signed Certificates](#).
- When required use one of the following options to query the Screen Recording Service (SR Service) version:
 - Run the following command line `wmic datafile where name='C:\\<Installation Directory>\\GenesysServiceHandler.exe`.
 - Open the web browser and navigate to <https://127.0.0.1/version> if the SR Service is deployed with HTTPS enabled or <http://127.0.0.1:8080/version> if the SR Service is running as HTTP.
- Proxy support for outbound connections from SRS can be enabled either with or without authentication support.
 - The parameters used to configure the SRS Proxy are available in [Advanced configuration for the Screen Recording Service](#).
- When a proxy is used it may interfere with the SR Service operation. The SR Service runs as an HTTP server and relies on an incoming socket connection to correctly identify the agent's windows session. If the HTTP requests are forwarded by a proxy, the SR Service may not be able to correctly identify the user session in a multi-user environment. With a single user, the SR Service will rollback to the currently active windows session.
When a proxy is used it is recommended that localhost (127.0.0.1) connections be excluded from the proxy settings.
When the proxy is an internal system service (like an Antivirus\Firewall), it is recommended that the SRS related processes (SrsProcess.exe and GenesysServiceHandler.exe) be added to the security software exception\white list.
- The Screen Recording Service can be used by a Citrix client. The following Citrix configurations are supported:
 - Citrix XenApp 7.x or Citrix XenDesktop 7.x running under Windows Server 2019
- In a Citrix environment (for Genesys SR Service 8.5.230.23 and later), SRS only supports a single session per remote PC (Session Sharing is not supported).
- In a Citrix environment, Genesys recommends to configure the `screen-recording.client.max-attempts` parameter to 15 to avoid ping timeout from Workspace Web Edition (WWE) during upgrade,

as upgrade usually takes longer duration in Citrix environment. For more information on the screen-recording.client.max-attempts parameter, see [Integrating with Workspace Web Edition](#).

- In a Citrix environment (for Genesys SR Service 8.5.370.85 and later), the SR Service can be configured to work with Citrix's Virtual Loopback feature.
 - Configure the authenticationHost parameter so that the SR Service uses a loopback IP address that is outside of the range being used by the Citrix Virtual Loopback Feature. See [Advanced Configuration for the Screen Recording Service](#) for more details on how to configure the authenticationHost parameter.
- If SRS is deployed on a Citrix VDA, you need to disable Citrix API hooks for vlc.exe by creating the following registry values. For more information, see [How to Disable Citrix API Hooks on a Per-application Basis](#).
 - **Keys:**
HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\CtxHook
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\CtxHook64
 - **Value Name:** ExcludedImageNames
 - **Type:** REG_SZ
 - **Value:** vlc.exe
- If the IPv4 SRS authenticationHost parameter is configured to something other than 127.0.0.1, then use that IP address instead of 127.0.0.1 in the above URLs. See [Advanced Configuration for the Screen Recording Service](#) for more details.
- If the IPv4 SRS authenticationHost parameter is configured to something other than 127.0.0.1, and SRS is configured to use HTTPS, then use that IP address when creating self-signed certificates. See [Creating Self-Signed Certificates to support IP Loopback Addresses other than 127.0.0.1](#) for more details.
- The SR Service can be used in a VMware Horizon environment. The following VMware Horizon configuration is supported:
 - VMware Horizon 7 running under Windows Server 2019
- If you are using [Workspace Web Edition](#) or [Workspace Desktop Edition](#) and the SR Service with Genesys Softphone in a VDI environment (such as Citrix Xenapp), you must configure the screen-recording.client.address option to point to the SRS Loopback address.

Screen Recording Service - operating systems

The Screen Recording Service is supported on the following operating systems in a non-Citrix mode:

- Windows 10 (64-bit)
- Windows 11

The Screen Recording Service is supported on the following operating systems for Citrix support:

- Windows Server 2022
- Windows Server 2019

The Screen Recording Service is supported on the following operating system for VMware Horizon support:

- Windows Server 2022
- Windows Server 2019

Recommended screen resolutions

Genesys has tested the Screen Recording Service under the following recommended screen resolutions. If you use the Screen Recording Service on a computer with a different screen resolution than listed above, you should do a field validation of the Screen Recording Service in your setup to ensure that it is working properly. If you encounter unexpected results, Genesys recommends that you set your screen resolution to one of the recommended and tested resolutions listed below.

Warning

If a field validation has been completed against an earlier version using a non-supported resolution, there is no guarantee that resolution will continue to work on upgrades to new releases. Only supported resolutions are continually tested against each new version.

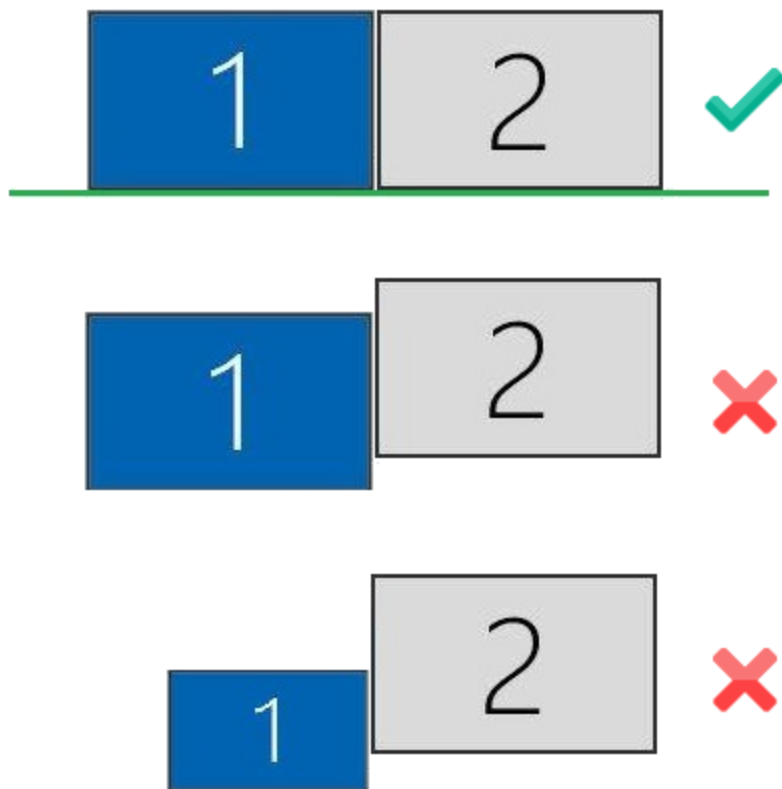
Single Monitor:

- 1024 x 768
- 1280 x 720
- 1600 x 1200
- 1920 x 1080

Dual Monitor:

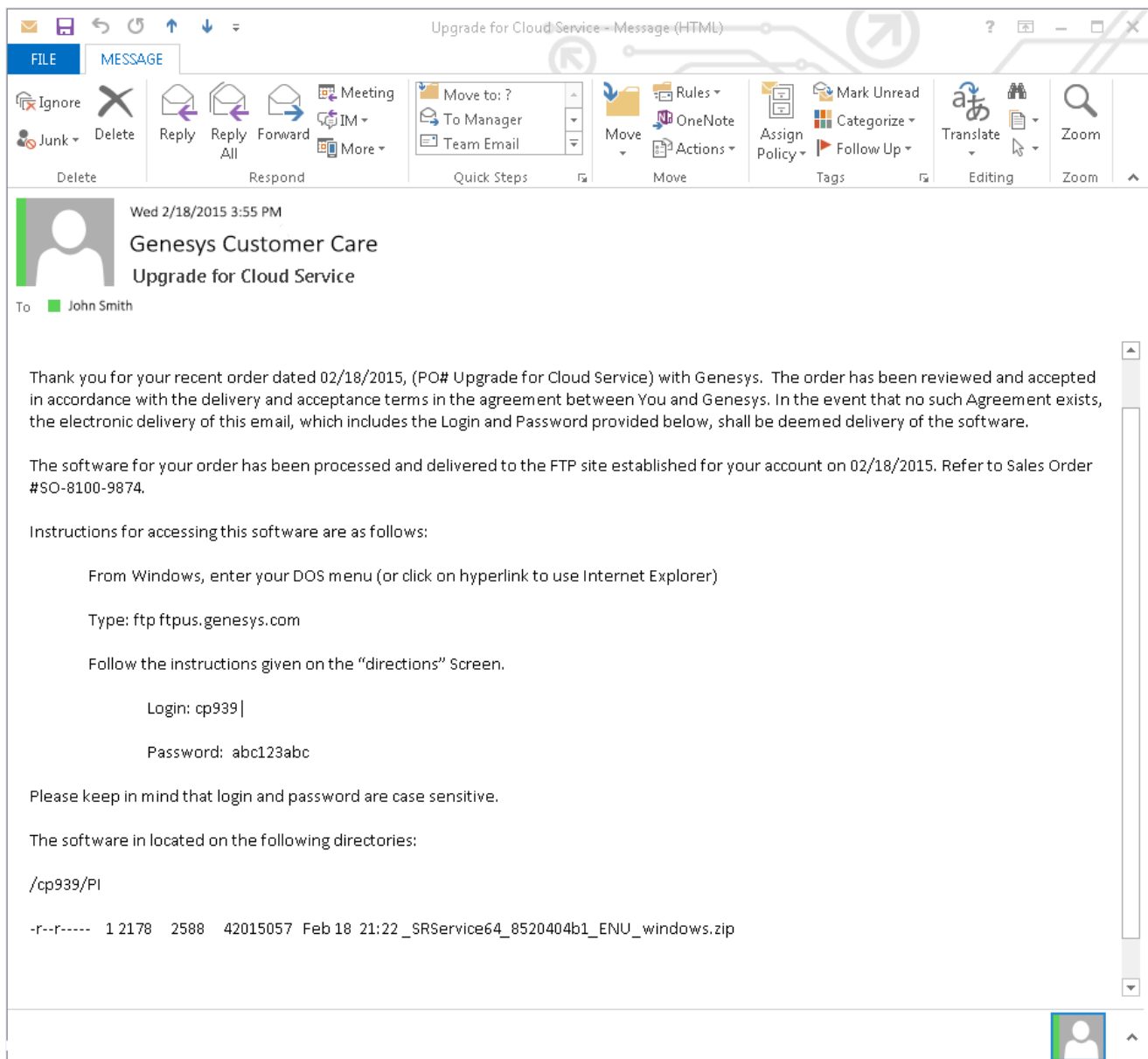
- Side-by-side 1024 x 768 + 1024 x 768
- Side-by-side 1280 x 720 + 1280 x 720
- Side-by-side 1600 x 1200 + 1600 x 1200
- Side-by-side 1920 x 1080 + 1920 x 1080

When using dual monitors, set both displays to the same resolution and arrange them side-by-side (*not* offset) in your display settings, as shown here:



Using dual monitors in a non-recommended configuration can result in errors.

Get your software



Find the email you received from Genesys with the details about your software (it will look similar to the example), and using your favorite FTP client—for example, Filezilla, connect with the credentials listed in the email.

Download the zipped file to a temporary folder on your computer.

Installing your software for the first time

There are two ways to install the SR Service by using:

- **Installation Wizard - for version 8.5.3 and later**
- **Command Prompt**

Important

- To install the Screen Recording Service you must have Administrator privileges.
- Firefox users must close the browser before installing the Screen Recording Service. If Firefox is open while Screen Recording Service is being installed, restart the browser after the installation is completed.

Installing the SR Service for the first time with the installation wizard

This installation procedure is for version 8.5.3 and later.

1. Locate the setup.exe and double-click its icon. The installation wizard is activated.
2. Select one of the following options and click **Next**:
 - **Standard**: Installation will not collect the user's input and proceeds with the default values.
 - **Advanced**: Installation will collect the user's input only for specific configuration parameters.
 - **Customized**: Installation will collect the user's input for all the required parameters.

Important

- For SRS versions 8.5.345.24 and later, selecting the **Standard** mode of installation installs SRS in the HTTPS mode. For SRS versions below 8.5.345.24, this option installs SRS in the HTTP mode.
- The **Use HTTPS self-signed certificates** option is configurable only when the **Advanced** or **Customized** mode of installation is selected. When the **Use HTTPS self-signed certificates** option is selected, SRS uses the HTTPS mode. When this option is not selected, SRS uses the HTTP mode.
- If you select **Use HTTPS self-signed certificates**, you must also specify the following:
 - Base URL for allowed Server Host Names: `https://*.genesyscloud.com`
 - GWS Server URL: `https://<server_name>:443`

3. Select **Use an existing configuration file** (optional) to copy the configuration of one machine, to all other installations of the SR Service on different machines in the same deployment. In the **Location** field, enter the location of the existing configuration file and click **Next**.
4. Select **Use HTTPS self-signed certificates** (SRS uses HTTPS mode). Enter the **Base URL for allowed Server Host Names** and the **GWS Server URL** (see the note, above).
5. For the **Select Certificate Validation** option, select one of the following options: **Do Not Validate the certificate option**, **Validate the certificate using Windows certificate store**, or **Validate the certificate using self-signed certificate**.
6. Verify that the location in the **Destination Folder**, is the correct location (that is, the location where the SR Service will be installed) for the SR Service. If it is not the correct location, enter the correct location and click **Next**.
7. Click **Install**, to complete the first time installation.

Installing the SR Service for the first time with the command prompt

1. Open a command prompt, and type `cd` to change directories to the installation folder.
2. At the prompt, enter the following command and press **Enter**:

```
setup.exe /s /z"-s '<C:\genesys_silent.ini>' -sl '<setup log file name>' -t '<setup wizard log file name>'"
```

For more information, refer to the [Advanced Configuration for the Screen Recording Service](#) section.

Important

- Set the configured `genesys_silent.ini` file path in the command line. Use the absolute path for the input file parameters.
For example, run `setup.exe /s /z"-s 'c:\genesys_silent.ini' -sl 'c:\setup.log' -t 'c:\setup_wizard.log'"`
- The `genesys_silent.ini` file must be configured when using command line silent installation and an unused parameter must be commented out in the `genesys_silent.ini` file. The standard **genesys_silent.ini** file is included with the installation package.
 - The **genesys_silent.ini** file provides all possible configuration parameters along with a description of each.
 - The file lists all the parameters with placeholders.
 - Verify that the unused configuration parameters are either deleted or commented.
 - Verify that the configuration file contains at least the following parameters:
[SRServer]
InstallationType=Standard
[IPCommon]
InstallPath=<Absolute path where the SRS needs to be installed>
[MaintMode]
Mode=FirstInstall

- For SRS versions 8.5.345.24 and later, HTTPS is used by default for the HTTP server within SRS. For the earlier versions of SRS, HTTP is used by default. To change this mode, you must use `InstallationType=Advanced`, `CertificateValidation=UseWinCertStore` and then set `HTTPS=true` or `HTTPS=false`.
- For additional security options, consult a Genesys Professional.
- During the installation process, the antivirus program may block the installation when the installation process detects that the antivirus program is attempting to make system changes. In this scenario, the user will have to unblock the installation program to continue the installation.

Verify the installation

Use Windows Explorer to locate the directory where you installed the software. For example, `C:\Program Files (x86)\GCTI\Genesys SR Service \Logs\GSR`. Once you see the folder is there, restart your computer to confirm that the service starts automatically.

To verify the version installed, browse to `https://127.0.0.1/version` or `http://127.0.0.1:8080/version`.

Enable Screen Recording

In addition to installing and configuring the Screen Recording Service, the agent desktop application must be configured appropriately to support screen recording, and screen recording must be enabled within Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). For further details, refer to the [Enable Screen Recording](#) section.

Test the service and validate the installation

After installation, use Windows Services to confirm that the Genesys SR Service is 'Started'. Check the startup log file as follows:

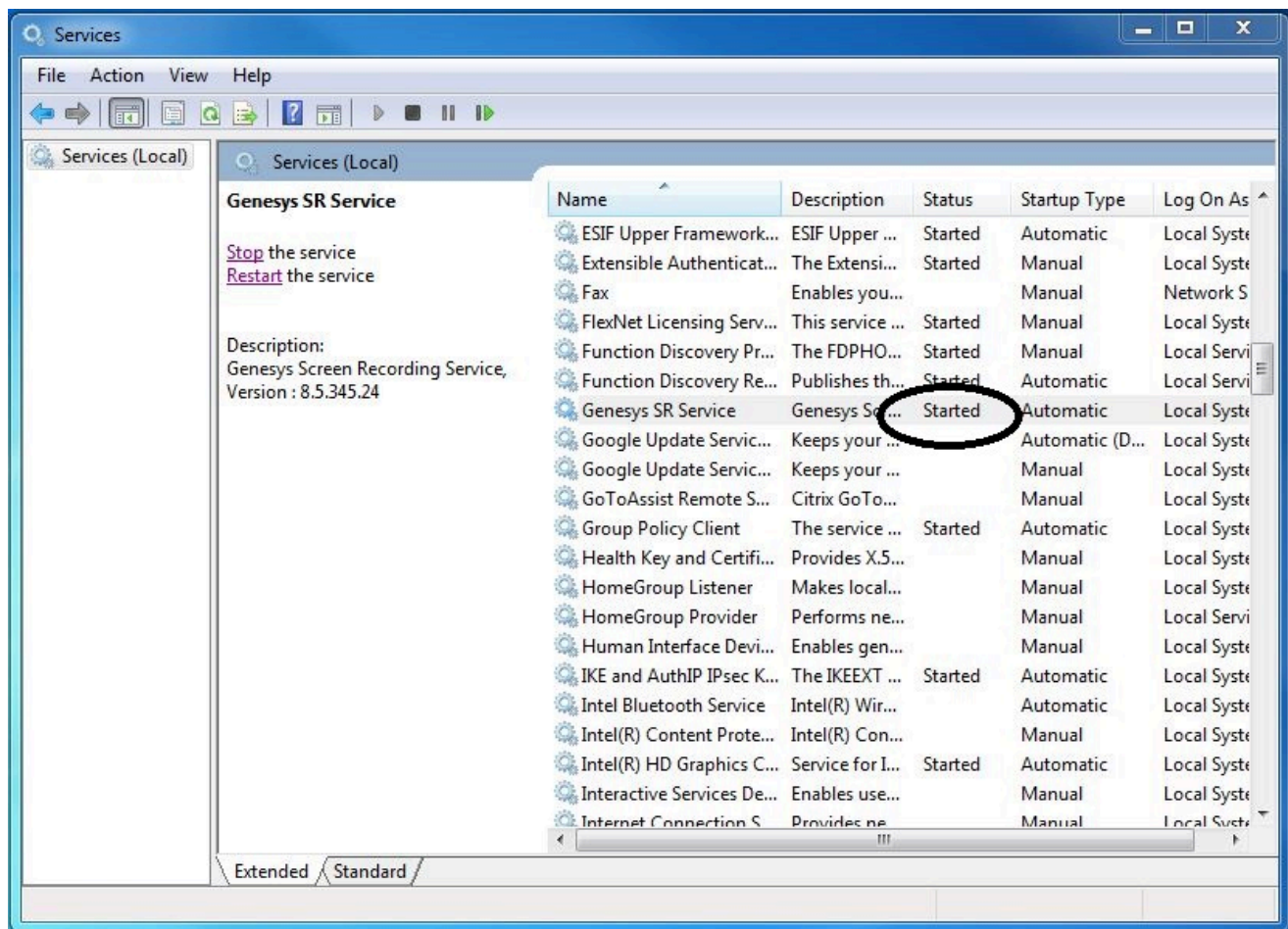
1. Open the `C:\Program Files (x86)\GCTI\Genesys SR Service \Logs\GSR.log` file, and make sure that something similar to the following lines are included (with the version reflecting the version you have just installed):
`ServiceHandler: Running Version:8.5.230.23, IP:135.39.66.17, OS:win32`
2. Make sure that the `C:\Program Files (x86)\GCTI\Genesys SR Service \Logs\GSR.log` file contains no errors

or exceptions.

3. Use the agent desktop to login as an agent that has been configured to have their voice interactions recorded. When the **recordingWhen** parameter is not set to off, the screens will also be recorded when the Screen Recording Service is running. Once logged-in as an agent, request an inbound call to that agent, or use the agent desktop to initiate an outbound call (For example, to a cell phone). Keep the interaction active for 10-20 seconds, and then disconnect the call. Proceed with step 4 to review the log file.
4. After the test, review the C:\Program Files (x86)\GCT\Genesys SR Service \Logs\GSR.log for the following line: Uploader: Upload of file <file-name-of-media> was successful.

Tip

If your installation is unsuccessful, contact your Genesys Professional.



Upgrading the Screen Recording Service

Screen Recording Service can be upgraded manually or automatically. Both types of upgrades assume a functional existing deployment of Screen Recording Service. If the functionality of the existing deployment is in question, it is recommended to look for and stop the service, delete the previous installation folder and proceed as though this is the first time deploying the software. Contact your Genesys Professional if you are not sure if the software is working.

Manual upgrade from any version to 8.5.302.10

1. Create a backup copy of the C:\Genesys\SRC directory and name the backup directory C:\Genesys\SRC.backup.
2. Unzip your new software in a temporary directory (for example, C:\temp).
3. Update the **.ini** file. Access the temporary directory and type the following command in a command prompt window:

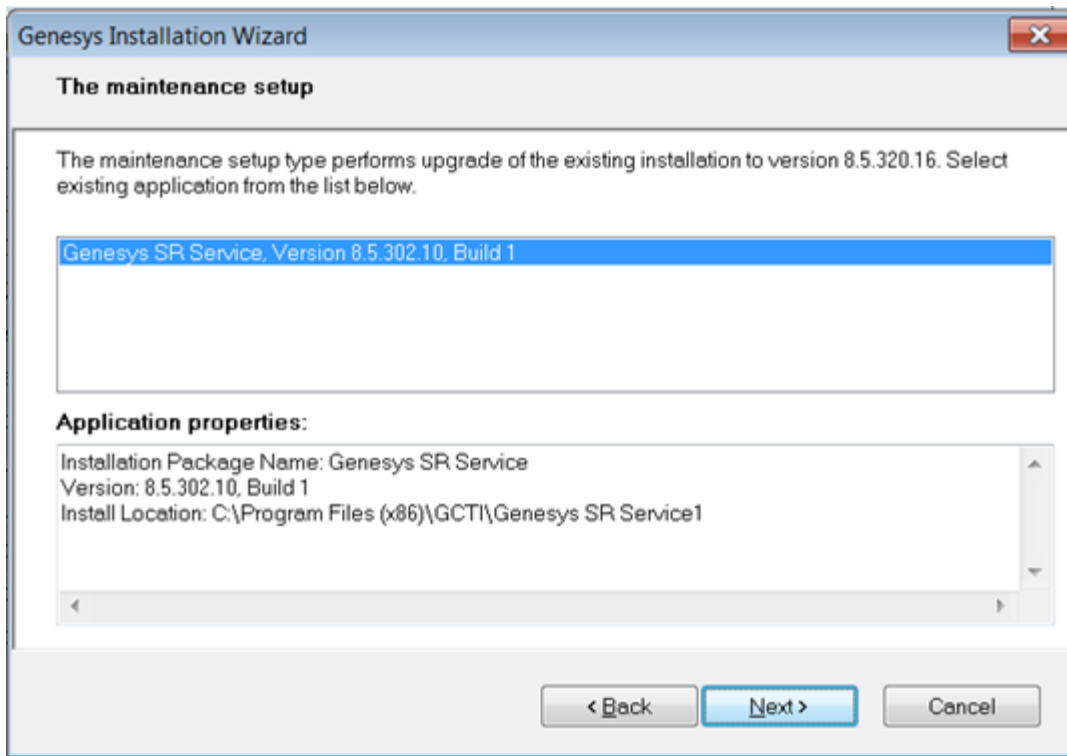
```
setup.exe /s /z"-s '<genesys_silent.ini>' -sl '<setup log file name>' -t '<setup wizard log file name>'"
```
4. Validate the upgrade using the steps in the [Test the Service and Validate the Installation](#) section above.

Manual Upgrade

Important

- The following steps must be performed by a System Administrator.
- Before you upgrade to a newer Screen Recording Service version, check with your Genesys Professional about compatibility with your system.

1. Copy the new SR Service software to a temporary directory.
2. Run the **setup.exe**. As shown in the following image, the setup process automatically detects the existing SR Service installation and selects it for upgradation.



3. Click **Next** and follow the instructions provided in the [Installing the SR Service for the first time with the installation wizard](#) section above.
4. Validate the upgrade using the steps in the [Test the Service and Validate the Installation](#) section.

Automatic Upgrade

Important

When the SR Service Automatic Upgrade attempts to download and install a new version, an Anti-Virus\Firewall may block the SR Service upgrade from downloading and executing the new version files and subsequently prevent automatic upgrades. To prevent this block, it is recommended to add the SR Service processes (SrsProcess.exe, GSRUpdateService.exe and GenesysServiceHandler.exe) to the security software exception / white list.

1. Upload the new SR Service software to the Web Server by copying the content in the IP folder to a location on your Web Server that does not require HTTP authorization. For example, `https://<IP Address>/src/ip/setup.exe`.
2. Configure Genesys Framework to push a new SR Service version to the Agent's Desktop.
 - a. Update the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) Cluster application object in the configuration database. On the **Annex** tab, edit the **[screen-recording-client]** section and add the parameters. **[+] Show the table that describes**

the parameters.

Parameter Name	Description	Required	Example
softwareVersion	The new version of the Screen Recording Service.	Y	8.5.303.02
softwareUrl	The link to the setup file that you copied to the Web Server as part of the IP folder content. If this location is not accessible from the agent desktop that you are intending to upgrade to, the upgrade will not be possible.	Y	https://<IP Address>/src/ip/setup.exe
updateWhen	<p>Determines when to run software update when available. Available options are:</p> <ul style="list-style-type: none"> restart - The safest option in order to not lose any recording. With this option the upgrade will be installed during the next system restart. logout - Runs software update once all agents have logged out. If the agent logs in before the update is complete, they risk losing the screen recording session. In this case, the SR Service will be restarted after the update is complete. immediate - Will shut down SRS and install the new version regardless of the current state (that 	N	restart

Parameter Name	Description	Required	Example
	is, even if a recording is taking place). In this case, SRS will be restarted after upgrade is complete.		

Important

Genesys SR Service on each agent desktop checks for the availability of new software at regular interval, once every 24 hours by default, but can be altered through the configuration parameter **sleepNoNewVersion** on the RWS application object. The new software version is identified and downloaded to the agent desktop when the SoftwareVersion configured on RWS application is greater than the one that is actually installed on the agent's desktop. Later it would start upgrading according to the **updateWhen** parameter.

Upgrading while using HTTPS with an IP address other than 127.0.0.1

When SR Service is upgraded, the self-signed HTTPS certificates are removed and new ones are generated and installed. The newly generated HTTPS certificates will be for the IP address 127.0.0.1. If the IPv4 SRS authenticationHost parameter (see [Advanced Configuration for the Screen Recording Service](#) for more details about the authenticationHost parameter) is configured to something other than 127.0.0.1, then the HTTPS certificates will not work.

To continue using HTTPS with an IP address other than 127.0.0.1, new HTTPS certificates must be generated. Follow the instructions in [Creating Self-Signed Certificates to support Virtual Loopback Addresses](#) to create and install new HTTPS certificates.

Rollback to a previous version

To rollback to a previous version of the Screen Recording Service:

Important

- The SR Service only supports a manual rollback.
- Recordings captured but not uploaded will need to be manually moved to the upload folder of the active SRS directory after the rollback is complete.

1. In the Task Manager, verify that **Genesys SR Service** is stopped. If it has not been stopped, stop it now.
2. Copy the current C:\Program Files (x86)\GCTI\Genesys SR Service directory to a different folder. (For example: C:\Program Files (x86)\GCTI\Genesys SR Service.<date>). This directory contains recordings that have not yet been uploaded; it may be needed for subsequent troubleshooting purposes.
3. Uninstall the existing SR Service installation.
4. Install the previous SR Service version.
5. Restart your computer or start the Genesys SR Service Windows service.
6. Validate the rollback using the steps in the [Verify the Installation](#) section above.

Uninstalling the Screen Recording Service

1. Open the **Start** menu and select **Control Panel**.
2. Click **Programs and Features**.
3. In the **Name** column, select the **Screen Recording Service** entry (for example, Genesys SR Service 8.5.xxx.yy), right click and select **Uninstall**.

The Screen Recording Service is uninstalled.

Deploying the Screen Recording Service - Advanced Configuration

The following sections provide advanced Screen Recording Service installation and configuration steps.

For basic instructions about how to install and configure the Screen Recording Service, see: [Deploying the Screen Recording Service](#).

Enable Screen Recording

Important

Before you can start to capture and play back the screen recordings, you must make sure that you have configured the [Interaction Recording Web Services components](#) (or [Web Services components](#) if you're using version 8.5.210.02 or earlier), and [encryption](#) specifically for screen recording.

To set up recording conditions, using Genesys Administrator Extension, add the **recordingWhen** parameter to the **[screen-recording-client]** section of the Interaction Recording Web Services or Web Services Cluster application object, depending on your deployment (see [Installing RWS](#)).

When this parameter is set in the Cluster object, the recording condition applies to all agents in the environment. You can create the **recordingWhen** parameter in a **[screen-recording-client]** section of each agent object to override the settings at the environment level.

The parameter value is an expression of conditions to enable screen recording for each agent. The format is:

- For Non-voice agents: **recordingWhen** = condition1,condition2,... where condition1, condition2, etc. are a set of conditions that must all be true in order for the screen recording to be taking place.
- For Voice agents: Screen recording starts when the voice recording starts except in cases where **recordingWhen** is explicitly set to off.

Important

For blended agents that are configured to support the handling of both voice and non-voice interactions, GIR will perform screen recording of voice interactions only.

Integrating with Workspace Web Edition

If your agents use Workspace Web Edition (WWE) as their desktop, screen recording must be set up as follows:

Important

- The SR Service does not support single sign-on for WWE.
- If the following Internet Explorer 11 settings are enabled when the SR Service is used together with WWE, you must work with SR Service version 8.5.302.14 or later:
 - Enhanced Protected Mode under the Miscellaneous settings
 - Enable Protected Mode under Security Setting
 - Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) address is added to the Local Intranet sites

Using Genesys Administrator Extension, add the following parameters to the interaction-workspace section of the Web Services Cluster application object.

Important

If you are working with HTTP, the Screen Recording Service default port number is 8080. If you are working with HTTPS, the default port number is 443. In addition, verify that the Workspace Web Edition configuration is set to 8080 or 443.

Parameter Name	Mandatory	Description	Default Value
privilege.screen-recording.can-use	Y	Specifies whether agents can use screen recording. If set to true, the integration module is loaded and sends credentials to the client.	false
screen-recording.client.address	N	Specifies the IP address that the Screen Recording Service listens for credentials on. Valid values: 127.0.0.1, [::1]	127.0.0.1
screen-recording.client.port	N	Specifies the port that the Screen Recording Service listens for credentials on.	443

Parameter Name	Mandatory	Description	Default Value
screen-recording.client.ping-interval	N	Specifies the interval, in milliseconds, between ping requests to the Screen Recording Service.	5
screen-recording.client.max-attempts	N	Specifies the maximum number of attempts to establish communication with the Screen Recording Service. Note: In a Citrix environment, set the value of this parameter to 15.	5
screen-recording.client.secure-connection	N	Specifies if a secure connection will be made to the Screen Recording Service using HTTPS.	true
screen-recording.client.server-url	N	Defines the Interaction Recording Web Services (Web Services) server address that the Screen Recording Service will use for communication.	

Integrating with Workspace Desktop Edition

If your agents use Workspace Desktop Edition as their desktop, screen recording must be set up according to the instructions in the [Workspace Desktop Edition Deployment Guide](#).

Important

The SR Service does not support single sign-on for Workspace Desktop Edition.

Enable Screen Recording for a Contact Center

To enable the screen recording feature for a given Contact Center refer to the [Configuration for Screen Recordings > Configuring the Interaction Recording Web Services Parameters](#) section.

Advanced Installation Procedures

Creating Self-Signed Certificates

During installation the SR Service can create self-signed certificates to be used as local host

connections. To do this, select the **Use HTTPS self-signed certificates** check box in the advance installation. For the SR Service version 8.5.345.24 and later, selecting the **Standard** option installs SR Service in the HTTPS mode and creates a self-signed certificate.

To create self-signed certificates as local host connections, following installation, perform the following:

1. Open a command window as an Administrator.
2. Navigate to the `<install_dir>\Certificates\Self-Signed` directory.
3. Run the **create_certificates.bat** file. This creates a set of unique self-signed certificates.
4. Run the **install_certificates.bat** file. This installs the new self-signed certificates to Windows trusted certificates store.

Important

- If the SR Service is installed with self-signed certificates for the local host server, the certificates are automatically imported into the Firefox certificate database. If Firefox is installed after the SR Service is installed, the certificates must be imported manually. To import the self-signed certificates into the Firefox database, run the following script as an administrator `<install_dir>\Certificates\Firefox\add_certificates.bat`.
- When the SR Service starts, it will attempt to read the certificate files `server.pem` and `serverIp6.pem` in the `<install_dir>\Certificates` directory. If these files are missing, the SR Service will run in HTTP mode instead of HTTPS mode.

Installing Your Own Certificates

If desired, you can use your own certificates as follows:

1. Provide a certificate for the IPv4 host, 127.0.0.1, in the `<install_dir>\Certificates\server.pem` file.
2. Install the .pfx form of this certificate to the local certificates store as a "Trusted Root Certification Authority" file.
3. Provide a certificate for the IPv6 host, `:::1`, in the `<install_dir>\Certificates\serverIp6.pem` file.
4. Install the .pfx form of this certificate to the local certificates store as a "Trusted Root Certification Authority" file.
5. The PEM certificate files should include both the private RSA key and the certificate itself. **[+] Show an example.**

```
-----BEGIN RSA PRIVATE KEY-----
.
.
.
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
.
.
.
-----END CERTIFICATE-----
```

Important

When the .pem certificates must be protected by a password, the password is configured in the `config.json` file using the `certificatePassword` parameter. The default `certificatePassword` is `genesyscreenrecording`. For more details, refer to the [Client Side parameters](#) table in step #2 of the Advanced Configuration for the Screen Recording Service section.

Creating Self-Signed Certificates to support IP Loopback Addresses other than 127.0.0.1

SRS can be configured so that its Authentication Server uses Loopback IP Addresses other than 127.0.0.1. The HTTPS Certificates that are created by default only work if SRS is configured to use the Loopback IP Address 127.0.0.1. To use SRS with Loopback Addresses besides 127.0.0.1 and HTTPS, new HTTPS Certificates must be created specifically for the Loopback IP Address that SRS is using.

To create self-signed certificates with Loopback addresses other than 127.0.0.1, following installation, perform the following:

1. Open a command window as an Administrator.
2. Navigate to the `<install_dir>\Certificates\Self-Signed` directory.
3. Run **`uninstall_certificates.bat`** to remove the existing certificates.
4. Run **`create_certificates.bat`** and pass a value for the **`IPV4_HOST`** parameter. Below is an example to create certificates for 127.1.1.2:

```
create_certificates.bat -IPV4_HOST 127.1.1.2
```
5. Run **`install_certificates.bat`** to install the new certificates. This installs the new self-signed certificates to the Windows trusted certificates store.
6. Configure SRS to use the newly created certificates. Please see the **`authenticationCertificate`** option in [Advanced Configuration for the Screen Recording Service](#) for more details.
7. Restart the Genesys SR Service Windows service.

Advanced Configuration for the Screen Recording Service

Some Screen Recording Service configurations are managed locally on the system (that is, using the `config.json` configuration file). Other configurations are managed centrally. Advanced configuration should be performed using the Interaction Recording Web Services or Web Services Cluster application object, depending on your deployment (see [Install RWS](#)) in Genesys Administrator Extension. All the configuration parameter values should be in JSON notation. More information about how JSON escapes rules can be found here: <https://msdn.microsoft.com/en-us/library/dn921889.aspx>.

Important

Screen Recording Service does not support the use of System Proxies configured via PAC (Proxy Auto-Configuration) files.

Important

The default port number for SRS is 443. If this port is used by another application, you must configure the **authenticationPort** and **authenticationPortIp6** parameters to use a different port. The following parameter for Agent Desktop must also be changed accordingly:

```
[interaction-workspace] screen-recording.client.port
```

1. If your server uses a self-signed certificate, set the **certificate** parameter to the path on the file system where the pem file is stored.
2. Edit the local **config.json** file on the Screen Recording Service machine, and add the client parameters.
Note: The following parameters should ONLY be configured locally and NOT using GAX. Please note that in a multiple user SR Service deployment these settings will take effect for all users using the system.

Important

All parameter names are case sensitive.

Name	Mandatory	Description	Default value
addressType	N	<p>Enables the identification of the SR Service for monitoring and reporting purposes on Interaction Recording Web Services (Web Services). addressType supports the following options:</p> <ul style="list-style-type: none">• fqdn - Use fully qualified domain name.• ip - Use IPv4.• ip6 - Use IPv6. <p>Note: With addressType you</p>	fqdn

Name	Mandatory	Description	Default value
		can also provide a custom name to identify a specific machine (for example, pc-id-1).	
allowedHosts	N	<p>Represents a list of allowed host names to be configured as the SRS Interaction Recording Web Services (Web Services) server address using POST API. The value can be a single address, a list of specific addresses or a wild card.</p> <p>Notes:</p> <ul style="list-style-type: none"> When SRS receives the server configuration parameter from Workspace Web Edition (WWE) after the agent logs in, it will check if the URI matched the allowedHosts configuration parameter. If there is a match, it will establish a connection to the Genesys Web Services with the information provided regardless of whether or not the Server parameter is configured locally. If the parameter does not match, the SR Service will only use the Server parameter if it is configured. If the Server parameter is not configured, the SR Service will not establish a 	https://*.genesyscloud.com

Name	Mandatory	Description	Default value
		<p>connection with Interaction Recording Web Services (Web Services).</p> <ul style="list-style-type: none"> If the server parameter is not configured in the config.json file and it is passed in real time as part of the login POST API, you must update the <code>allowedHosts</code> so that it matches the server address of Interaction Recording Web Services (Web Services) in the actual deployment. You can configure multiple host URLs for <code>allowedHosts</code> in the manner in which JSON presents multiple values (<code>["URL1", "URL2", . . . , "URLN"]</code>). For example, <code>{"name": "allowed Hosts", "value": ["URL1", "URL2"]}</code>. 	
allowedOrigins	N	<p>Specifies the approved CORS Origin headers that Screen Recording Service approves. If it is not provided, the <code>*</code> character is set as the default, which means any request will be approved, with or without origin header.</p> <p>If it is provided, the value can be a single origin, or a list of approved origins, that is used to approve the CORS requests. The defined server parameter is always added to the list of approved</p>	*

Name	Mandatory	Description	Default value
		origins automatically.	
authenticationCertificate	N	<p>The relative or full path to the authentication server's PEM certificate. If a value is available, the authentication server uses it for the HTTPS connection to the agent's desktop.</p> <p>Note: This parameter is not needed in the default Screen Recording Service installation. The Screen Recording Service uses the default self-signed certificate (%install_dir\Certificates\server.pem) automatically.</p>	'%install_dir\Certificates\server.pem'
authenticationCertificateIp6	N	<p>The relative or full path to the authentication server's PEM certificate for IPv6. If a value is available, the authentication server uses it for the HTTPS connection to the agent's desktop.</p> <p>Notes:</p> <ul style="list-style-type: none"> This parameter is not needed in the default Screen Recording Service installation. The Screen Recording Service uses the default self-signed certificate (%install_dir\Certificates\server.pem) automatically. Each host requires a unique certificate. 	%install_dir\Certificates\serverIp6.pem
authenticationHost	N	The IPv4 Address that the Authentication Server will bind to when SRS starts if SRS is configured to use IPv4. The parameter	127.0.0.1

Name	Mandatory	Description	Default value
		value must be an IPv4 address within 127.0.0.0/8. The IP addresses 127.0.0.0 and 127.255.255.255 are not allowed.	
authenticationPort	N	The port used for internal communication with Web Services.	If using HTTP, the port is 8080. If using HTTPS, the port is 443.
authenticationPortIp6	N	The port used for internal communication with Web Services.	If using HTTP, the port is 8080. If using HTTPS, the port is 443.
certificate	N	Indicates how the Screen Recording Service validates the Web Services server TLS certificate. If set to false, the Screen Recording Service will not validate the certificate. If set to true, the client will validate the certificate using the Windows certificate store when the server is using a certificate from the public CA. If set to a file path (for example, C:\Automation\server.pem), the Screen Recording Service will validate the certificate using a self-signed certificate when the server is using a private self-signed certificate.	true
certificatePassword	N	The password for the PEM certificate's private RSA key.	Empty
certificatePasswordIp6	N	The password for the IPv6 PEM certificate's private RSA key.	Empty
credentialsTimeout	N	The timeout duration, in minutes, between the keep alive (GET/Ping) requests from the agent's desktop and the Screen Recording Service. When the	35 Note: This value must be longer than the Web Services session timeout duration for the agent's desktop. By default the Web

Name	Mandatory	Description	Default value
		timeout expires, the agent's credentials are deleted from the Screen Recording Service's cache.	Services session timeout is 30 minutes, and the credentialsTimeout is 35 minutes. The latter must be increased if the Web Services session timeout is increased.
diskCheckInterval	N	The interval, in seconds, between disk space checks.	30
diskFreeSpaceLimit	N	The minimum disk space, in MB, on the client machine. When the disk space drops below this value, the screen recording will stop any active recording sessions.	2000
diskFreeSpaceThreshold	N	The amount of free space available above the defined limit, before recordings can be restored, after dropping below the disk space limit.	500
ip6	N	Indicates whether to support IPv6 in addition to IPv4 for communication with Web Services.	true
isVlcSlowCapture	N	Indicates that VLC has delay in starting the screen recording. If set to true, the Screen Recording Service (SRS) will update the start time of the screen recording with the time the media file is created. The valid values are true and false. Warning: This parameter is deprecated by preLoadVlc .	false
partitionedCookies	N	Enables partitioned cookie to support new changes in the Google Chrome browser related to sharing of third-party cookies. The configuration parameter,	2

Name	Mandatory	Description	Default value
		<p>partitionedCookies, supports the following options:</p> <ul style="list-style-type: none"> • 0 - disabled (never add the Partitioned cookie attribute) • 1 - enabled (always add the Partitioned cookie attribute) • 2 - auto (enable the Partitioned cookie attribute conditionally when Chromium version requires it; this is the default) • If Chrome/Edge version is 118 or higher, partitionedCookies will be enabled. • If Chrome/Edge version is lower than 118, partitionedCookies will be disabled. • For Firefox, partitionedCookies will be enabled for all versions. 	
peer_server	N	<p>The server base url of the backup data center. The default port is 80; to use a different port, use the url:port format.</p> <p>This value will be overridden if supplied by the client application.</p> <p>Note: This parameter is not applicable for single data center deployments.</p>	Empty
postProcessingSavePath	N	<p>The post processing temp location. When used as a UNC path, verify that the computer running SRS</p>	%LOCALAPPDATA%/Genesys/SRS (C:/Users/<user_name>/AppData/Local/Genesys/SRS)

Name	Mandatory	Description	Default value
		(SYSTEM account) has read\write permissions.	
preLoadVlc	N	Decides whether to load VLC process in advance after the agent logs in. Valid values are true and false. Warning: Only configure this parameter if instructed by Genesys.	false
proxyServerHost	N	The proxy server hostname or IP address.	Empty
proxyServerPort	N	The proxy server port.	Empty
proxyServerUsername	N	The username to connect to the proxy server.	Empty
proxyServerPassword	N	The password to connect to the proxy server.	Empty
reEncodingTimeoutSeconds	N	Specifies the number of seconds that Screen Recording Service will wait for VLC to finish processing a screen recording after a call that includes pause and resume operations. Valid values are any integer greater than 0. Warning: Only configure this parameter if instructed by Genesys.	120
rwsRetryBeforeSwitchOver	N	The number of times SRS will attempt to connect to the primary RWS before switching over to RWS in the backup data center and vice versa.	1
sendLogToGWS	N	Disables the sending of an error log to Interaction Recording Web Services (Web Services) from the SR Service.	false
server	N	The server base url.	Empty

Name	Mandatory	Description	Default value
		The default port is 80; to use a different port, use the url:port format.	
sharedSavePath	N	The Shared folder. The location in which recordings are saved to be uploaded. When used as a UNC path, verify that the computer running SRS (SYSTEM account) has read\write permissions.	<Installation_dir>
statusTimeout	N	The timeout duration, in seconds, between the keep alive GET/ Ping requests from the agent's desktop and the Screen Recording Service.	60
systemMetricTimeout	N	The timeout duration, in seconds, for reading the system metrics. On a slow machine, set a higher timeout value to avoid timing out from reading the system metrics.	5
userSavePath	N	The user recordings temp location. The location must be a local folder. If a non-default location is used, verify that the user has read\write permissions.	%LOCALAPPDATA%/Genesys/SRS (C:/Users/<user_name>/AppData/Local/Genesys/SRS)
useSystemProxy	N	If this value is true, the Screen Recording Service uses the Windows System Proxy settings.	false
vlcHttpTimeout	N	The HTTP request timeout, in seconds, for VLC start and stop recording commands.	10
vlcPortBegin	N	The beginning of the port range for the VLC http interface.	4916
vlcPortEnd	N	The end of the port range for the VLC http interface.	65530

Important

Proxy server parameters specified in the config.json file take precedence over the **useSystemProxy** parameter.

3. In the most basic configuration, you will not need to add the following parameters, they are all optional. However, if you intend to use any of the server parameters, use Genesys Administrator Extension, and follow the next steps:
 - a. At the Environment level, locate the Interaction Recording Web Services or Web Services Cluster application object, depending on your deployment (see [Install RWS](#)).
 - b. Edit the application object, and create a new section named **screen-recording-client**. The following table provides an example of the **screen-recording-client** section.

Important

All parameter names are case sensitive.

Name	Mandatory	Description	Default value
CaPath	N	The path for the authority PEM certificate file used for verification of encryption certificates. If not present, verification will not take place.	false
cleanupPolicy	N	<p>Specifies the method for managing failed screen recording files on the Client machine. The available values are:</p> <ul style="list-style-type: none">• delete – Deletes the recording from the local drive, regardless if the upload was successful or not.• keep – Deletes successfully uploaded recordings. Recordings whose upload failed are	keep

Name	Mandatory	Description	Default value
		<p>kept in the Recordings folder and retried until they are successfully uploaded.</p> <ul style="list-style-type: none"> keepForever - All recordings are permanently stored on the local drive. Successfully uploaded recordings are stored in the Uploaded sub-folder. Recordings whose upload failed are kept in the Recordings folder and retried until they are successfully uploaded. <p>Note: This setting is only recommended for debugging, as it can cause disk space to run out quickly.</p>	
clockColor	N	The color of the time stamp clock. Use HTML color codes.	0xffffffff (white)
clockFormat	N	The display format for the time stamp clock. See the table later in this section for the valid values.	%H:%M:%S-%Y-%m-%d %Z (HH:MM:SS-YYYY-MM-DD TZ)
clockOpacity	N	How non-transparent the time stamp clock displays. Valid values: 0-255	150
clockPosition	N	The position for time stamp clock. Valid values: 0=center, 1=left, 2=right, 4=top, 8=bottom. You can also use combinations of these values—for example, 6 = top-right.	8 (bottom-center)

Name	Mandatory	Description	Default value
clockSize	N	The font size for the timestamps written to the screen. Note: This option is available if the timeStamp option is set to true.	40
delayShutdown	N	The time, in seconds, to delay shutting down the SRS and the system if the uploadPolicy parameter is set to immediate. This allows all the screen recording files to upload before the shutdown starts. The maximum value is 125 seconds (limited by Windows).	15
encodingLevel	N	The H.264 encoding level restriction. Valid values: 10,11,12,13,20,21,22,30,31,32,40,41,42,50,51. For more information, see H.264/MPEG-4 AVC Levels .	
encodingProfile	N	The H264 encoding profile. Valid values: baseline, main, high.	high
folder	N	The folder name where the media is uploaded in the WebDAV server.	/
fps	N	Frames per second.	1
grayScale	N	Indicates whether to record the screen in color or gray scale. Set to true to record in gray scale. Set to false to record in color.	false
ignoreCertificateVerificationErrors	N	Ignores errors that occur during certificate verification for screen recording encryption. This option is used only when certificate verification is enabled by configuring the CaPath parameter. Valid values:	true

Name	Mandatory	Description	Default value
		<ul style="list-style-type: none"> • true: The errors that occur during certificate verification will be ignored with a warning message being logged. • false: The errors that occur during certificate verification will not be ignored. 	
isACWEnabled	N	<p>Indicates whether to record the agent when they are in the After Call Work (ACW) state.</p> <p>Note: You must also configure the wrap-up-time parameter under the T-Server or Agent Login object. (The Agent Login object is not supported for deployments using SIP Cluster.) For more information, see Agent Login on the Deploying SIP Server for GIR page.</p>	<p>true</p> <p>Note: If isACWEnabled is set to any value other than false, then the value is true.</p>
keepAspectRatio	N	Indicates whether to keep the original aspect ratio or stretch the video to fill the screen if the screen resolution is large than the maximum resolution, and the screen is down scaled.	true
logsToKeep	N	The number of log files to keep.	10
logLevel	N	The logging level. Set to one of the following: debug, info, warning, error, critical. Only messages with a level set equal to or above the defined level will be logged.	info
maxDurationMinutes	N	The maximum duration, in minutes, before slicing a screen recording file.	<p>According to the selected qualityPreset:</p> <p>low-180 standard-120 high-75</p>

Name	Mandatory	Description	Default value
maxHeight	N	The maximum height resolution in pixels. The client will always use the lower resolution defined by either the maxHeight/maxWidth parameters or the resolutionScale parameter.	1080
maxLogSize	N	The maximum size, in MB, of the log file before a new log file is created. The old log file is named with the .1 extension. Set the value to 0 if you do not want to limit the log file size.	5
maxWidth	N	The maximum width resolution in pixels. The client will always use the lower resolution defined by either the maxHeight/maxWidth parameters or the resolutionScale parameter.	1920
multipleMonitorsEnabled	N	Indicates whether to record on all available monitors. If set to false, the client will record on the primary display monitor only.	true
qualityPreset	N	Defines the desired bitrate, depending on the agent's screen resolution. Valid Values: <ul style="list-style-type: none"> low—Emphasis is on storage capacity, and text is readable 90% of the time. For example, 120 kbit/s for 1920 x 1080 resolution with color. standard—Text should be readable 100% of the time 	standard

Name	Mandatory	Description	Default value
		<p>with normal use. For example, 150 kbit/s for 1920 x 1080 resolution with color.</p> <ul style="list-style-type: none"> high—Emphasis is on quality, and text should be readable 100% even on a high movement environment. For example, 190 kbit/s for 1920 x 1080 resolution with color. <p>See the table later in this section for the full list of preset examples.</p>	
recordingWhen	N	<p>An expression from configuration states when screen recording should be taking place for a particular recording client. The format is:</p> <p>recordingWhen= <code>condition1,condition2,...</code> where <code>condition1,condition2,...</code> are a set of conditions that must all be true in order for the screen recording to take place. In other words: Screen Recording Active = condition1 && condition2 && ...</p> <p>Note: If the state of any of the conditions is unknown (occurs only before first determining agent state, so limited to initial state), then the state of screen recording is unknown. See the table later in this section for the full list of conditions.</p>	random_voice(100)
resolutionScale	N	Used to scale the screen size. Setting resolutionScale to 0.5 will resize the screen resolution in half. Setting it to 1 will do nothing. The client	1

Name	Mandatory	Description	Default value
		will always use the lower resolution defined by either the maxHeigh/maxWidth parameters or the resolutionScale parameter.	
rwsFailedRecordingRetrySleep	N	The time, in minutes, to sleep before retrying recordings that failed to upload.	15
rwsRetryBeforeSwitchOver	N	The number of times SRS will attempt to connect to the primary RWS before switching over to RWS in the backup data center and vice versa.	1
sleepNoConnection	N	The maximum time, in minutes, that the client will sleep if there is no connection with the server before attempting to reconnect.	1
sleepNoNewVersion	N	The time, in minutes, that the client updater thread will sleep if a new version is not available.	1440(24H)
slowMachine	N	Indicates whether the Screen Recording Service is installed on a slow machine, so that the extra time is available to save the video files before closing the client. Note: slowMachine has been deprecated by vlcCloseTimeout as of 8.5.302.14.	false
softwareChecksum	N	The SHA512 checksum of the latest software setup file.	Empty
softwareUrl	N	The URI used to fetch the latest Screen Recording Service software installation package.	None
softwareVersion	N	The latest Screen	Empty

Name	Mandatory	Description	Default value
		Recording Service software version number.	
systemMetricTimeout	N	The timeout duration, in seconds, for reading the system metrics. On a slow machine, set a higher timeout value to avoid timing out from reading the system metrics.	5
timeout	N	The timeout duration, in seconds, for HTTP requests. This value must be bigger than the cometD Server request timeout.	60
timeStamp	N	Indicates whether a time stamp will be water marked on the video.	false
updateWhen	N	<p>Determines when to run software update when available.</p> <p>Available options are:</p> <ul style="list-style-type: none">• restart - The safest option in order to not to loose any recording. With this option the upgrade will be installed during the next system restart.• logout - Runs software update once all agents have logged out. If the agent logs in before the update is complete, they risk losing the screen recording session. In this case, the SR Service (SRS) will be restarted after the update is complete.• immediate - Will shut down SRS and install the new	restart

Name	Mandatory	Description	Default value
		<p>version regardless of the current state (that is, even if a recording is taking place). In this case, SRS will be restarted after upgrade is complete.</p> <p>Note:</p> <ul style="list-style-type: none">• To receive a new version of the SR Service, you must first log into Workspace Web Edition (WWE). Use the immediate option with caution. Since the SR Service is restarted immediately, this may cause screen recordings to be lost and may require the agent to logout and login again to restore the screen recording operation. If possible always use the logout or restart option.• If updateWhen is set to restart and the system is restarted before all the SRS installation files are downloaded, the download process continues after the system is restarted. However, the software is updated only during the next system restart.	
uploadPolicy	N	Specifies the screen recording upload policy. If set to window, the screen recording files are uploaded to	immediate

Name	Mandatory	Description	Default value
		<p>storage during the times specified by the windowStartTime and windowEndTime parameters. If set to immediate, the files are uploaded immediately; however, after the agent's last call, the Screen Recording Service needs some time to upload the recording to the server before the Agent's desktop shuts down. The amount of time needed depends on the duration of the last call and network speed. Genesys recommends to estimate one minute for every minute of screen recording on a network with 150 kbit/s per second and an upload speed approximate to 20 KB per second. For example, if the last screen recording lasted 10 minutes, and the network speed is 300 kbit/s (~40KB/s), five minutes is required.</p> <p>Note: If the Agent's PC is shutdown before the upload is completed, the recording will be uploaded on next PC start up.</p>	
videoBitrate	N	Encoding bitrate. Use this parameter to override the default bitrate that is calculated based on the resolution and the selected qualityPreset value.	150 kbit/s for 1920 x 1080 resolution (standard preset, color recording)
vlcCloseTimeout	N	Sets the amount of time the SR Service will wait, after stopping a screen recording, before closing VLC. This time is required to	2

Name	Mandatory	Description	Default value
		ensure VLC completes writing the file correctly. This time should not be changed unless the SR Service is running on a very slow machine, and the screen recording file is invalid but without an error in the log. If such a scenario occurs, increase the time the SR Service must wait before closing VLC.	
windowEndTime	N	Specifies the upload end time, in the local time. The format is hh:mm—for example, 23:00. This parameter is mandatory for the Window upload policy.	Empty
windowStartTime	N	Specifies the upload start time, in the local time. The format is hh:mm—for example, 23:00. This parameter is mandatory for the Window upload policy.	Empty

Video File Size/Compression Optimization Estimate

The following table provides file size estimates according to the selected **Quality Preset**, FPS and color scheme, given a specific resolution.

Preset	Color	Resolution	Frame Rate	Encoding Level	Average File Size MB/Minute
Low	Color	1920x1080	1	High 4.1	0.864
Standard	Color	1920x1080	1	High 4.1	1.055
High	Color	1920x1080	1	High 4.1	1.37
Low	Grayscale	1920x1080	1	High 4.1	0.608
Standard	Grayscale	1920x1080	1	High 4.1	0.732
High	Grayscale	1920x1080	1	High 4.1	0.886

Recording Conditions

The following table describes the recording conditions for the **recordingWhen** parameter:

Condition	Description
off	A special case. Cannot appear with other conditions. When specified as such, screen recording never occurs for the agent.
loggedin	When the agent is logged in
DNDoff	When agent sets DND (do not disturb) to off
ready(any)	True when any media type is set to ready, or <code>list(ready media).count != 0</code>
ready(abc)	True when the abc media type is set to ready
ready(abc,...xyz)	A list of media types that are set to ready. Note that <code>ready(abc,...xyz) = ready(abc) ... ready(xyz)</code> .
random_voice(%)	Records the agent's screens based on a percentage of the total voice call volume for that agent.

Important

Each individual setting's key/value can be overwritten at the agent level by setting the Person object with the Annex of the same section name (**screen-recording-client**).

Clock Format Directives

The follow table lists and describes the values that are available for the **clockFormat** parameter.

Directive	Meaning
%a	Locale's abbreviated weekday name.
%A	Locale's full weekday name.
%b	Locale's abbreviated month name.
%B	Locale's full month name.
%c	Locale's appropriate date and time representation.
%d	Day of the month as a decimal number [01,31].
%H	Hour (24-hour clock) as a decimal number [00,23].
%I	Hour (12-hour clock) as a decimal number [01,12].
%j	Day of the year as a decimal number [001,366].
%m	Month as a decimal number [01,12].
%M	Minute as a decimal number [00,59].
%p	Locale's equivalent of either AM or PM.

Directive	Meaning
%S	Second as a decimal number [00,61].
%U	Week number of the year (Sunday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Sunday are considered to be in week 0.
%w	Weekday as a decimal number [0(Sunday),6].
%W	Week number of the year (Monday as the first day of the week) as a decimal number [00,53]. All days in a new year preceding the first Monday are considered to be in week 0.
%x	Locale's appropriate date representation.
%X	Locale's appropriate time representation.
%y	Year without century as a decimal number [00,99].
%Y	Year with century as a decimal number.
%Z	Time zone name (no characters if no time zone exists).
%%	A literal '%' character.

Deploying Recording Muxer Script

Genesys Interaction Recording (GIR) needs the Recording Muxer Script to combine the call and screen recordings for a seamless playback.

Recording Muxer Script (Python 3)

Prerequisites

Before installing and configuring the Recording Muxer Script, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage or S3 Premise where the recordings are stored.
- For Recording Muxer Script 8.5.500.10 (or higher), Recording Processor Script must be upgraded to 8.5.500.13 (or higher) if using Recording Processor Script.

Installing Recording Muxer Script

Installing on Windows

1. Install 64-bit Python 3.11.5 from the [Python](#) website. To make Python 3 to work with OpenSSL 3.0.13, follow the below steps:
 - Download `libcrypto-3.dll` and `libssl-3.dll` from the [Python Binary repository](#).
 - In `[python-source-folder]\DLLs`, replace with the above downloaded DLL files.
2. Install **Recording Muxer Script IP** with the installer.

Note: Install the following third-party libraries in the order they appear and untar the files in Administrator mode.
 3. Untar the `<Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz` file.
 4. Run `py -m pip install .` from the `<Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1` directory.
 5. Untar the `<Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz` file.

-
6. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
 7. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
 8. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
 9. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
 10. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
 11. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
 12. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
 13. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
 14. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
 15. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
 16. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
 17. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
 18. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
 19. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.
 20. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0 directory.
 21. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
 22. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
 23. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
 24. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
 25. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
 26. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
 27. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
 28. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
 29. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
 30. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
 31. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
-

32. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
33. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
34. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
35. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
36. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
37. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
38. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
39. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
40. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
41. Unzip the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-win64-static-gpl3.0.zip.
42. Unzip the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-3.0.13-win64.zip. This OpenSSL library is used to encrypt the resulting muxed recording file when required.

Installing on Linux (RHEL)

1. Install `zlib-devel` (`yum install zlib-devel`).
2. Install `sqlite-devel` (`yum install sqlite-devel.x86_64`).
3. Install `libffi-devel` (`yum install libffi-devel`).
4. Install OpenSSL.
 - For 8.5.500.09 or lower versions, install OpenSSL version 1.1.1.
 - For 8.5.500.10 or higher versions, install OpenSSL 3.0.13. Download OpenSSL 3.0.13 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-3.0.13 --openssldir=/usr/home/openssl-3.0.13 --libdir=lib no-shared`
5. Install 64 bit Python 3.11.5.
 - For 8.5.500.09 or lower versions, compile with OpenSSL 1.1.1 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
 - For 8.5.500.10 or higher versions, compile with OpenSSL 3.0.13 from the [Python](#) website. While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-3.0.13 --enable-optimizations`
6. Install the **Recording Muxer Script IP** with the installer provided.

Note: Install the following third-party libraries in the order they appear.

-
7. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz file.
 8. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1 directory.
 9. Untar the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz file.
 10. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
 11. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
 12. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
 13. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
 14. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
 15. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
 16. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
 17. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
 18. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
 19. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
 20. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
 21. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
 22. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
 23. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.
 24. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0 directory.
 25. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
 26. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
 27. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
 28. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
 29. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
 30. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
 31. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
 32. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
-

33. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
34. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
35. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
36. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
37. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
38. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
39. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
40. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
41. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
42. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
43. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
44. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
45. Perform one of the following steps depending on the Muxer version.
 - For 8.5.500.03, untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-centos7-x86_64-static-gpl3.0.tar.bz2.
 - For 8.5.500.09 or higher versions, untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-rhel8-x86_64-static-gpl3.0.tar.bz2.
46. Execute `chmod a+x ffmpeg` and `chmod a+x ffprobe`.
47. Perform one of the following steps depending on the Muxer version. The OpenSSL library is used to encrypt the resulting muxed recording file when required.
 - For 8.5.500.03, untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-linux-x86_64.tar.bz2.
 - For 8.5.500.09, untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-rhel8-x86_64.tar.bz2.
 - For 8.5.500.10 or higher, untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-3.0.13-rhel8-x86_64.tar.bz2.
48. Execute `chmod a+x openssl`.

Important

- GIR does not support direct upgrade of Muxer from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip`

install command as mentioned above.

- Run `sudo dnf install libnsl` if you encounter the following error while executing Muxer installation script (`install.sh`):
./Perl: error while loading shared libraries: libnsl.so.1: cannot open shared object file: No such file or directory.

Configuring Recording Muxer Script

This section describes how to configure the Recording Muxer Script for your environment.

Configure Passwords (Optional)

Important

In a Linux or Windows environment, Muxer supports the use of environment variables instead of parameters in the configuration file for certain parameters. When both are available, the environment variable take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

- **HTCC_PASSWORD**—maps to the existing configuration parameter under the `htcc` section, password value.
- **RCS_PASSWORD**— maps to the existing configuration parameter under the `rsc` section, password value.
- **WEBDAV_PASSWORD**—maps to the existing configuration parameter under the `webdav` section, password value.

In a Windows only environment, Recording Muxer Script supports storing all passwords in a secure keystore instead of storing in plain-text in the **`muxer.cfg`** file.

1. From the **`muxer`** directory folder in the Recording Muxer installation folder (for example, **<Recording Muxer Install Directory>\muxer**), execute the following command:
`py encryptPassword.py`
The command will prompt for the appropriate values to be entered for the password/key configuration parameters. See the [Genesys Interaction Recording Options Reference](#) for the descriptions of the parameters.
2. Configure the **`muxer.cfg`** file leaving the following parameter values empty:

```
[webdav]
password =
```

```
[htcc]
password=

[rcs]
password =
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Muxer **muxer.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri		Specifies the host and port of the Interaction Recording Web Services (Web Services) server—for example, https://<web services host>:<web services port>/.
contact_center_id		Specifies the unique identifier of the contact center.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note:</p> <ul style="list-style-type: none">If the "Configuring the Secure Password Storage" step was performed, leave this value empty.The password can be overridden by the HTCC_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is

Parameter Name	Default Value	Description
		corrupted. For more information, see Configuring TLS connection to Interaction Recording Web Services on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.
rws_timeout	30	<p>Specifies the timeout duration, in seconds, for Recording Muxer Script while sending a request to Interaction Recording Web Services.</p> <p>Note: The timeout value must be greater than or equal to 30.</p>

Configuring the Connection to Recording Crypto Server

To configure the connection to the Recording Crypto Server, set the following parameters in the **[rcs]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
base_uri	Empty	Specifies the host and port of the Recording Crypto Server: https://<Recording Crypto Server host>:<Recording Crypto Server port>
username	Empty	<p>Specifies the contact center admin username used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section.</p> <p>Note: The user must have the media decrypt permission.</p>
password	Empty	<p>Specifies the contact center admin password used to access the Recording Crypto Server account belonging to the contact center specified by the the contact_center_id option in the [htcc] section.</p> <p>Note:</p> <ul style="list-style-type: none">• If the Configuring the Secure Password Storage step was performed, leave this value

Parameter Name	Default Value	Description
		empty. <ul style="list-style-type: none">The password can be overridden by the <code>RCS_PASSWORD</code> environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Recording Crypto Server on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Processing Commands

1. The Recording Muxer uses libraries for analyzing and handling multimedia data. To configure these commands, set the following parameters in the **muxer.cfg** file, the **[processing]** section:
 - **ffmpeg** = The path to the ffmpeg executable file.

Important

The ffmpeg executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- **ffprobe** = The path to the ffprobe executable file.

Important

The ffprobe executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

2. To enable Muxer to read multiple screen recordings metadata with one request, configure the following parameters using the **muxer.cfg** configuration file (optional):

- **batch_read_screen_recording_metadata:** Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.
Valid Values: Using Bulk API = 1 / Using previous algorithm the integer ≤ 1
Default Value: 1
- **query_slice_size:** Defines the maximum number of call recording records whose screen recordings should be queried.
Valid Values: all integers > 0
Default Value: 100

3. Configure the **openssl** parameter to set the path to the openssl executable.

Important

- The openssl executable is located under the directory where the thirdparty openssl package was unzipped/untarred.
- On Linux, specifying the absolute path to the openssl executable path is recommended to ensure that the default installed openssl (for example, `/usr/bin/openssl`) is not executed instead.

4. Configure the **window_past** and **window_past_older_than** parameters to set the time in the past to search for the call recordings to multiplex with the screen recordings. See the "Configure HA" section for the recommended values for these parameters.

5. Configure the **clean_temp_folder_timeout** parameter in the **[processing]** section to determine how often the recording files are cleaned up in the **temp folder**. **clean_temp_folder** should only be configured when **auto_clean_temp_folder** is set to 1. By default the **clean_temp_folder** value is 43200 (that is, cleanup occurs every 12 hours). If this value is set to -1, Muxer will attempt to perform a cleanup when it is idle.

For more information about the **[processing]** section parameters, see the [Genesys Interaction Recording Options Reference](#).

Configuring Sharding (Optional)

Sharding can be used to increase the capacity of the Recording Muxer Script solution. When configured, the muxing workload is divided among multiple active instances. By default, Sharding is disabled and `muxer_id = -1`.

When Sharding is in use, a Muxer instance can be configured to run in primary or in backup mode:

- In primary mode, the Muxer should be configured to query for call records from the last `n` minutes (`window_past_older_than=0`, `window_past=n` minutes), based on configuration in the `muxer.cfg` file for that instance.

- In backup mode, the Muxer should be configured to query for call records that are older than the last `n` minutes but newer than `m` minutes (`window_past_older_than= n, window_past= m` minutes), based on configuration in the `muxer.cfg` file for that instance.

Sharding is configured based on the following command line or configuration file parameters within the `[processing]` section:

- **muxer_id:** A unique Muxer ID.
Valid values: A non-negative integer starting with 0 (the Muxer ID should be incremented by 1 for each additional instance).
If you are not using Sharding, the value should be empty or -1.
- **total_muxers:** The total number of primary Muxer instances deployed (excluding the backup).
Valid Values: `max(muxer_id) + 1`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.
- **muxer_type:** indicates if the Muxer is operating in primary mode or backup mode.
Valid Values: `primary, backup`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.

To specify Sharding parameters using the command line, the following arguments are used:

- `muxer-type`
- `muxer-id`
- `total-muxers`

Note: The Sharding parameter values passed in the command line overrides the corresponding values specified within the configuration file. The following is the supported command line:
`python.exe muxer_process.py --config-file=CONFIG_FILE --muxer-type=MUXER_TYPE --muxer-id=MUXER_ID --total-muxers=TOTAL_MUXERS`

For example: When using the following values, the system will have two instances of Muxer running:

- `muxer_type=primary`
- `muxer_id=0` (for the first instance)
- `muxer_id=1` (for the second instance)
- `total_muxers=2`

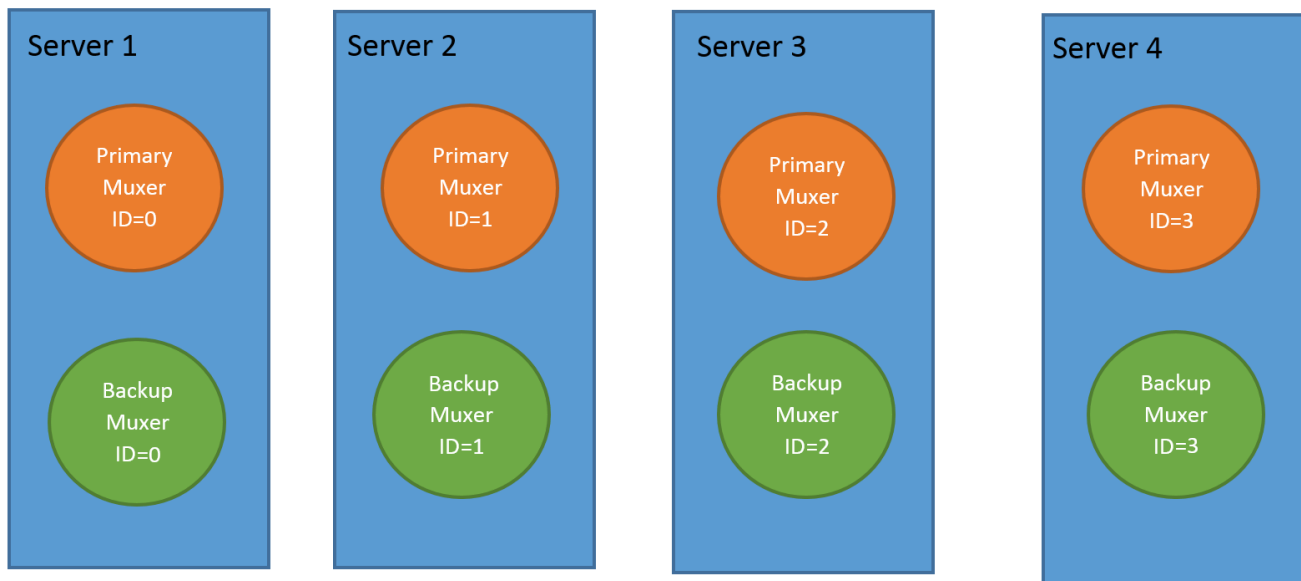
The following is the command line example for running the first instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=0 --total-muxers=2`

The following is the command line example for running the second instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=1 --total-muxers=2`

Note: When there are multiple instances of Muxers deployed on the same machine, then, a different **temp_dir** value for each instance of the Muxer must be configured in the `[processing]` section of the `muxer.cfg` file, and each Muxer instance must use a separate Muxer configuration file. This avoids the issues of one Muxer deleting the temporary files for the other instances.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that each active primary instance be run on a separate machine. For a high availability deployment, a primary instance and a backup instance can be run on the same machine; however, in this case the instances should be configured so that the node IDs overlap (so that a single machine does not provide primary and backup coverage for the same muxer_id).



When running in backup mode, the Muxer will automatically calculate the muxer_id to be used to support this deployment mechanism, based on the specified muxer_id. The configured muxer_id used for the backup instance should match the muxer_id that is configured for the primary instance on the same machine, if both primary mode and backup mode instances are deployed together. For example, if muxer_id=2 and total_muxers=4 in the Muxer configuration file:

- When muxer_type=primary, the muxer_id used will be 2.
- When muxer_type=backup, the muxer_id used will be 3.

Important

If a Muxer instance is added or removed:

- The `total_muxers` value must be changed for each existing Muxer instance.
- All muxer instances must be restarted.
- Before starting the Muxer application, create and configure the **`temp_dir`** and **`logfile_path`** folders for both the Primary Muxer instance and the Backup Muxer instances running on the same machine.

Configuring High Availability (HA)

Important

The content in the Configure HA tab only applies if the Sharding configuration is not in use (see: Configure Sharding (Optional) tab). If Sharding is in use, refer to the high availability configuration described in the Configure Sharding (Optional) tab.

Recording Muxer Cluster

The Recording Muxer Script provides High Availability support using multiple instances of the Recording Muxer Script (all active). HA supports:

- Active/active pairs with the aim to load balance equally between the Recording Muxer nodes by splitting and configuring the time window on each node, so that it is close to equal the number of recordings found on each time window.
- When one of the node dies, recordings are still multiplexed.

Limitations:

- If the node with time window, now - $N/2$, dies, multiplexing will still occur, but a slower rate since the second node's time window is from $N/2$ to N .
- If the node with time window, $N/2$ - N , dies, screen recordings that are uploaded with the delay more than $N/2$ might not be multiplexed.
- Nodes should be configured so that the time windows are exclusive of each other, otherwise it may result in two multiplexed files being uploaded.

To configure HA:

1. In each Recording Muxer's **`muxer.cfg`** configuration file, in the **`[processing]`** section, set the following values for each node. For example,
 - On first node:
 - **`window_past= 720`**
 - **`window_past_older_than = 5`**

2. On second node:

- **window_past** = 1440
- **window_past_older_than** = 725

The above will multiplex all recordings that were recorded within the last 1 day.

3. As a general rule, if the screen recording upload occurs with a delay of N , the configuration on each node can be set to:

- On first node:
 - **window_past** = $N / 2$
 - **window_past_older_than** =
- **min-poll_interval** = $N/200$

4. On second node:

- **window_past** = N
- **window_past_older_than** = $N / 2$
- **min-poll_interval** = $N/200$

Ensure that all Recording Muxer instances have the same configuration other than the above.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that multiple Recording Muxer instances be deployed on different hosts to provide better HA and also not to have machine resource contentions.
- If the recording upload is delayed by more than the time window configured for the Recording Muxer Script, it is possible that the recording will be outside of the processing window and not be multiplexed. For such cases, the Recording Muxer Script can be run as a migration tool to batch process the records matching any desired criteria. For more information see the **call_recording_query_string** parameter under **Configuring the Advanced Options** in the **Advanced Configuration** tab.
- If the screen recording upload is delayed longer than 24 hours, configure a separate Muxer instance or Muxer sharding group for every 12 hours. When the Screen Recording Service is provisioned to upload files during non-business hours, the actual delay can be a couple of days if the agent workstation is shut down when the agent signs off from the Agent Desktop.

Configuring the Connection to WebDAV

To configure the connection to WebDAV, set the following parameters in the **[webdav]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
username	Empty	Specifies the username to allow read/write access to the WebDAV storage server.
password	Empty	<p>Specifies the password to allow read/write access to the WebDAV storage server.</p> <p>Note:</p> <ul style="list-style-type: none">• If multiple WebDAV storage are used for same contact center region, make sure to use the same username and password.• If the "Configuring the Secure Password Storage" step was performed, leave the password value empty.• A password can be overridden by the WEBDAV_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to WebDAV. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to WebDAV on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Advanced Options

The following advanced options can be configured in the **[advanced]** section of the **muxer.cfg** file:

- **worker_threads** = The number of parallel processing threads.
- **pagination** = The maximum number of records returned with each Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) query.

- **max_overlap_allowed** = The overlap time before truncating.
- **video_padding_slice_length_ms** = If the video starts later or ends earlier than the audio, set the duration needed to prepend or append a padded video slice. Genesys recommends to set it to 5000.
- **mark_screen_recording_label** = Whether to apply the label "screenRecording" to the associated call recording metadata after muxing. This configuration is optional. The default value is 1.
- **call_recording_extra_query_string** = Used to specify parameter value pairs other than startTime, endTime, and limit.
If left empty, the **call_recording_extra_query_string** value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if the RWS version is `>= 8.5.201.14`, otherwise, it remains an "" (empty string).
Specify "disable" (without quotes) to force it to be an empty string without checking the RWS version. When the final value of this configuration is not empty, the Recording Muxer Script will continually poll for records that match the searching criteria according to the final value of the configuration that should be processed.
Genesys recommends that this parameter be left empty. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`.
The following table describes values (query parameters) that are available (except startTime and endTime).
- **call_recording_query_string** = When not empty, [call_recording_query_string] queries Interaction Recording Web Services (Web Services) with the given string for records to process. Instead of continually polling for records to process, the Recording Muxer script will exit once the returned records are processed. Genesys recommends that this parameter be left empty unless the Muxer script is to be used for batch migrating the old recordings. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`. The following table describes values (query parameters) that are available:

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified ANI attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard which can substitute any number of any symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified DNIS attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard - which can substitute any number of any symbols in request. Search is case-sensitive.
startTime	Retrieves all recordings that started <code>>=</code> the specified time.
endTime	Retrieves all recordings that ended <code><=</code> the specified time.
userName	Retrieves all recordings in eventHistory->contacts of which the passed userName/ firstName/Lastname is present. User can use wildcards to specify only part of the username/ firstname/lastname. If more than 1 word is used (divided by spaces) -the records containing any of provided terms as username, firstname or lastname will be included. If user wants to

Parameter Name	Description
	retrieve records containing ALL terms - the AND keyword should be used. Sample: ?userName=Alice AND Amber - will seek for recording with events->contact-> username/firstName/ lastName containing Alice and Amber (possible - in different users). Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap. These matches are supported: <ul style="list-style-type: none"> Exact match - match the entire value (for example, "tom" will find "tom"). Wildcarded value (for example, "tom*" will find a record with "tomas"). Combination of matches - If the query terms are separated by spaces (for example, "tom jerry" will look for recordings that contain "tom" or "jerry").

Configuring the Recording Muxer Using Genesys Administrator Extension (Optional)

The Recording Muxer uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Muxer as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Muxer as a "third party server" application in Genesys Administrator Extension. For more information, see the *Using the Management Layer* section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring Recording Muxer Script to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest Recording Muxer script.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Muxer script of type Third Party Server.
4. Create a new application (for example, myRecordingMuxer) in Genesys Administrator Extension using this new application template.
5. On Windows:
 - a. Set the Command Line parameter to the python executable (for example, C:\Python311\python.exe).
 - b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>\muxer directory. For example, C:\Program Files\GCTI\Recording Muxer Script\muxer.
 - d. Set the Command Line Arguments parameter to the python arguments: muxer_process.py -- config-file=muxer.cfg.
6. On Linux:

- a. Set the Command Line parameter to env.
- b. Set the Host parameter in the application's server info to the correct Host object.
- c. Set the Working Directory parameter to the <Recording Muxer Install Directory>/muxer directory. For example, /opt/genesys/Recording_Muxer_Script_8.5/muxer/.
- d. Set the Command Line Arguments parameter. The LD_LIBRARY_PATH must be set to include the openssl binary directory before muxer script execution. For example,
LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:<untarred openssl directory> /opt/python311/python muxer_process.py --config-file=muxer.cfg.

Important

The Recording Muxer does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

Configuring Transport Layer Security (TLS) Connections (Optional)

Python provides the OpenSSL library that is used to establish TLS connections. The OpenSSL library that Python uses is not related to the OpenSSL library installed during installation of third-party libraries, which are used to encrypt muxed recording files.

Configuring TLS connection to Interaction Recording Web Services

1. Set up TLS on Interaction Recording Web Services (RWS). For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the [htcc] section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use https.
3. In the [htcc] section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to true.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, then set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject

alternative name.

Configuring TLS connection to Recording Crypto Server

1. Set up TLS on Recording Crypto Server. For more information, see [Configuring an HTTP Port](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[rcs]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use the secure port.
3. In the **[rcs]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to WebDAV

1. Set up TLS on WebDAV. For more information, see [Configuring TLS for the WebDAV Server](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[webdav]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the

last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

For more information about the Recording Muxer Script parameters, see the [Genesys Interaction Recording Options Reference](#).

Starting the Recording Muxer Script

Important

For **muxer.cfg**, if **temp_dir** is configured, verify that the path exists and is writable by the muxer process.

To launch the Recording Muxer script, run the following command from the <Recording Muxer Install Directory> (where x = 6):

On Windows:

```
<python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

On Linux:

```
env LD_LIBRARY_PATH=<untarred openssl directory>:$LD_LIBRARY_PATH <python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

By default the Recording Muxer's log file is stored in the working directory. This can be changed by specifying a preexisting folder in the **logfile_path** parameter in the **[logfile]** section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingMuxer
```

Recording Muxer Script (Python 3) RHEL 7

Prerequisites

Before installing and configuring the Recording Muxer Script, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) 8.5.205.32 (or higher) instance where the call recording and screen recording metadata is stored.

- A **Recording Crypto Server** 8.5.095.16 (or higher) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage or S3 Premise where the recordings are stored.

Installing Recording Muxer Script

Installing on Windows

1. Install 64-bit Python 3.11.5 from the **Python** website.
2. Install **Recording Muxer Script IP** with the installer.

Note: Install the following third-party libraries in the order they appear and untar the files in Administrator mode.

3. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz file.
4. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1 directory.
5. Untar the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz file.
6. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
7. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
8. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
9. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
10. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
11. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
12. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
13. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
14. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
15. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
16. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
17. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
18. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
19. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.
20. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/

pyasn1-0.5.0 directory.

21. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
22. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
23. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
24. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
25. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
26. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
27. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
28. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
29. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
30. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
31. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
32. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
33. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
34. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
35. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
36. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
37. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
38. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
39. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
40. Run `py -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
41. Unzip the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-win64-static-gpl3.0.zip.
42. Unzip the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-win64.zip. This OpenSSL library is used to encrypt the resulting muxed recording file when required.

Installing on Linux (RHEL)

1. Install zlib-devel (yum install zlib-devel).
2. Install sqlite devel (yum install sqlite-devel.x86_64).

3. Install libffi devel (yum install libffi-devel).
4. Install OpenSSL 1.1.1.
 - For RHEL 7:
 1. Download OpenSSL 1.1.1 from [OpenSSL website](#) and compile it. Example config command - `./config --prefix=/usr/home/openssl-1.1.1 --openssldir=/usr/home/openssl-1.1.1`
 2. Add OpenSSL lib path in LD_LIBRARY_PATH. Example command - `export LD_LIBRARY_PATH=/usr/home/openssl-1.1.1/lib:$LD_LIBRARY_PATH`
5. Install 64 bit Python 3.11.5 compiled with OpenSSL 1.1.1 from the [Python](#) website.
 - While compiling Cpython 3.11.5 with custom openssl, use `--with-openssl` flag while compilation. Example config command - `./configure --with-openssl=/usr/home/openssl-1.1.1 --enable-optimizations`
6. Install the **Recording Muxer Script IP** with the installer provided.

Note: Install the following third-party libraries in the order they appear.

7. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1.tar.gz file.
8. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/jmespath-1.0.1 directory.
9. Untar the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16.tar.gz file.
10. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/urllib3-1.26.16 directory.
11. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.16.0.tar.gz file.
12. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/six-1.16.0 directory.
13. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2.tar.gz file.
14. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.8.2 directory.
15. Untar the <Recording Muxer Install Directory>/thirdparty/idna-3.4.tar.gz file.
16. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/idna-3.4 directory.
17. Untar the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22.tar.gz file.
18. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/certifi-2023.7.22 directory.
19. Untar the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0.tar.gz file.
20. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/charset-normalizer-3.2.0 directory.
21. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0.tar.gz file.
22. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/requests-2.31.0 directory.
23. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0.tar.gz file.

-
24. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.5.0 directory.
 25. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0.tar.gz file.
 26. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/pyasn1_modules-0.3.0 directory.
 27. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36.tar.gz file.
 28. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/botocore-1.31.36 directory.
 29. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2.tar.gz file.
 30. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.6.2 directory.
 31. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36.tar.gz file.
 32. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/boto3-1.28.36 directory.
 33. Untar the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0.tar.gz file.
 34. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/h11-0.14.0 directory.
 35. Untar the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0.tar.gz file.
 36. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/sniffio-1.3.0 directory.
 37. Untar the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0.tar.gz file.
 38. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/anyio-4.0.0 directory.
 39. Untar the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0.tar.gz file.
 40. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpcore-0.18.0 directory.
 41. Untar the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0.tar.gz file.
 42. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/httpx-0.25.0 directory.
 43. Untar the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8.tar.gz file.
 44. Run `python3 -m pip install .` from the <Recording Muxer Install Directory>/thirdparty/webdav4-0.9.8 directory.
 45. Untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-4.4-rhel7-x86_64-static-gpl3.0.tar.bz2.
 46. Execute `chmod a+x ffmpeg` and `chmod a+x ffprobe`.
 47. Untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.1.1l-rhel7-x86_64.tar.bz2.
 48. Execute `chmod a+x openssl`.
-

Important

- GIR does not support direct upgrade of Muxer from Python 2 to Python 3.
- Do not use the `setup.py install` command for installing libraries, instead use `pip install` command as mentioned above.
- Run `sudo dnf install libnsl` if you encounter the following error while executing Muxer installation script (`install.sh`):
./Perl: error while loading shared libraries: libnsl.so.1: cannot open shared object file: No such file or directory.

Configuring Recording Muxer Script

This section describes how to configure the Recording Muxer Script for your environment.

Configure Passwords (Optional)

Important

In a Linux or Windows environment, Muxer supports the use of environment variables instead of parameters in the configuration file for certain parameters. When both are available, the environment variable take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

- **HTCC_PASSWORD**—maps to the existing configuration parameter under the `htcc` section, password value.
- **RCS_PASSWORD**— maps to the existing configuration parameter under the `rccs` section, password value.
- **WEBDAV_PASSWORD**—maps to the existing configuration parameter under the `webdav` section, password value.

In a Windows only environment, Recording Muxer Script supports storing all passwords in a secure keystore instead of storing in plain-text in the **muxer.cfg** file.

1. From the **muxer** directory folder in the Recording Muxer installation folder (for example, **<Recording Muxer Install Directory>\(muxer)**), execute the following command:
`py encryptPassword.py`
The command will prompt for the appropriate values to be entered for the password/key configuration parameters. See the [Genesys Interaction Recording Options Reference](#) for the descriptions of the parameters.

2. Configure the **muxer.cfg** file leaving the following parameter values empty:

```
[webdav]
password =

[htcc]
password=

[rsc]
password =
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Muxer **muxer.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri		Specifies the host and port of the Interaction Recording Web Services (Web Services) server—for example, https://<web services host>:<web services port>/.
contact_center_id		Specifies the unique identifier of the contact center.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note:</p> <ul style="list-style-type: none">If the "Configuring the Secure Password Storage" step was performed, leave this value empty.The password can be overridden by the HTCC_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in

Parameter Name	Default Value	Description
		PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web Services on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.
rws_timeout	30	Specifies the timeout duration, in seconds, for Recording Muxer Script while sending a request to Interaction Recording Web Services. Note: The timeout value must be greater than or equal to 30.

Configuring the Connection to Recording Crypto Server

To configure the connection to the Recording Crypto Server, set the following parameters in the **[rcs]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
base_uri	Empty	Specifies the host and port of the Recording Crypto Server: https://<Recording Crypto Server host>:<Recording Crypto Server port>
username	Empty	Specifies the contact center admin username used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section. Note: The user must have the media decrypt permission.
password	Empty	Specifies the contact center admin password used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section.

Parameter Name	Default Value	Description
		Note: <ul style="list-style-type: none"> If the Configuring the Secure Password Storage step was performed, leave this value empty. The password can be overridden by the RCS_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Recording Crypto Server on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Processing Commands

1. The Recording Muxer uses libraries for analyzing and handling multimedia data. To configure these commands, set the following parameters in the **muxer.cfg** file, the **[processing]** section:
 - **ffmpeg** = The path to the ffmpeg executable file.

Important

The ffmpeg executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- **ffprobe** = The path to the ffprobe executable file.

Important

The ffprobe executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

untarred.

2. To enable Muxer to read multiple screen recordings metadata with one request, configure the following parameters using the **muxer.cfg** configuration file (optional):
 - **batch_read_screen_recording_metadata**: Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.
Valid Values: Using Bulk API = 1 / Using previous algorithm the integer <>1
Default Value: 1
 - **query_slice_size**: Defines the maximum number of call recording records whose screen recordings should be queried.
Valid Values: all integers > 0
Default Value: 100
3. Configure the **openssl** parameter to set the path to the openssl executable.

Important

- The openssl executable is located under the directory where the thirdparty openssl package was unzipped/untarred.
- On Linux, specifying the absolute path to the openssl executable path is recommended to ensure that the default installed openssl (for example, /usr/bin/openssl) is not executed instead.

4. Configure the **window_past** and **window_past_older_than** parameters to set the time in the past to search for the call recordings to multiplex with the screen recordings. See the "Configure HA" section for the recommended values for these parameters.
5. Configure the **clean_temp_folder_timeout** parameter in the **[processing]** section to determine how often the recording files are cleaned up in the **temp folder**. **clean_temp_folder** should only be configured when **auto_clean_temp_folder** is set to 1. By default the **clean_temp_folder** value is 43200 (that is, cleanup occurs every 12 hours). If this value is set to -1, Muxer will attempt to perform a cleanup when it is idle.

For more information about the **[processing]** section parameters, see the [Genesys Interaction Recording Options Reference](#).

Configuring Sharding (Optional)

Sharding can be used to increase the capacity of the Recording Muxer Script solution. When configured, the muxing workload is divided among multiple active instances. By default, Sharding is

disabled and `muxer_id = -1`.

When Sharding is in use, a Muxer instance can be configured to run in primary or in backup mode:

- In primary mode, the Muxer should be configured to query for call records from the last `n` minutes (`window_past_older_than=0, window_past=n` minutes), based on configuration in the `muxer.cfg` file for that instance.
- In backup mode, the Muxer should be configured to query for call records that are older than the last `n` minutes but newer than `m` minutes (`window_past_older_than= n, window_past= m` minutes), based on configuration in the `muxer.cfg` file for that instance.

Sharding is configured based on the following command line or configuration file parameters within the `[processing]` section:

- **muxer_id:** A unique Muxer ID.
Valid values: A non-negative integer starting with 0 (the Muxer ID should be incremented by 1 for each additional instance).
If you are not using Sharding, the value should be empty or -1.
- **total_muxers:** The total number of primary Muxer instances deployed (excluding the backup).
Valid Values: `max(muxer_id) + 1`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.
- **muxer_type:** indicates if the Muxer is operating in primary mode or backup mode.
Valid Values: `primary, backup`
If you are not using Sharding, (indicated by `muxer_id` not being set, or being set to -1), the Muxer ignores this value.

To specify Sharding parameters using the command line, the following arguments are used:

- `muxer-type`
- `muxer-id`
- `total-muxers`

Note: The Sharding parameter values passed in the command line overrides the corresponding values specified within the configuration file. The following is the supported command line:
`python.exe muxer_process.py --config-file=CONFIG_FILE --muxer-type=MUXER_TYPE --muxer-id=MUXER_ID --total-muxers=TOTAL_MUXERS`

For example: When using the following values, the system will have two instances of Muxer running:

- `muxer_type=primary`
- `muxer_id=0` (for the first instance)
- `muxer_id=1` (for the second instance)
- `total_muxers=2`

The following is the command line example for running the first instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=0 --total-muxers=2`

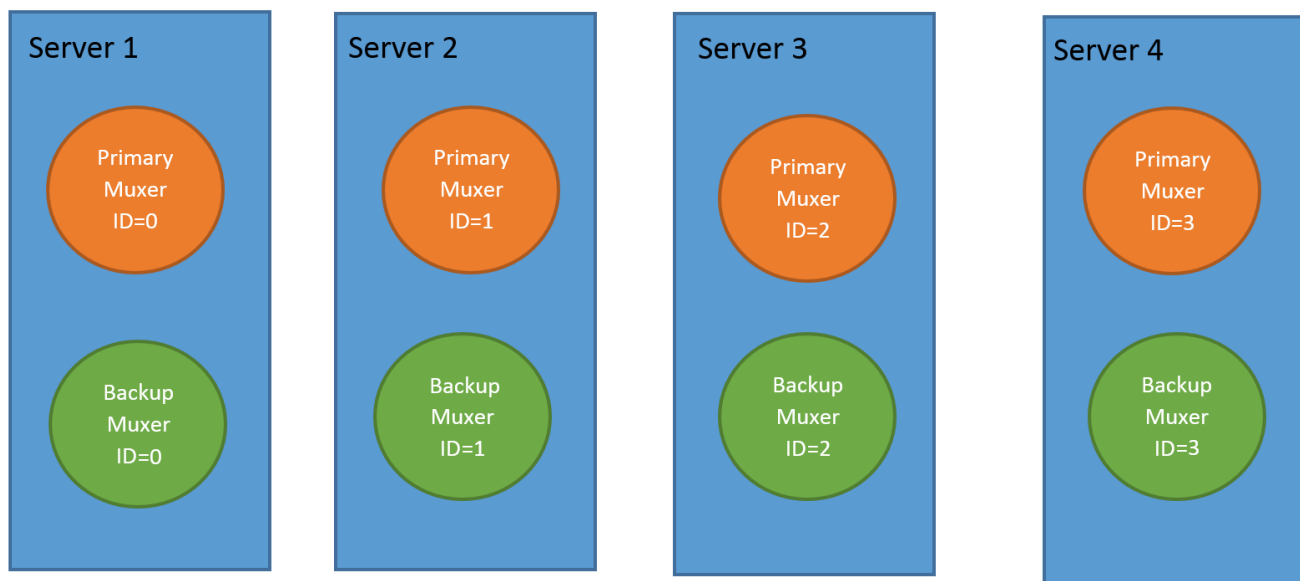
The following is the command line example for running the second instance: `python.exe ../muxer/`

```
muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=1 --total-muxers=2
```

Note: When there are multiple instances of Muxers deployed on the same machine, then, a different **temp_dir** value for each instance of the Muxer must be configured in the [processing] section of the `muxer.cfg` file, and each Muxer instance must use a separate Muxer configuration file. This avoids the issues of one Muxer deleting the temporary files for the other instances.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that each active primary instance be run on a separate machine. For a high availability deployment, a primary instance and a backup instance can be run on the same machine; however, in this case the instances should be configured so that the node IDs overlap (so that a single machine does not provide primary and backup coverage for the same muxer_id).



When running in backup mode, the Muxer will automatically calculate the `muxer_id` to be used to support this deployment mechanism, based on the specified `muxer_id`. The configured `muxer_id` used for the backup instance should match the `muxer_id` that is configured for the primary instance on the same machine, if both primary mode and backup mode instances are deployed together. For example, if `muxer_id=2` and `total_muxers=4` in the Muxer configuration file:

- When `muxer_type=primary`, the `muxer_id` used will be 2.

- When `muxer_type=backup`, the `muxer_id` used will be 3.

Important

If a Muxer instance is added or removed:

- The `total_muxers` value must be changed for each existing Muxer instance.
- All muxer instances must be restarted.
- Before starting the Muxer application, create and configure the **`temp_dir`** and **`logfile_path`** folders for both the Primary Muxer instance and the Backup Muxer instances running on the same machine.

Configuring High Availability (HA)

Important

The content in the Configure HA tab only applies if the Sharding configuration is not in use (see: Configure Sharding (Optional) tab). If Sharding is in use, refer to the high availability configuration described in the Configure Sharding (Optional) tab.

Recording Muxer Cluster

The Recording Muxer Script provides High Availability support using multiple instances of the Recording Muxer Script (all active). HA supports:

- Active/active pairs with the aim to load balance equally between the Recording Muxer nodes by splitting and configuring the time window on each node, so that it is close to equal the number of recordings found on each time window.
- When one of the node dies, recordings are still multiplexed.

Limitations:

- If the node with time window, now - $N/2$, dies, multiplexing will still occur, but a slower rate since the second node's time window is from $N/2$ to N .
- If the node with time window, $N/2 - N$, dies, screen recordings that are uploaded with the delay more than $N/2$ might not be multiplexed.
- Nodes should be configured so that the time windows are exclusive of each other, otherwise it may result in two multiplexed files being uploaded.

To configure HA:

1. In each Recording Muxer's **`muxer.cfg`** configuration file, in the **`[processing]`** section, set the following values for each node. For example,

- On first node:
 - **window_past**= 720
 - **window_past_older_than** = 5
- 2. On second node:
 - **window_past** = 1440
 - **window_past_older_than** = 725

The above will multiplex all recordings that were recorded within the last 1 day.

3. As a general rule, if the screen recording upload occurs with a delay of N , the configuration on each node can be set to:
 - On first node:
 - **window_past** = $N / 2$
 - **window_past_older_than** =
 - **min-poll_interval** = $N/200$
4. On second node:
 - **window_past**= N
 - **window_past_older_than** = $N / 2$
 - **min-poll_interval** = $N/200$

Ensure that all Recording Muxer instances have the same configuration other than the above.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that multiple Recording Muxer instances be deployed on different hosts to provide better HA and also not to have machine resource contentions.
- If the recording upload is delayed by more than the time window configured for the Recording Muxer Script, it is possible that the recording will be outside of the processing window and not be multiplexed. For such cases, the Recording Muxer Script can be run as a migration tool to batch process the records matching any desired criteria. For more information see the **call_recording_query_string** parameter under **Configuring the Advanced Options** in the **Advanced Configuration** tab.
- If the screen recording upload is delayed longer than 24 hours, configure a separate

Muxer instance or Muxer sharding group for every 12 hours. When the Screen Recording Service is provisioned to upload files during non-business hours, the actual delay can be a couple of days if the agent workstation is shut down when the agent signs off from the Agent Desktop.

Configuring the Connection to WebDAV

To configure the connection to WebDAV, set the following parameters in the **[webdav]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
username	Empty	Specifies the username to allow read/write access to the WebDAV storage server.
password	Empty	<p>Specifies the password to allow read/write access to the WebDAV storage server.</p> <p>Note:</p> <ul style="list-style-type: none">• If multiple WebDAV storage are used for same contact center region, make sure to use the same username and password.• If the "Configuring the Secure Password Storage" step was performed, leave the password value empty.• A password can be overridden by the WEBDAV_PASSWORD environment variable.
trusted-ca	false	<p>Configures TLS certificate validation when making a secure outbound connection to WebDAV. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to WebDAV on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.</p>

Configuring the Advanced Options

The following advanced options can be configured in the **[advanced]** section of the **muxer.cfg** file:

- **worker_threads** = The number of parallel processing threads.
- **pagination** = The maximum number of records returned with each Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) query.
- **max_overlap_allowed** = The overlap time before truncating.
- **video_padding_slice_length_ms** = If the video starts later or ends earlier than the audio, set the duration needed to prepend or append a padded video slice. Genesys recommends to set it to 5000.
- **mark_screen_recording_label** = Whether to apply the label "screenRecording" to the associated call recording metadata after muxing. This configuration is optional. The default value is 1.
- **call_recording_extra_query_string** = Used to specify parameter value pairs other than startTime, endTime, and limit.
If left empty, the **call_recording_extra_query_string** value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if the RWS version is `>= 8.5.201.14`, otherwise, it remains an "" (empty string).
Specify "disable" (without quotes) to force it to be an empty string without checking the RWS version. When the final value of this configuration is not empty, the Recording Muxer Script will continually poll for records that match the searching criteria according to the final value of the configuration that should be processed.
Genesys recommends that this parameter be left empty. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`.
The following table describes values (query parameters) that are available (except startTime and endTime).
- **call_recording_query_string** = When not empty, [call_recording_query_string] queries Interaction Recording Web Services (Web Services) with the given string for records to process. Instead of continually polling for records to process, the Recording Muxer script will exit once the returned records are processed. Genesys recommends that this parameter be left empty unless the Muxer script is to be used for batch migrating the old recordings. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`. The following table describes values (query parameters) that are available:

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified ANI attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard which can substitute any number of any symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified DNIS attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard - which can substitute any number of any symbols in request. Search is case-sensitive.
startTime	Retrieves all recordings that started <code>>=</code> the specified time.
endTime	Retrieves all recordings that ended <code><=</code> the

Parameter Name	Description
	specified time.
userName	Retrieves all recordings in eventHistory->contacts of which the passed userName/ firstName/Lastname is present. User can use wildcards to specify only part of the username/ firstname/lastname. If more than 1 word is used (divided by spaces) -the records containing any of provided terms as username, firstname or lastname will be included. If user wants to retrieve records containing ALL terms - the AND keyword should be used. Sample: ?userName=Alice AND Amber - will seek for recording with events->contact-> username/firstName/ lastName containing Alice and Amber (possible - in different users). Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap. These matches are supported: <ul style="list-style-type: none">• Exact match - match the entire value (for example, "tom" will find "tom").• Wildcarded value (for example, "tom*" will find a record with "tomas").• Combination of matches - If the query terms are separated by spaces (for example, "tom jerry" will look for recordings that contain "tom" or "jerry").

Configuring the Recording Muxer Using Genesys Administrator Extension (Optional)

The Recording Muxer uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Muxer as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Muxer as a "third party server" application in Genesys Administrator Extension. For more information, see the *Using the Management Layer* section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring Recording Muxer Script to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest Recording Muxer script.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Muxer script of type Third Party Server.
4. Create a new application (for example, myRecordingMuxer) in Genesys Administrator Extension using this new application template.
5. On Windows:
 - a. Set the Command Line parameter to the python executable (for example, C:\Python311\python.exe).

- b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>\muxer directory. For example, C:\Program Files\GCTI\Recording Muxer Script\muxer.
 - d. Set the Command Line Arguments parameter to the python arguments: muxer_process.py --config-file=muxer.cfg.
6. On Linux:
- a. Set the Command Line parameter to env.
 - b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>/muxer directory. For example, /opt/genesys/Recording_Muxer_Script_8.5/muxer/.
 - d. Set the Command Line Arguments parameter. The LD_LIBRARY_PATH must be set to include the openssl binary directory before muxer script execution. For example, LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:<untarred openssl directory> /opt/python311/python muxer_process.py --config-file=muxer.cfg.

Important

The Recording Muxer does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

Configuring Transport Layer Security (TLS) Connections (Optional)

Python provides the OpenSSL library that is used to establish TLS connections. The OpenSSL library that Python uses is not related to the OpenSSL library installed during installation of third-party libraries, which are used to encrypt muxed recording files.

Configuring TLS connection to Interaction Recording Web Services

1. Set up TLS on Interaction Recording Web Services (RWS). For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[htcc]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use https.
3. In the **[htcc]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to true.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the

certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, then set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to Recording Crypto Server

1. Set up TLS on Recording Crypto Server. For more information, see [Configuring an HTTP Port](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[rcs]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use the secure port.
3. In the **[rcs]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to WebDAV

1. Set up TLS on WebDAV. For more information, see [Configuring TLS for the WebDAV Server](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[webdav]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set

trusted_ca to true.

- If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

For more information about the Recording Muxer Script parameters, see the [Genesys Interaction Recording Options Reference](#).

Starting the Recording Muxer Script

Important

For **muxer.cfg**, if **temp_dir** is configured, verify that the path exists and is writable by the muxer process.

To launch the Recording Muxer script, run the following command from the <Recording Muxer Install Directory> (where x = 6):

On Windows:

```
<python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

On Linux:

```
env LD_LIBRARY_PATH=<untarred openssl directory>:$LD_LIBRARY_PATH <python3.11.5 executable> muxer_process.py --config-file=muxer.cfg
```

By default the Recording Muxer's log file is stored in the working directory. This can be changed by specifying a preexisting folder in the **logfile_path** parameter in the **[logfile]** section of the configuration file. For example, in Windows:

```
logfile_path = C:\logs\recordingMuxer
```

Recording Muxer Script Legacy (Python 2) Deprecated

Important

Recording Muxer Script Legacy (based on Python 2) has been discontinued as of March 31, 2024.

Prerequisites

Before installing and configuring the Recording Muxer Script, you must have the following prerequisites:

- An [Interaction Recording Web Services](#) (or [Web Services](#) if you're using version 8.5.210.02 or earlier) instance where the call recording and screen recording metadata is stored.
- A [Recording Crypto Server](#) instance to decrypt the encrypted recordings.
- Network access to the WebDAV storage where the recordings are stored.

Installing Recording Muxer Script

Installing on Windows

1. Install 32 bit Python 2.7.x from the [Python](#) website.
2. Install the Recording Muxer Script IP.

Note: Install the following third party libraries in the order they appear.

3. Untar the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2.tar.gz file.
4. From the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2 directory, run `python setup.py install`.
5. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1.tar.gz file.
6. From the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1 directory, run `python setup.py install`.
7. Untar the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1.tar.gz file.
8. From the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1 directory, run `python setup.py install`.
9. Untar the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0.tar.gz file.
10. From the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0 directory, run `python setup.py install`.
11. Untar the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5.tar.gz file.
12. From the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5 directory, run

```
python setup.py install.
```

13. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7.tar.gz file.
14. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7 directory, run python setup.py install.
15. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5.tar.gz file.
16. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5 directory, run python setup.py install.
17. Unzip the <Recording Muxer Install Directory>/thirdparty/ffmpeg/ffmpeg-2.4.3-win64-static-gpl3.0.zip.
18. Unzip the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.0.2j-win64.zip. This OpenSSL library is used to encrypt the resulting muxed recording file when required.

Important

The following steps are only applicable for Muxer 8.5.265.66 or higher.

19. Untar the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1.tar.gz file.
20. From the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1 directory, run python setup.py install.
21. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.10.0.tar.gz file.
22. From the <Recording Muxer Install Directory>/thirdparty/six-1.10.0 directory, run python setup.py install.
23. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0.tar.gz file.
24. From the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0 directory, run python setup.py install.
25. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1.tar.gz file.
26. From the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1 directory, run python setup.py install.
27. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57.tar.gz file.
28. From the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57 directory, run python setup.py install.
29. Untar the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5.tar.gz file.
30. From the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5 directory, run python setup.py install.
31. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10.tar.gz file.
32. From the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10 directory, run python setup.py install.
33. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0.tar.gz file.
34. From the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0 directory, run python setup.py install.

Installing on Linux (RHEL)

1. Install Python 2.7.6 or later:

- Download the software from the [Python](#) website.
It is recommend that newer versions of Python are installed separately from an existing versions (do not update).

2. Install the Recording Muxer Script IP.

Note: Install the following third party libraries in the order they appear.

3. Untar the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2.tar.gz file.
4. From the <Recording Muxer Install Directory>/thirdparty/setuptools-1.3.2 directory, run `python setup.py install`.
5. Untar the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1.tar.gz file.
6. From the <Recording Muxer Install Directory>/thirdparty/requests-2.4.1 directory, run `python setup.py install`.
7. Untar the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1.tar.gz file.
8. From the <Recording Muxer Install Directory>/thirdparty/boto-2.32.1 directory, run `python setup.py install`.
9. Untar the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0.tar.gz file.
10. From the <Recording Muxer Install Directory>/thirdparty/easywebdav-1.2.0 directory, run `python setup.py install`.
11. Untar the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5.tar.gz file.
12. From the <Recording Muxer Install Directory>/thirdparty/filechunkio-1.5 directory, run `python setup.py install`.
13. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7.tar.gz file.
14. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-0.1.7 directory, run `python setup.py install`.
15. Untar the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5.tar.gz file.
16. From the <Recording Muxer Install Directory>/thirdparty/pyasn1-modules-0.0.5 directory, run `python setup.py install`.
17. Untar the <Recording Muxer Install Directory>/thirdparty/ffmpeg/
ffmpeg-2.4.3-centos5-x86_64-static-gpl3.0.tar.bz2.
18. Execute `chmod a+x ffmpeg` and `chmod a+x ffmpegprobe`.
19. Untar the <Recording Muxer Install Directory>/thirdparty/openssl/openssl-1.0.2j-centos5-x86_64.tar.bz2. This OpenSSL library is used to encrypt the resulting muxed recording file when required.
20. Execute `chmod a+x openssl`.

Important

The following steps are only applicable for Muxer 8.5.265.66 or higher.

21. Untar the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1.tar.gz file.
22. From the <Recording Muxer Install Directory>/thirdparty/docutils-0.13.1 directory, run `python setup.py install`.
23. Untar the <Recording Muxer Install Directory>/thirdparty/six-1.10.0.tar.gz file.
24. From the <Recording Muxer Install Directory>/thirdparty/six-1.10.0 directory, run `python setup.py install`.
25. Untar the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0.tar.gz file.
26. From the <Recording Muxer Install Directory>/thirdparty/python-dateutil-2.6.0 directory, run `python setup.py install`.
27. Untar the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1.tar.gz file.
28. From the <Recording Muxer Install Directory>/thirdparty/jmespath-0.9.1 directory, run `python setup.py install`.
29. Untar the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57.tar.gz file.
30. From the <Recording Muxer Install Directory>/thirdparty/botocore-1.4.57 directory, run `python setup.py install`.
31. Untar the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5.tar.gz file.
32. From the <Recording Muxer Install Directory>/thirdparty/futures-3.0.5 directory, run `python setup.py install`.
33. Untar the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10.tar.gz file.
34. From the <Recording Muxer Install Directory>/thirdparty/s3transfer-0.1.10 directory, run `python setup.py install`.
35. Untar the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0.tar.gz file.
36. From the <Recording Muxer Install Directory>/thirdparty/boto3-1.4.0 directory, run `python setup.py install`.

Upgrading Recording Muxer Script

1. Backup the Recording Muxer Script installation directory including logs and configuration file.
2. Uninstall the Recording Muxer Script component.
3. Install the new Recording Muxer Script component.
4. Update the Recording Muxer Script configuration according to the standard installation procedures.

Important

Uninstalling the previous Recording Muxer Script is optional.

Configuring Recording Muxer Script

This section describes how to configure the Recording Muxer Script for your environment.

Configure Passwords (Optional)

Important

In a Linux or Windows environment, Muxer supports the use of environment variables instead of parameters in the configuration file for certain parameters. When both are available, the environment variable take precedence.

The following definitions describe the mapping of the environment variables to the corresponding configuration parameter:

- **HTCC_PASSWORD**—maps to the existing configuration parameter under the htcc section, password value.
- **RCS_PASSWORD**— maps to the existing configuration parameter under the rcs section, password value.
- **WEBDAV_PASSWORD**—maps to the existing configuration parameter under the webdav section, password value.

In a Windows only environment, Recording Muxer Script supports storing all passwords in a secure keystore instead of storing in plain-text in the **muxer.cfg** file.

1. From the **muxer** directory folder in the Recording Muxer installation folder (for example, **<Recording Muxer Install Directory>\muxer**), execute the following command:
`python encryptPassword.py`
The command will prompt for the appropriate values to be entered for the password/key configuration parameters. See the [Genesys Interaction Recording Options Reference](#) for the descriptions of the parameters.
2. Configure the **muxer.cfg** file leaving the following parameter values empty:

```
[webdav]
password =

[htcc]
password=

[rsc]
password =
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

To configure the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection, set the following parameters in the **[htcc]** section of the Recording Muxer **muxer.cfg** configuration file:

Parameter Name	Default Value	Description
base_uri		Specifies the host and port of the Interaction Recording Web Services (Web Services) server—for example, <code>https://<web_services_host>:<web_services_port>/</code> .
contact_center_id		Specifies the unique identifier of the contact center.
username	ops	Specifies the username used to access the Interaction Recording Web Services (Web Services) account.
password	ops	<p>Specifies the password used to access the Interaction Recording Web Services (Web Services) account.</p> <p>Note:</p> <ul style="list-style-type: none">• If the "Configuring the Secure Password Storage" step was performed, leave this value empty.• The password can be overridden by the <code>HTCC_PASSWORD</code> environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are <code>true</code> , <code>false</code> , and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web Services on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Parameter Name	Default Value	Description
rws_timeout	30	<p>Specifies the timeout duration, in seconds, for Recording Muxer Script while sending a request to Interaction Recording Web Services.</p> <p>Note: The timeout value must be greater than or equal to 30.</p>

Configuring the Connection to Recording Crypto Server

To configure the connection to the Recording Crypto Server, set the following parameters in the **[rcs]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
base_uri	Empty	<p>Specifies the host and port of the Recording Crypto Server: https://<Recording Crypto Server host>:<Recording Crypto Server port></p>
username	Empty	<p>Specifies the contact center admin username used to access the Recording Crypto Server account belonging to the contact center specified by the contact_center_id option in the [htcc] section.</p> <p>Note: The user must have the media decrypt permission.</p>
password	Empty	<p>Specifies the contact center admin password used to access the Recording Crypto Server account belonging to the contact center specified by the the contact_center_id option in the [htcc] section.</p> <p>Note:</p> <ul style="list-style-type: none">• If the Configuring the Secure Password Storage step was performed, leave this value empty.• The password can be overridden by the RCS_PASSWORD environment variable.

Parameter Name	Default Value	Description
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Recording Crypto Server on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Processing Commands

1. The Recording Muxer uses libraries for analyzing and handling multimedia data. To configure these commands, set the following parameters in the **muxer.cfg** file, the **[processing]** section:

- **ffmpeg** = The path to the ffmpeg executable file.

Important

The ffmpeg executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

- **ffprobe** = The path to the ffprobe executable file.

Important

The ffprobe executable is located under the directory where the thirdparty ffmpeg package was unzipped/untarred.

2. To enable Muxer to read multiple screen recordings metadata with one request, configure the following parameters using the **muxer.cfg** configuration file (optional):
 - **batch_read_screen_recording_metadata**: Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.
Valid Values: Using Bulk API = 1 / Using previous algorithm the integer <>1
Default Value: 1
 - **query_slice_size**: Defines the maximum number of call recording records whose screen recordings should be queried.

Valid Values: all integers > 0
Default Value: 100

3. Configure the **openssl** parameter to set the path to the openssl executable.

Important

- The openssl executable is located under the directory where the thirdparty openssl package was unzipped/untarred.
- On Linux, specifying the absolute path to the openssl executable path is recommended to ensure that the default installed openssl (for example, /usr/bin/openssl) is not executed instead.

4. Configure the **window_past** and **window_past_older_than** parameters to set the time in the past to search for the call recordings to multiplex with the screen recordings. See the "Configure HA" section for the recommended values for these parameters.
5. Configure the **clean_temp_folder_timeout** parameter in the **[processing]** section to determine how often the recording files are cleaned up in the **temp folder**. **clean_temp_folder** should only be configured when **auto_clean_temp_folder** is set to 1. By default the **clean_temp_folder** value is 43200 (that is, cleanup occurs every 12 hours). If this value is set to -1, Muxer will attempt to perform a cleanup when it is idle.

For more information about the **[processing]** section parameters, see the [Genesys Interaction Recording Options Reference](#).

Configuring Sharding (Optional)

Sharding can be used to increase the capacity of the Recording Muxer Script solution. When configured, the muxing workload is divided among multiple active instances. By default, Sharding is disabled and `muxer_id = -1`.

When Sharding is in use, a Muxer instance can be configured to run in primary or in backup mode:

- In primary mode, the Muxer should be configured to query for call records from the last n minutes (`window_past_older_than=0, window_past=n minutes`), based on configuration in the `muxer.cfg` file for that instance.
- In backup mode, the Muxer should be configured to query for call records that are older than the last n minutes but newer than m minutes (`window_past_older_than= n, window_past= m minutes`), based on configuration in the `muxer.cfg` file for that instance.

Sharding is configured based on the following command line or configuration file parameters within the **[processing]** section:

- **muxer_id:** A unique Muxer ID.
Valid values: A non-negative integer starting with 0 (the Muxer ID should be incremented by 1 for each additional instance).
If you are not using Sharding, the value should be empty or -1.
- **total_muxers:** The total number of primary Muxer instances deployed (excluding the backup).
Valid Values: $\max(\text{muxer_id}) + 1$
If you are not using Sharding, (indicated by muxer_id not being set, or being set to -1), the Muxer ignores this value.
- **muxer_type:** indicates if the Muxer is operating in primary mode or backup mode.
Valid Values: primary, backup
If you are not using Sharding, (indicated by muxer_id not being set, or being set to -1), the Muxer ignores this value.

To specify Sharding parameters using the command line, the following arguments are used:

- muxer-type
- muxer-id
- total-muxers

Note: The Sharding parameter values passed in the command line overrides the corresponding values specified within the configuration file. The following is the supported command line:
`python.exe muxer_process.py --config-file=CONFIG_FILE --muxer-type=MUXER_TYPE --muxer-id=MUXER_ID --total-muxers=TOTAL_MUXERS`

For example: When using the following values, the system will have two instances of Muxer running:

- muxer_type=primary
- muxer_id=0 (for the first instance)
- muxer_id=1 (for the second instance)
- total_muxers=2

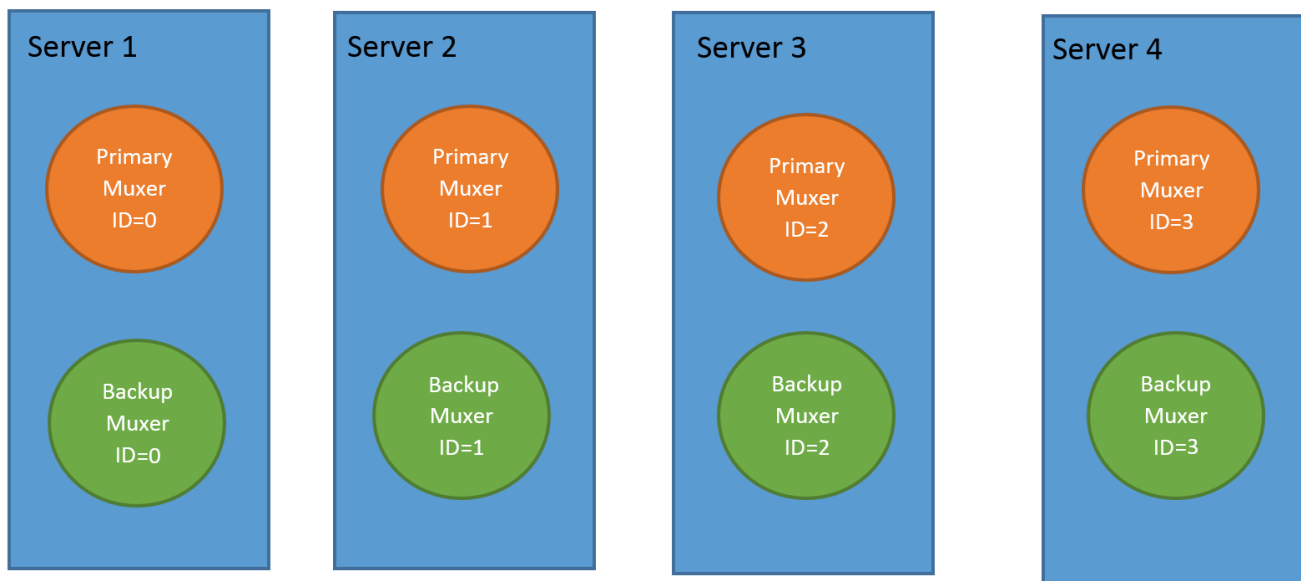
The following is the command line example for running the first instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=0 --total-muxers=2`

The following is the command line example for running the second instance: `python.exe ../muxer/muxer_process.py --config-file=muxer.cfg --muxer-type=primary --muxer-id=1 --total-muxers=2`

Note: When there are multiple instances of Muxers deployed on the same machine, then, a different **temp_dir** value for each instance of the Muxer must be configured in the [processing] section of the muxer.cfg file, and each Muxer instance must use a separate Muxer configuration file. This avoids the issues of one Muxer deleting the temporary files for the other instances.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that each active primary instance be run on a separate machine. For a high availability deployment, a primary instance and a backup instance can be run on the same machine; however, in this case the instances should be configured so that the node IDs overlap (so that a single machine does not provide primary and backup coverage for the same muxer_id).



When running in backup mode, the Muxer will automatically calculate the muxer_id to be used to support this deployment mechanism, based on the specified muxer_id. The configured muxer_id used for the backup instance should match the muxer_id that is configured for the primary instance on the same machine, if both primary mode and backup mode instances are deployed together. For example, if muxer_id=2 and total_muxers=4 in the Muxer configuration file:

- When muxer_type=primary, the muxer_id used will be 2.
- When muxer_type=backup, the muxer_id used will be 3.

Important

If a Muxer instance is added or removed:

- The `total_muxers` value must be changed for each existing Muxer instance.
- All muxer instances must be restarted.
- Before starting the Muxer application, create and configure the **temp_dir** and **logfile_path** folders for both the Primary Muxer instance and the Backup Muxer instances running on the same machine.

Configuring High Availability (HA)

Important

The content in the Configure HA tab only applies if the Sharding configuration is not in use (see: Configure Sharding (Optional) tab). If Sharding is in use, refer to the high availability configuration described in the Configure Sharding (Optional) tab.

Recording Muxer Cluster

The Recording Muxer Script provides High Availability support using multiple instances of the Recording Muxer Script (all active). HA supports:

- Active/active pairs with the aim to load balance equally between the Recording Muxer nodes by splitting and configuring the time window on each node, so that it is close to equal the number of recordings found on each time window.
- When one of the node dies, recordings are still multiplexed.

Limitations:

- If the node with time window, now - $N/2$, dies, multiplexing will still occur, but a slower rate since the second node's time window is from $N/2$ to N .
- If the node with time window, $N/2 - N$, dies, screen recordings that are uploaded with the delay more than $N/2$ might not be multiplexed.
- Nodes should be configured so that the time windows are exclusive of each other, otherwise it may result in two multiplexed files being uploaded.

To configure HA:

1. In each Recording Muxer's **muxer.cfg** configuration file, in the **[processing]** section, set the following values for each node. For example,
 - On first node:
 - **window_past**= 720
 - **window_past_older_than** = 5

2. On second node:

- **window_past** = 1440
- **window_past_older_than** = 725

The above will multiplex all recordings that were recorded within the last 1 day.

3. As a general rule, if the screen recording upload occurs with a delay of N , the configuration on each node can be set to:

- On first node:
 - **window_past** = $N / 2$
 - **window_past_older_than** =
- **min-poll_interval** = $N/200$

4. On second node:

- **window_past** = N
- **window_past_older_than** = $N / 2$
- **min-poll_interval** = $N/200$

Ensure that all Recording Muxer instances have the same configuration other than the above.

Important

- Genesys recommends that the maximum window length configured in each Recording Muxer Script instance be 12 hours (720 minutes). That is, the difference between the **window_past** and **window_past_older_than** parameters should be a maximum of 720 minutes. If the window length is greater than 12 hours, the configuration may cause problems with Elasticsearch.
- Genesys recommends that multiple Recording Muxer instances be deployed on different hosts to provide better HA and also not to have machine resource contentions.
- If the recording upload is delayed by more than the time window configured for the Recording Muxer Script, it is possible that the recording will be outside of the processing window and not be multiplexed. For such cases, the Recording Muxer Script can be run as a migration tool to batch process the records matching any desired criteria. For more information see the **call_recording_query_string** parameter under **Configuring the Advanced Options** in the **Advanced Configuration** tab.
- If the screen recording upload is delayed longer than 24 hours, configure a separate Muxer instance or Muxer sharding group for every 12 hours. When the Screen Recording Service is provisioned to upload files during non-business hours, the actual delay can be a couple of days if the agent workstation is shut down when the agent signs off from the Agent Desktop.

Configuring the Connection to WebDAV

To configure the connection to WebDAV, set the following parameters in the **[webdav]** section of the Recording Muxer **muxer.cfg** file:

Parameter Name	Default Value	Description
username	Empty	Specifies the username to allow read/write access to the WebDAV storage server.
password	Empty	<p>Specifies the password to allow read/write access to the WebDAV storage server.</p> <p>Note:</p> <ul style="list-style-type: none">• If multiple WebDAV storage are used for same contact center region, make sure to use the same username and password.• If the "Configuring the Secure Password Storage" step was performed, leave the password value empty.• A password can be overridden by the WEBDAV_PASSWORD environment variable.
trusted-ca	false	Configures TLS certificate validation when making a secure outbound connection to WebDAV. Valid values are true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. Muxer will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to WebDAV on the Configuring Transport Layer Security (TLS) Connections (Optional) tab.

Configuring the Advanced Options

The following advanced options can be configured in the **[advanced]** section of the **muxer.cfg** file:

- **worker_threads** = The number of parallel processing threads.
- **pagination** = The maximum number of records returned with each Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) query.

- **max_overlap_allowed** = The overlap time before truncating.
- **video_padding_slice_length_ms** = If the video starts later or ends earlier than the audio, set the duration needed to prepend or append a padded video slice. Genesys recommends to set it to 5000.
- **mark_screen_recording_label** = Whether to apply the label "screenRecording" to the associated call recording metadata after muxing. This configuration is optional. The default value is 1.
- **call_recording_extra_query_string** = Used to specify parameter value pairs other than startTime, endTime, and limit.
If left empty, the **call_recording_extra_query_string** value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if the RWS version is `>= 8.5.201.14`, otherwise, it remains an "" (empty string).
Specify "disable" (without quotes) to force it to be an empty string without checking the RWS version. When the final value of this configuration is not empty, the Recording Muxer Script will continually poll for records that match the searching criteria according to the final value of the configuration that should be processed.
Genesys recommends that this parameter be left empty. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`.
The following table describes values (query parameters) that are available (except startTime and endTime).
- **call_recording_query_string** = When not empty, [call_recording_query_string] queries Interaction Recording Web Services (Web Services) with the given string for records to process. Instead of continually polling for records to process, the Recording Muxer script will exit once the returned records are processed. Genesys recommends that this parameter be left empty unless the Muxer script is to be used for batch migrating the old recordings. Query parameters have to be formatted as: `<parameter name>=<value>[&<parameter name>=<value>...]`. The following table describes values (query parameters) that are available:

Parameter Name	Description
callerPhoneNumber	Retrieves all recordings which apply to any call containing the specified ANI attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard which can substitute any number of any symbols in the request. Search is case-sensitive.
dialedPhoneNumber	Retrieves all recordings which apply to any call containing the specified DNIS attribute. The exact match of stored number (alphanumeric-only) and request parameter (alphanumeric-only) is used. The request string can contain * wildcard - which can substitute any number of any symbols in request. Search is case-sensitive.
startTime	Retrieves all recordings that started <code>>=</code> the specified time.
endTime	Retrieves all recordings that ended <code><=</code> the specified time.
userName	Retrieves all recordings in eventHistory- >contacts of which the passed userName/ firstName/Lastname is present. User can use wildcards to specify only part of the username/ firstname/lastname. If more than 1 word is used (divided by spaces) -the records containing any of provided terms as username, firstname or lastname will be included. If user wants to

Parameter Name	Description
	retrieve records containing ALL terms - the AND keyword should be used. Sample: ?userName=Alice AND Amber - will seek for recording with events->contact-> username/firstName/ lastName containing Alice and Amber (possible - in different users). Search is case-insensitive.
userData	Retrieves all recordings in eventHistory->data of which the passed userData is present as value of HashMap. These matches are supported: <ul style="list-style-type: none">• Exact match - match the entire value (for example, "tom" will find "tom").• Wildcarded value (for example, "tom*" will find a record with "tomas").• Combination of matches - If the query terms are separated by spaces (for example, "tom jerry" will look for recordings that contain "tom" or "jerry").

Configuring the Recording Muxer Using Genesys Administrator Extension (Optional)

The Recording Muxer uses a configuration file instead of a specific application object in Configuration Server. However, it is possible to configure the Recording Muxer as a "third-party server" application enabling Genesys Administrator Extension to monitor, start, and stop the process.

The following steps describe how to setup Recording Muxer as a "third party server" application in Genesys Administrator Extension. For more information, see the *Using the Management Layer* section of the [Framework 8.5.1 Management Layer User's Guide](#)

Configuring Recording Muxer Script to Start/Stop via LCA using Genesys Administrator Extension:

1. Install and deploy the latest Recording Muxer script.
2. Make sure that the Local Control Agent (LCA) is running.
3. Create a new application template in Genesys Administrator Extension called Recording Muxer script of type Third Party Server.
4. Create a new application (for example, myRecordingMuxer) in Genesys Administrator Extension using this new application template.
5. On Windows:
 - a. Set the Command Line parameter to the python executable (for example, C:\Python27\python.exe).
 - b. Set the Host parameter in the application's server info to the correct Host object.
 - c. Set the Working Directory parameter to the <Recording Muxer Install Directory>\muxer directory. For example, C:\Program Files\GCTI\Recording Muxer Script\muxer.
 - d. Set the Command Line Arguments parameter to the python arguments: muxer_process.py -- config-file=muxer.cfg.
6. On Linux:

- a. Set the Command Line parameter to env.
- b. Set the Host parameter in the application's server info to the correct Host object.
- c. Set the Working Directory parameter to the <Recording Muxer Install Directory>/muxer directory. For example, /opt/genesys/Recording_Muxer_Script_8.5/muxer/.
- d. Set the Command Line Arguments parameter. The LD_LIBRARY_PATH must be set to include the openssl binary directory before muxer script execution. For example,
LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:<untarred openssl directory> /opt/python27/python muxer_process.py --config-file=muxer.cfg.

Important

The Recording Muxer does not support configuration through Genesys Administrator Extension. Configuration is acquired using a local configuration file.

Configuring Transport Layer Security (TLS) Connections (Optional)

Python provides the OpenSSL library that is used to establish TLS connections. To use a newer version of OpenSSL, upgrade the version of Python being used (within the 2.7.x family). The OpenSSL library that Python uses is not related to the OpenSSL library installed during installation of third-party libraries, which are used to encrypt muxed recording files.

Configuring TLS connection to Interaction Recording Web Services

1. Set up TLS on Interaction Recording Web Services (RWS). For more information, see [Configuring TLS on the Server-Side for Interaction Recording Web Services](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the [htcc] section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use https.
3. In the [htcc] section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to true.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, then set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject

alternative name.

Configuring TLS connection to Recording Crypto Server

1. Set up TLS on Recording Crypto Server. For more information, see [Configuring an HTTP Port](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[rcs]** section of the Recording Muxer Script configuration file, set the **base_uri** parameter to use the secure port.
3. In the **[rcs]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to `false`. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

Configuring TLS connection to WebDAV

1. Set up TLS on WebDAV. For more information, see [Configuring TLS for the WebDAV Server](#) section. For information on acquiring TLS certificates and private keys, see [Genesys Security Deployment Guide](#).
2. In the **[webdav]** section of the Recording Muxer Script configuration file, configure the **trusted_ca** parameter as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted_ca** to `true`.
 - If the TLS certificate was issued by a certificate authority, set **trusted_ca** to the path of the CA certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then append all certificates in the chain into a single file. All the files containing certificates must be in PEM format. The file should have the certificates in order of lowest in the chain to the root of the chain. The root certificate authority should be the

last certificate listed in the file. Set **trusted_ca** to the path to this file.

- If the TLS certificate is a self-signed certificate, set **trusted_ca** to the path of the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted_ca** to false. If certificate verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name.

For more information about the Recording Muxer Script parameters, see the [Genesys Interaction Recording Options Reference](#).

Starting the Recording Muxer Script

Important

For **muxer.cfg**, if **temp_dir** is configured, verify that the path exists and is writable by the muxer process.

To launch the Recording Muxer script, run the following command from the <Recording Muxer Install Directory> (where x = 6):

On Windows:

```
<python2.7.x executable> muxer_process.py --config-file=muxer.cfg
```

On Linux:

```
env LD_LIBRARY_PATH=<untarred openssl directory>:$LD_LIBRARY_PATH <python2.7.x executable> muxer_process.py --config-file=muxer.cfg
```

By default the Recording Muxer's log file is stored in the working directory. This can be changed by specifying a preexisting folder in the **logfile_path** parameter in the **[logfile]** section of the configuration file. For example:

```
logfile_path = C:\logs\recordingMuxer
```

Deploying SpeechMiner for GIR

GIR uses the SpeechMiner application to play back call and screen recordings that are stored in the GIR system. Note that call or screen recordings that have been backed up and then purged from the GIR system cannot be played back through SpeechMiner. These should be played with your own media player.

Installing SpeechMiner

To install and configure SpeechMiner, follow the instructions in the [Deploying SpeechMiner](#) topic of the *SpeechMiner Administration Guide*. Pay attention to the instructions specific to Analytics and Recording UI, or Recording UI Only modes.

Important

The UConnector service is not required for GIR.

Important

To access this content:

- **Customers:** Log in to [My Support](#) and select *Documentation*.
- **Partners:** Log in to [Partner Portal](#) and select *Genesys Technical Docs*.
- **Employees:** Go to the [internal access point](#).

Upgrading SpeechMiner

For information about upgrading your SpeechMiner components, see the [SpeechMiner 8.5.x Upgrade Guide](#).

The SpeechMiner documents are restricted and require specific credentials to access the content. If you have a login, you can access the docs here: [Login](#).

If you are a Genesys employee:

- To access the online documents and if you don't have an account on this site, click the **Employee Login / Restricted Content** link in the lower-right corner of the page. Create your login and send your username to Tech Pubs Admins so that we can grant you access.

If you are a Genesys Partner or customer:

- Contact the product manager to access to these documents.

If you are upgrading SpeechMiner, you must upload and deploy the Solution Definition (SPD) files in order to gain access to any new Genesys Administrator Extension roles or privileges.

For more information about uploading the SPD file, see the [Upgrading the Plug-in](#).

Configuring SpeechMiner users

In addition to the configuration that is described in the *SpeechMiner Administration Guide*, the SpeechMiner database and application must be configured specifically for Genesys Interaction Recording (GIR).

1. Use Genesys Administrator Extension to create four new Application Templates:
 - a. Import the following templates from the SpeechMiner CD:
 - Speechminer_ClientApplications.apd
 - Speechminer_InteractionReceiver.apd
 - Speechminer_Platform.apd
 - Speechminer_Web.apd
 - b. Verify that each template has Genesys Generic Server in the **Type** field.
 - c. Verify that the log parameters are defined.
2. Create four new Application objects using the newly created templates.
 - a. Enter the names in the **Name** field as they are indicated in the template. For example, name SpeechMiner Platform application, Speechminer_Platform, where SpeechMiner is the default and can be changed in the SMConfig Recording panel. The name used and the name in the SMConfig Recording panel should match.
 - b. Set the hosts.
 - c. The Start info Working Directory, Command Line must not be empty. Set it to ". ".

Important

The SpeechMiner components do not integrate with LCA.

3. Use Genesys Administrator Extension to create an additional new Application Template:

- a. Import the following template from the SpeechMiner CD:
 - `Speechminer_node.apd`
 - b. Verify that the template has Genesys Generic Client in the **Type** field.
4. Create three new Application objects.
- a. In the **Name** field, enter the name of each application object. The three new Application objects should be named as follows: (where SpeechMiner is the default and can be changed in the SMConfig Recording panel. The name used and the name in the SMConfig Recording panel should match):
 - `Speechminer_Platform_Node`
 - `Speechminer_InteractionReceiver_Node`
 - `Speechminer_Web_Node`
 - b. Create a connection for each Application object to the Server application with the similar name. For example, for *Speechminer_Web_Node* use the name *Speechminer_Web*.
5. In **Genesys Administrator Extension**, navigate to **Configuration > Environment > Tenants** and in the **Options** tab of each Tenant (including the Environment tab), in the **[recording.archive]** section configure the following parameters:
- user
 - password

Important

The user and password value must be the same as the username and password configured in both of the following sections:

- **Configuring SpeechMiner Interaction Receiver Authorization Header** in the **Recording Destinations** section of **IVR profile configuration**.
- **Configuring SpeechMiner settings** in RWS.

6. In the SMConfig-Login screen (SpeechMiner Configuration), login with your Configuration Server credentials (for example, default/password) and set the Configuration Server host and port.

Important

When you are logging in for the first time, you must go to the license tab, and add your recording-only license, and login to SM Config again. You will see the **Recording** tab in the UI only after you have added the license.

7. In the Sites and Machines/Machines and Tasks screen:
- Configure the Interaction Receiver tasks.

- Click **Select Languages** and select **English USA**.

8. Perform the following:

- Configure the **User Application Name** so that it refers to the application you want to use for authenticating users. By default, it is **default** and should only be changed in a multi-tenant environment, for security reasons. For details, refer to the [Configuring SpeechMiner users](#) section in the Permissions page.
- Enable the following tasks in the SpeechMiner configuration (Recording Only mode):
 - web server
 - interaction receiver
 - indexer
 - uplatform

9. In the Media panel of SMConfig, set the following to avoid creating unnecessary audio files and storing them for too long (Analytics Only mode):

Parameter	Value
Recognition Audio Format	WAV_PCM
Create compressed audio file	Do not Generate
WAV_PCM Retention Period	0

10. In the Recording tab, set the parameters.

[+] Show the table that describes the parameters.

Section	Parameter	Description	Example
Configuration	Tenant	Specifies the tenant as configured in Genesys Administrator Extension. Note: For single-tenant contact centers, the Tenant should match the tenant used in the configuration server.	Resources
	User and Password	The Configuration Server user and password that SpeechMiner applications should use when connecting to the Configuration Server. Verify that the specific user was given read and execute permissions for the tenant object in the configuration server and all its objects in the tenant object	

Section	Parameter	Description	Example
		hierarchy.	
	Application Name	Specifies the prefix of the SpeechMiner application objects as configured in Genesys Administrator Extension.	SpeechMiner
	User Application Name	The name of the Configuration Manager application object that will be used to validate user credentials.	
	Update Agents Every	Specifies how often to update the agent tree in the user interface.	24 hours
Interaction Receiver	Default Program	Specifies the Program ID to be used if the Recording Processor Script does not assign a Program ID to the call.	default
	Extension Speaker Type	Specifies the type of speaker to be used for the extension side of the call.	agent
	Trunk Speaker Type	Specifies the type of speaker to be used for the trunk side of the call.	customer
RP Authorization Note: The values of RP Authorization User and Password must match the values that are configured in the SpeechMiner Interaction Receiver Authorization Header in the IVR Profile.	User	Specifies the username used by the Recording Processor Script when posting metadata to the SpeechMiner Interaction Receiver.	<rp_username>
	Password	Specifies the password used by the Recording Processor Script when posting metadata to the SpeechMiner Interaction Receiver.	<rp_password>
MCP Authorization Note: The values of MCP Authorization User and Password must match the values that are configured in the SpeechMiner HTTP Authorization Header in the IVR Profile. Leave these parameters empty unless you have purchased and enabled speech analytics	User	Specifies the username used by the MCP when posting files to the SpeechMiner Interaction Receiver.	<username>
	Password	Specifies the password used by the MCP when posting files to the SpeechMiner Interaction Receiver.	<password>

Section	Parameter	Description	Example
mode on SpeechMiner.			
Playback	RCS URI	Specifies the URI that the SpeechMiner Web Service component uses to communicate with the Recording Crypto Server as a server-to-server connection. This parameter is used for playback of call recordings.	http://<Recording Crypto Server Host>:<port>/rcs or https://<Recording Crypto Server Host>:<port>/rcs
	RWS URI	Specifies the URI that the SpeechMiner Web Service component uses to communicate with Interaction Recording Web Services as a server-to-server connection. This parameter must be set to use the tagging or deletion protection functionality. Note: You must disable CSRF protection functionality in RWS if you are using tagging or deletion protection.	http://<Interaction Recording Web Services host>:<port> or https://<Interaction Recording Web Services host>:<port>
	External RCS URI	Specifies the URI that the SpeechMiner browser application uses to access the Recording Crypto Server. Note: This parameter is required to playback encrypted screen recordings, unless you have configured the local decrypt URI prefix (refer to Local Decrypt URI Prefix for Call Recording and Screen Recording).	http://<Recording Crypto Server Host>:<port>/rcs or https://<Recording Crypto Server Host>:<port>/rcs
	External RWS URI	Specifies the URI that the SpeechMiner browser application uses to access Interaction Recording Web Services. Note: This parameter is required for screen recording playback only. If you do not	http://<Interaction Recording Web Services host>:<port> or https://<Interaction Recording Web Services host>:<port>

Section	Parameter	Description	Example
		want to use screen recording, leave this value blank.	

11. Depending on the license you are using and only after you have verified that the license matches the installation mode, perform one of the following:
- In **Recording and Analytics** mode, create and apply a program in SMART for every Program ID that RP may send.

Important

Both of these actions can be skipped, if the default in the database is satisfactory, or if the Recording Processor Script includes the program ID in the metadata.

On a per-call basis, the attached data key `GRECORD_PROGRAM` can be set to define the program external ID to be used for this call. For example, attached data can be set in a routing strategy.

12. Configure the roles and permissions for the **SpeechMiner Users**.

Deploying Workspace Desktop Edition for GIR

You can use the Workspace Desktop Edition (formerly known as Interaction Workspace) for GIR as the agent's desktop.

Installing Workspace Desktop Edition

To install and configure the Workspace Desktop Edition, see the [Workspace Desktop Edition 8.5 Deployment Guide](#). You can learn more about the desktop [here](#).

Configuring Workspace Desktop Edition

In addition to the configuration described in the deployment guide, you must configure the Workspace Desktop Edition as follows:

1. Set the MSML recording parameters:

Parameter Name	Value
<code>active-recording.voice.recorder-uri</code>	Leave empty. The file recording destination is configured through the GVP IVR Profile.
<code>active-recording.voice.recording-type</code>	MSML

2. Configure the desktop to attach data:

- In the `interaction-workspace` section, set `interaction.case-data.format-business-attribute=CaseData` where `CaseData` is the name of the Business Attributes object that contains a list of attributes that are the attached data keys.

To allow integration with the Screen Recording Client, see:

[Integrating with Workspace Desktop Edition](#)

Configuring permissions, access control, and privacy

The following sections describe, and provide examples of how to configure access control for Genesys Interaction Recording Users.

For more information about controlling the access for voice recording users, see [Access Control for Voice Recording Users](#).

Configuring SpeechMiner roles and permissions

Configuring SpeechMiner users

All SpeechMiner users must be assigned to the Users Access Group. If agent hierarchy and partition features are not used, assign all the SpeechMiner users to the / (slash) Access Group. If agent hierarchy or partition features are used, the users must be granted to the specific Access Groups in order to be able to access recordings for the various agent hierarchy and partitions.

Important

- To restrict log-in to the SpeechMiner UI, a new Configuration Manager application object must be created. Backup the default Configuration Manager object, since this object is accessible by all users from all tenants. The new Configuration Manager application object should be configured to allow Environment administrators, Environment users and Super administrators access to it.
- To see members in the User Access Group (by default, SpeechMiner Users) in the Speechminer UI, Log On As the account of Speechminer_WEB application should have Read rights to User Access Group.

You must configure Genesys Interaction Recording to enable the SpeechMiner UI search option to display a list of agent names:

1. In the Agent's **Person** object, create a **[recording]** section in the **Annex** (if it doesn't already exist).
2. Add the **agent_hierarchy** option in the **[recording]** section, and set the value to slash: "/" or what is appropriate for access control.
3. Repeat these steps for any additional agents that might be searched for in the SpeechMiner UI.
4. This configuration will not take effect until the SpeechMiner cache is updated:
 - In the **SMConfig > Recording** tab, update the **Update Agents Every** parameter to the number of hours between the SpeechMiner person object updates. SpeechMiner will check the Configuration

Server according to this option to retrieve the list of person objects under the **Recording folder** access group. The names of these agents are then available when searching for call recordings or screen recordings.

- To force the list of agents to update sooner, update the **NextAgentsUpdate** column in the **configServer** table of the SpeechMiner database to a date in the near future.

Important

- The Access Group / (forward slash) grants access to all recordings.

The following is a screen shot showing the assignment of Access Group members to /Anthony/Paul in Genesys Administrator Extension:

The screenshot shows the Genesys Administrator Extension interface. The top navigation bar includes GAX, System Dashboard, Agents, Configuration (selected), Routing Parameters, and Administration. The breadcrumb trail is Home > Access Groups > Access Groups > Recording > /Anthony/Paul Properties. On the right, there are buttons for Clone, Delete, and Move To. The left sidebar has tabs for General, Members, Options, Permissions, and Dependencies. The main content area is titled 'Persons' and contains a table with columns: Name, First Name, Last Name, and Agent. The table has one row with the following data: Name (agent1000), First Name (Firstname1000), Last Name (lastname), and Agent (checked). A red arrow points to the 'Members' tab in the left sidebar. Another red arrow points to the 'Save' button at the bottom right of the interface.

	Name	First Name	Last Name	Agent
<input checked="" type="checkbox"/>	agent1000	Firstname1000	lastname	<input checked="" type="checkbox"/>

The Recording Plug-in for GAX includes a **Solution Definition (SPD) file** that can be used to configure roles and access groups.

Configuring roles

For information about configuring roles for Genesys Interaction Recording users, see [Role Privileges](#) in the Genesys Administrator Extension Deployment Guide.

Configuring permissions for recording labels

A label definition defines a label, which can then be applied to a recording. For example, a label definition could be created to mark a recording for further review.

Permissions are required to perform these operations. You can configure the label permissions using Genesys Administrator Extension (GAX), in the IRWS_Cluster (or WS_Cluster, where applicable) application object or the Person object.

To configure label permissions, do this:

1. If you're configuring this at the application level: add a new recording settings group to the Annex/ Application options group for the IRWS_Cluster (or WS_Cluster where applicable) application object, or update the existing recording group. For details, refer to [Genesys Administrator Extension Help: Configuration Manager](#) and [Installing Interaction Recording Web Services](#).

Important

You are not required to do it this way; you can also set this at the Person object level.

2. Configure one or all of the following options in the recording settings as follows:

```
[recording]
RECORDING_PERMISSION_ADD_LABEL_DEFINITION = true
RECORDING_PERMISSION_DELETE_LABEL_DEFINITION = true
RECORDING_PERMISSION_ADD_LABEL = true
RECORDING_PERMISSION_DELETE_LABEL = true
```

The system applies permissions in the following order:

1. Default settings that are defined in the application code, which are overridden by:
2. Settings that are specified in the Application, which are overridden by:
3. Settings that are specified in the Person object that corresponds to the agent.

The following permissions are required to allow users with the role supervisor or agent to be able to access and use the different label operations.

Permission	Description	Applies to	Checks against
RECORDING_PERMISSION_ADD_LABEL_DEFINITION	Permission to create a label definition	<ul style="list-style-type: none"> Creating a label definition Updating a label definition 	<ul style="list-style-type: none"> Supervisor Agent
RECORDING_PERMISSION_DELETE_LABEL_DEFINITION	Permission to delete a label definition	<ul style="list-style-type: none"> Deleting a label definition 	<ul style="list-style-type: none"> Supervisor Agent
RECORDING_PERMISSION_ADD_LABEL	Permission to add/Update label(s) on a recording	<ul style="list-style-type: none"> Adding a label to a recording Updating a label on a recording Adding a label to multiple recordings 	<ul style="list-style-type: none"> Supervisor Agent
RECORDING_PERMISSION_DELETE_LABEL	Permission to delete label(s) from a recording	<ul style="list-style-type: none"> Deleting a label from a recording 	<ul style="list-style-type: none"> Supervisor Agent

Configuring Permissions for Recording Non-Deletion

You can protect recordings from deletion using SpeechMiner, or using the [Recording Non-Deletion API](#), if you have the appropriate permissions that are required.

You can configure the non-deletion permissions using Genesys Administrator Extension (GAX), in the Configuration Manager view, the IRWS_Cluster (or WS_Cluster where applicable) application object or the Person object. Contact center administrators have full access by default.

To configure non-deletion permissions, do this:

1. If you're configuring this at the application level: add a new recording settings group to the Annex/ Application options group for the IRWS_Cluster (or WS_Cluster, where applicable) application object, or update the existing recording group. For details, refer to [Genesys Administrator Extension Help: Configuration Manager](#) and [Installing Interaction Recording Web Services](#).

Important

You are not required to do it this way; you can also set this at the Person object level.

2. Configure one or both of the following options in the recording settings as follows:

```
[recording]
RECORDING_PERMISSION_APPLY_NON_DELETE = true
RECORDING_PERMISSION_UNAPPLY_NON_DELETE = true
```

The system applies permissions in the following order:

1. Default settings that are defined in the application code, which are overridden by:
2. Settings that are specified in the Application, which are overridden by:
3. Settings that are specified in the Person object that corresponds to the agent.

The following permissions are required to allow users with the role of supervisor or agent to be able to access and use the different non-deletion operations.

Permission	Description	Applies to	Checks against
RECORDING_PERMISSION_APPLY_NON_DELETE	Permission to protect a recording from being deleted	Apply Non-Deletion to a Recording	<ul style="list-style-type: none">• Supervisor• Agent
RECORDING_PERMISSION_UNAPPLY_NON_DELETE	Permission to remove deletion protection from a recording	Remove Non-Deletion from a Recording	<ul style="list-style-type: none">• Supervisor• Agent

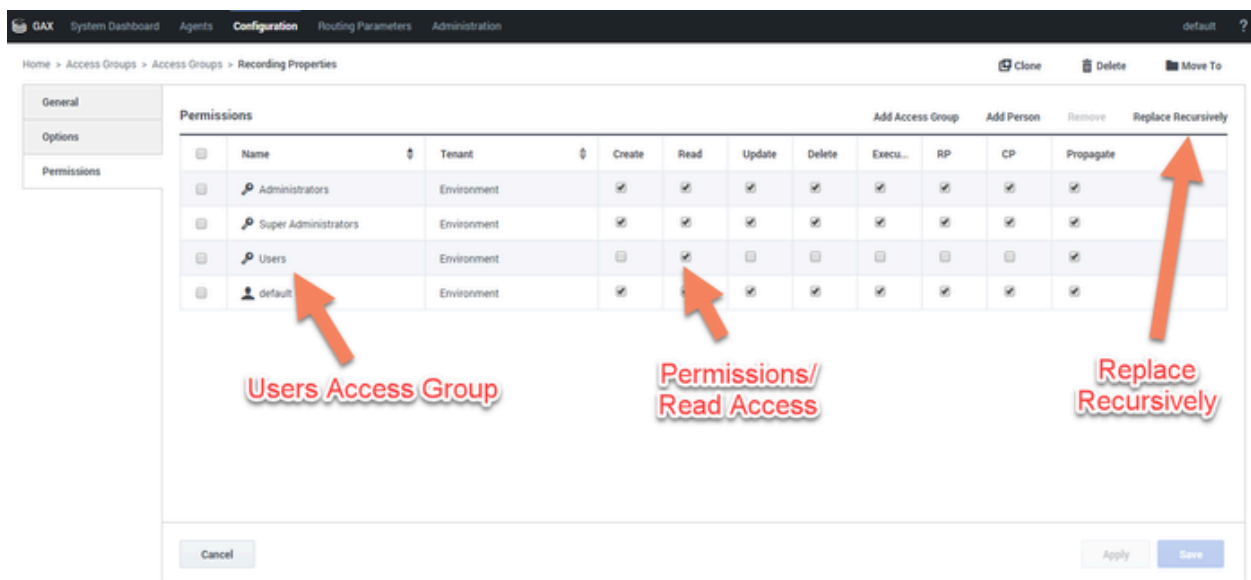
Configuring access control and agent hierarchy

Configuring access groups

By default, the Configuration Server has an **Access Group** called Users stored in the configuration database.

Install the Solution Deployment SPD file "Creation of base access groups" option to perform the following steps:

1. Create an **Access Group**, and set the permission to grant the users in the Access Group with Read access.
2. Add a new folder within Access Groups, called **Recording**, and set the permission to add the Users Access Group with Read access. Make sure the Replace Permissions Recursively action is set as shown in the following diagram:



3. Create the / (forward slash) Access Group within the **Recording** folder.

Important

If this **User Access Group** exists in more than one tenant, use unique naming conventions; otherwise, the users will not appear in the SpeechMiner UI.

Configuring partitions

For each partition used in the contact, create an **Access Group** object with the name of the partition within the **Recording** folder. For example, if there are three partitions— /sales, /support, and /marketing, create three **Access Group** objects named /sales, /support, and /marketing, respectively.

Important

Access Group names for partitions and agent hierarchy must be unique for each tenant.

Configuring agent hierarchy

Agent hierarchy and partitions are not required to record calls or access recordings; however, all agents must be assigned to the Users Access Group.

If agent hierarchy is required, assign the agent's hierarchy by configuring the `agent_hierarchy` option in the recording section of the Person object's Annex tab. For each hierarchy name, create a corresponding **Access Group** object within the **Recording** folder.

For the example above, create the following **Access Groups**:

- /
- /Anthony
- /Anthony/John
- /Anthony/Paul

Important

The `agent_hierarchy` field for a user should not include that user's name. For example, David's `agent_hierarchy` can be:

- /Genesys/Tel Aviv/
Not:
- /Genesys/Tel Aviv/David

Every user can only be part of one hierarchy (a single path) in the entire hierarchy tree. For example:

- If the hierarchy for David is /Genesys/Toronto, then John's hierarchy cannot be /Genesys/Tel Aviv/David. That is, David cannot be a part of two different hierarchies.
- /Genesys/Tel Aviv/BE and /Genesys/Toronto/BE should not exist in the same hierarchy tree. But, /Genesys/BE/Tel Aviv and /Genesys/BE/Toronto can exist in the same hierarchy tree.

Configuring user access control

Agents and users can be seen by a logged in user based on the logged in user's read permissions to the agents and users Person objects in the Configuration Server. Additionally, access to items within SpeechMiner (for example, Forms, Evaluations, Reports and so on), is also limited based on read permissions to the creator of those items.

To limit which agents and users can be seen by a logged in user you must set **AccessControlEnabled** to **1** (true) in the **ConfigServer** table in the SpeechMiner Database (that is, the database selected during the SpeechMiner installation).

Important

If **AccessControlEnabled** is not set to true, all users can see and access all agents and users items within SpeechMiner.

Configuring sensitive data privileges

Sensitive information (for example, credit card numbers, telephone numbers, home addresses and so on) can be hidden from agents when stored in the system.

To configure sensitive data privileges:

1. Add a new Recording settings group to the Annex/Application options group for the GIR cluster application object. For details, refer to ["Genesys Administrator Extension User Guide > Configuration Manager"](#)
2. Configure one or both of the following options in the Recording settings group created in step #1:
 - **metadata.privacy.agent_fields:** Add a comma-separated value of all the metadata fields that must be hidden if the user does not have permission to view the agent metadata fields. For example, callerPhoneNumber, dialedPhoneNumber, dnis, ani, agentId, username, phoneNumber, username, firstName, lastName, GSIP_RECORD, and so on.
 - **metadata.privacy.customer_fields:** Add a comma-separated value of all the metadata fields that must be hidden if the user does not have permission to view the customer metadata fields. For example, firstName, lastName, and so on.

Important

Metadata fields with angle brackets or backslashes are not supported.

With the following privileges you can view recording metadata fields that are usually masked from unauthorized users:

- **Customer Sensitive Data:** This privilege enables the user to display customer-sensitive data in the SpeechMiner GUI. When this privilege is enabled, the data is visible.
- **Agent Sensitive Data:** This privilege enables the user to display agent-sensitive data in the SpeechMiner GUI. When this privilege is enabled, the data is visible.

For more information on how to configure the above privileges, refer to [Configuring Roles and Privileges in GAX](#).

Important

- Both the Customer Sensitive Data privilege and the Agent Sensitive Data privilege will not affect report results. That is, sensitive data will be included in reports. If you do not want sensitive data to be included in reports you must disable the relevant report.

For more information about configuring Access Controls in Genesys Administrator Extension, see the [Genesys Administrator Extension User Guide](#).

Secure Transport Configuration

This section describes how to configure Transport Layer Security (TLS) for the Genesys Interaction Recording solution.

Server-Side Configuration

The following components must configure secure transports for HTTP.

Interaction Recording Web Services

Configuring TLS for Interaction Recording Web Services

See [Configuring TLS on the Server Side for Interaction Recording Web Services](#).

Configuring TLS for the Recording Processor Script

1. Configure HTTPS on the primary recording server. For more information, see the "Configure SSL" section of [Configuring Recording Processor Script](#).
 - a. For Windows, make sure the pyOpenSSL is installed. pyOpenSSL is already be installed on RHEL6.
 - b. Create a self-signed certificate and private key for the Recording Processor host. For example, on Ubuntu run:

```
openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem
```
 - c. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—Point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228.pem`.
 - d. Send the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section of the [GVP 8.5 User's Guide](#).
 - e. Restart Recording Processor.
2. Configure HTTPS on the backup recording server by following the same instructions as above using a new certificate and private key.

Configuring TLS for the Voice Processor

See [Voice Processor Service Level Configuration](#).

Configuring TLS for the Recording Crypto Server

See [Configure HTTP Port](#) tab in the [Configuring Recording Crypto Server](#) section.

Configuring TLS for the WebDAV Server

See [Configuring TLS for the WebDAV Server](#).

Configuring TLS for the Interaction Receiver and SpeechMiner UI Server

See [Enabling HTTPS for SpeechMiner](#).

Configuring TLS for the HTTP Load Balancer

See [Configuring TLS for the HTTP Load Balancer](#) in a single-tenant environment.
See [Configuring TLS for the HTTP Load Balancer](#) in a multi-tenant environment.

Client-Side Configuration

Configuring TLS for the Media Control Platform (MCP)

To add a Certificate Authority (CA):

1. Place the CA file on the MCP.
2. Using Genesys Administrator or Genesys Administrator Extension, in the **[fm]** section set the **ssl_ca_info** option to the location of the CA file.
3. Restart MCP.

To add client-side authentication:

1. Place the certificate file (PEM format) on the MCP.
2. Using Genesys Administrator or Genesys Administrator Extension, in the **[fm]** section set the **ssl_cert** option to the location of the certification file.
3. Restart MCP.

For more information about the MCP options, see the [Voice Platform Media Control Platform Configuration Options](#).

Configuring TLS for the IVR Profile

Using Genesys Administrator Extension, navigate to the Recording tab of the IVR Profile. Update the following addresses with the HTTPS locations:

- Storage Destination
- Recording Processor URI

- SpeechMiner Interaction Receiver
- SpeechMiner Destination for Analytics only

Configuring TLS for the Recording Processor Script

The Recording Processor Script creates three client connections, to:

- Interaction Recording Web Services (Web Services)
- SpeechMiner Interaction Receiver
- Backup Recording Processor Script

For details on configuring each connection, refer to the appropriate section at the [Configure SSL](#) link on the page [Deploying Recording Processor Script](#).

Configuring TLS for the Voice Processor

The Voice Processor creates three client connections, to:

- Interaction Recording Web Services (Web Services)
- SpeechMiner Interaction Receiver
- Genesys Info Mart

For details on configuring these connections, see [Configuring Voice Processor](#).

Configuring TLS for Interaction Recording Web Services

Interaction Recording Web Services (RWS) may be configured to use secure connections to the following components:

- Configuration Server
- SIP Server
- Interaction Server
- WebDAV
- Recording Crypto Server
- SpeechMiner Interaction Receiver
- Cassandra

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Security](#).

Configuring TLS for the Recording Muxer Script

The Recording Muxer Script creates client connections to the following:

- Interaction Recording Web Services
- Recording Crypto Server (if the recordings are encrypted)
- WebDAV

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Transport Layer Security \(TLS\) Connections](#).

Configuring TLS for the Recording Crypto Server

The Recording Crypto Server creates client connections to the following:

- Interaction Recording Web Services
- SpeechMiner Interaction Receiver
- Message Server
- Configuration Server

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Transport Layer Security \(TLS\) Connections](#).

Configuring TLS for the Recording Plug-in for GAX

See [Configuring Transport Layer Security](#).

Configuring Media Lifecycle Management

When it is time to purge old recording files, Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) requires additional configuration to allow the records to purge and/or backup successfully. For instance, when it is time to purge old recording files, Interaction Recording Web Services (Web Services) sends a purge request to the SpeechMiner database indicating which records to delete.

Interaction Recording Web Services (Web Services)

To enable Genesys Interaction Recording to purge and back up recording files, configure the Interaction Recording Web Services (Web Services) node as follows:

In the **backgroundScheduledMediaOperationsSettings** section of the **serverSettings** section within the **application.yaml** file:

- Set **enableBackgroundScheduledMediaOperations** to true
- Set **defaultBackupExportURI** to a backup folder—for example, `file:///tmp/archLocDefault` is the default backup folder.
- Set **enableScanAndScroll** to true to enable the Scan and Scroll feature of Elasticsearch. The default value is false.

Important

Verify that the SIP Server is running before accessing the MLM configuration. If the SIP Server is not running, a **User not authorized** message will appear when you try to access MLM in GAX.

For more information about these options, see the [Advanced Settings for the MLM API](#).

In the **recordingSettings** section of the **serverSettings** section within the **application.yaml** file, set **auditLogDeletedFiles** to true if you want to log all deleted recordings in the audit log when they are purged.

For more information about these options, see [Configuring the Call Recording Audit Log](#).

SpeechMiner

To enable Interaction Recording Web Services (Web Services) to contact Interaction Receiver and purge the requested recordings, use a text editor to add the following to the location based setting group in the `json.txt` file:


```
{
  "name": "interaction-receiver",
  "location": "/US/CA",
  "value": {
    "uri-prefix": "http://speechminer-server/interactionreceiver",
    "userName": "interaction receiver user name",
    "password": "interaction receiver password"
  }
}
```

Important

- <http://speechminer-server/interactionreceiver> is the Load Balancer URL that points to the Interaction Receiver.
- The Interaction Receiver user name and password must be the same as the **user** and **password** property values found within the **[recording.archive]** section of the tenant Annex in the configuration, and are set when configuring Recording Crypto Server. If these values are not found there, they should be added.

Execute the following command:

```
curl -u <user:password> -X POST -d @json.txt --header
"Content-Type: application/json" http://<Web Services-cluster-address>/api/v2/settings/
speechminer
```

Important

The username and password provided in the above command line must be associated with a user defined in the Genesys configuration environment.

For more information on the properties of this settings group, see [Interaction Recording Web Services Settings Groups](#).

For more information about the location based setting group for encryption, see [Encrypting and Provisioning Certificates](#).

Creating Rules and Schedules

Use Genesys Administrator Extension to create rules and schedules. For step-by-step instructions, see [Recording Lifecycle Scheduler](#).

Consider the following when creating backup and purge tasks:

- Do not schedule backup tasks to run concurrently on the same Interaction Recording Web Services (Web Services) node if these tasks back up overlapping records.

- Do not schedule backup and purge tasks to run concurrently if they act on overlapping records.
- Ensure that all the Interaction Recording Web Services (Web Services) nodes have accurate clocks.
- Genesys Administrator Extension's time is based on UTC.

Important

- When using the MLM Purge feature, if you specify a rule for voice recordings, the corresponding muxed screen recordings will not be purged unless you also select the **Include Screen Recordings** checkbox.
- Recordings that are protected from deletion (using the Non-Deletion API or SpeechMiner) will not be deleted by Media Lifecycle Management purge tasks.
- Do not schedule a purge task to run independently in its own rule unless you are willing to lose the associated data. Even if a backup has been scheduled, it is not guaranteed to complete successfully before the purge task is executed.
- Review the [Recording Lifecycle Scheduler Parameters](#) page in the Genesys Interaction Recording Help. It contains important details about the rule parameters, such as any limitations you should be aware of.

Warning

When you are scheduling rules containing purge tasks, adhere to the following guidelines to avoid an unexpected failure of Purge or Backup tasks:

- Run only *one* Purge task in a rule.
- Run the Purge task *last* in a rule.
- Do not run two rules with overlapping minAge/maxAge time intervals too close together (less than 5 seconds) if the first rule contains a Purge task. Note that the interval is the time between the rules that are running (that is, the completion of one rule and the start of the next) and not between the scheduled start time of rules.

You can look at the `recording.log` file to determine when a rule has finished. Look for the following message:

```
... [] [] [] Scheduled rule [<rule name>] at location [<node path>] completed
```

The `<rule name>` and `<node path>` depend on the customer configuration. Note that the amount of time to run a rule depends on many factors, including call volume. The interval should be much greater than that suggested above to make allowances for day to day variations.

Configuring For Multiple Regions

The following sections describe how to configure MLM for multiple regions.

Need For An MLM Node In Each Region Requiring Backup and/or Purge

By design, an MLM node will only backup and/or purge call and screen recordings for which the metadata region property exactly matches the `crRegion` (call recording region) property found in the node's Interaction Recording Web Services (Web Services) `application.yaml` configuration file (if you are using Web Services and Application version 8.5.201.09 or earlier it is found in the `server-settings.yaml` file). This design prevents these nodes from "pulling" media between data centers.

For example, if there are two data centers defining regions "east" and "west", and the client Interaction Recording Web Services (Web Services) nodes with `nodePaths` (in the `application.yaml` file or in the `server-settings.yaml` file if you are using Web Services and Application version 8.5.201.09 or earlier) `/US/EAST/10.2.0.1` through `/US/EAST/10.2.0.10` are in region "east", and client Interaction Recording Web Services (Web Services) nodes with `nodePaths` `/US/WEST/10.2.1.1` through `/US/WEST/10.2.1.10` are in region "west", and there is a requirement for deleting all call recordings after 90 days, then there will need to be at least one MLM node in each region (possibly with `nodePaths` `/US/EAST/10.2.0.20` and `/US/WEST/10.2.1.20`) each with a 90-day purge rule.

Configuring SpeechMiner Purge API

If a deployment supports call recording and SpeechMiner, a deployment will need to have the SpeechMiner Purge API configured (see [SpeechMiner](#) for more information).

For a multi-region deployment that has only one SpeechMiner, the SpeechMiner Purge API should be configured with a location property value that is the nearest common ancestor of the `nodePaths` of all the MLM nodes. For instance, using the example above, the nearest common ancestor of `nodePaths` `/US/EAST/10.2.0.20` and `/US/WEST/10.2.1.20` is `/US`.

For a multi-region deployment that has one SpeechMiner per region, the SpeechMiner Purge API should be configured for the SpeechMiner of each region, using a location property value for each that is the nearest common ancestor of all the `nodePaths` of the region's Interaction Recording Web Services (Web Services) nodes. For instance, using the example above, the nearest common ancestor of `nodePaths` `/US/EAST/10.2.0.1` through `/US/EAST/10.2.0.10` and `/US/EAST/10.2.0.20` is `/US/EAST`, and the nearest common ancestor of `nodePaths` `/US/WEST/10.2.1.1` through `/US/WEST/10.2.1.10` and `/US/WEST/10.2.1.20` is `/US/WEST`.

Configuring Pre-Recording

You can configure MLM to keep the entire audio and the screen of calls that might need review of a Contact Center supervisor or manager. Use the following steps to set up Pre-recording:

1. Using Genesys Administrator Extension (GAX), select **Business Attributes** and create a new **Custom** business attribute object. Name the object **Recording**.
2. In GAX, select **Business Attribute Values** and select the **Recording** object you created in step #1 above.

3. Select **Attribute Values**, and create a new attribute value named **Keep Recording**.
4. In the **Interaction-Workspace** section, create the following parameters:
 - display-type=enum
 - enum.business-attribute=enumkeep_recording
 - enum.default-value=no
 - ready-only=false
5. In GAX, select **Business Attributes** and create a **Customer** new business attribute object. Name the new object **enumkeep_recording**.
6. In GAX, select **Business Attribute Values** and select the **enumkeep_recording** object you created in the step above.
7. Select **Attribute Values** and create the following attribute values:
 - no (set to default)
 - yes
8. In the **Workspace Web Edition Cluster** object (WWEWS_Cluster) or from the **Workspace Desktop Edition** application object, select the **Application Options** tab.
9. In the **[interaction-workspace]** section, set the **interaction.case-data.format-business-attribute** option to **Recording**.
10. From **Routing Strategy**, attach the following user data to the interaction: keep_recording="no" .. For additional information, refer to the [Universal Routing 8.1 Reference Manual](#).

Selective Recording

If your business retention policy is to keep a random percentage, say 20% of calls, then the routing strategy would call a function to determine whether to keep the call. If the call should not be kept, set the value to **keep_recording="no"**. If the call should be kept based on the rule, set the value to **keep_recording="yes"**. The agent does not need to mark the call to be kept for review.

Use the following steps to setup Pre-recording for selective recording:

1. In the routing strategy, attach the following user data to the call:
 - keep_recording="no"
2. Add the same user data on the Agent's desktop, so that the agent can change the value from no to yes if the agent wants to keep the recording based on what the caller said.

For more information about how to create rules and schedules, see [Recording Lifecycle Scheduler](#).

Upgrading the GIR Components

When upgrading from version 8.5.205.01 or earlier to version 8.5.206.01 or later, the GIR components can be upgraded in any order (Web Services, Recording Plug-in, Recording Processor

Script or Voice Processor, etc), but `callType` should not be specified in MLM tasks within the upgraded Recording plug-in until all Web Services nodes have been upgraded.

Rolling Back the GIR Components

When rolling back the components from version 8.5.206.01 or later to version 8.5.205.01 or earlier, the GIR components can be rolled back in any order (Web Services, Recording Plug-in, Recording Processor Script or Voice Processor, etc), as long as no MLM tasks specify `callType` in the filter. If a `callType` is specified as a filter of a task, the task must be removed before rolling back Web Services to a previous version. Disabling the task is not sufficient.

Advanced Settings for the MLM API

The following table describes the parameters that are in the **backgroundScheduledMediaOperationsSettings** section of the `serverSettings` section within the **application.yaml** file.

Parameter Name	Description	Default Value
<code>enableBackgroundScheduledMediaOperations</code>	Specifies whether to turn on the MLM feature.	false
<code>schedulerThreads</code>	Specifies the maximum number of enabled MLM scheduled rules that can run concurrently (that is, they start and run at the same time everyday). Valid range of values: 1 - 64. Note: If you set this value too low, the scheduled rules will run sequentially as necessary to save the number of threads opened, resulting in scheduled rules running some time after the time they are scheduled.	4
<code>schedulePollingInterval</code>	Specifies, in seconds, the time to poll for gir-scheduler settings and synchronize the rule schedule.	60
<code>speechMinerMaxConnection</code>	Specifies the maximum number of concurrent TCP connections when issuing API request to the SpeechMiner.	20
<code>speechMinerMaxTotalConnection</code>	Specifies the size of the connection pool for issuing API requests to the SpeechMiner. If set to a value less than 1, the value is automatically set to speechMinerMaxConnection *.	-1

Parameter Name	Description	Default Value
	10.	
speechMinerSocketTimeout	Specifies the time, in milliseconds, to wait for the SpeechMiner API response before timing out.	15000
defaultBackupExportURI	Specifies the file path (file:// URI) to the backup the export directory to if gir-scheduler rule setting does not specify the backup location.	Empty
enableScanAndScroll	Specifies whether to turn on the feature where MLM uses Elasticsearch scan and scroll queries to determine the recording IDs on which to act.	false
scanIntervalsPerDay	<p>When MLM is configured to use Elasticsearch scan and scroll queries to determine the recording IDs on which to act, this parameter determines the number of scan intervals used in a day of recordings. Reduce this value to reduce the number of Elasticsearch scan queries performed by an MLM Task for its work, assuming that all other things remain equal. Reducing this value also increases the lifetime of the search context created by each Elasticsearch scan query, which in turn increases the number of open file descriptors in use by Elasticsearch.</p> <p>Note: When configuring, ensure that the number of minutes in a day (i.e. 24 * 60) is exactly divisible by the configured value. If this condition is not met, RWS replaces the value with the default value if the specified value is less than or equal to zero, the next lowest integer from the specified value that is valid if the specified value is greater than zero, or 1440 if the specified value is greater than 1440.</p>	24

Creating Folder Hierarchy for Recording Storage

WebDAV performance degrades over time and the file system becomes inoperable if all your recording files are saved in a single directory. If you are using Apache HTTP Server for your WebDAV server, Genesys recommends using the following example procedures to create a folder hierarchy.

Important

Genesys recommends that you create the storage directories before your contact center goes into production. Keep in mind, that if you use the following procedure to create the directories, you will need to extend the directories at the end of 2023.

Prerequisites

Before you can create recording storage with folder hierarchy, you must have the following items in place:

- The SIP Server's **recording-filename** option must be set to `$UUID$_$DATE$_$TIME$`
- The IVR Profile Recording Storage must be set to `http://<WebDAV server>/recordings`

Important

These instructions assume that `/recordings` is the base path of the WebDAV URI. If you use something else as the base path, change it accordingly in the rewrite rule.

- The IVR Profile **Recording Filename Template** option must be set to `id`

Important

The filename must be prefixed with `id`, to allow for additional parameters after this.

- For screen recording, these instructions assume that the storage path has a base path of screens. If you use something else as the base path, change it accordingly in the rewrite rule.

Create and Use New Hierarchy

For Linux

1. Create a file and copy and paste the content below into the file (for example, "createfolders.sh"). This will create many directories in yyyy/mm/dd/hh format. The example provided creates directories to the end of 2023.

```
#!/bin/sh
for year in 2019 2020 2021 2022 2023
do
for month in 01 02 03 04 05 06 07 08 09 10 11 12
do
for day in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31
do
for hour in 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
do
echo "mkdir -p $year/$month/$day/$hour"
mkdir -p $year/$month/$day/$hour
done
done
done
done
```

After saving the file, ensure the script has read and execute privileges and then run the script from /mnt/recordings.

2. Make sure the following Apache modules are loaded in the /etc/httpd/conf/httpd.conf file.
 - LoadModule rewrite_module modules/mod_rewrite.so
 - LoadModule proxy_module modules/mod_proxy.so
3. Depending on the Apache version, add the following rewrite rules in the /etc/httpd/conf/httpd.conf file to parse the file name into /recordings/{yyyy}/{mm}/{dd}/{hh}/{filename}:
 - Apache version 2.2 and lower

```
RewriteLog "logs/rewrite.log"
RewriteLogLevel 1
RewriteEngine on
RewriteRule ^/recordings/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*)
/recordings/$2/$3/$4/$5/$1 [P,L]
RewriteRule ^/screens/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*) /screens/$2/$3/$4/$5/$1
[P,L]
RewriteRule ^/screens/([A-Za-z0-9]*_(\d{4})_(\d\d)_(\d\d)_(\d\d).*)
/screens/$2/$3/$4/$5/$1 [P,L]
```

Tip

Test this rewrite rule in your environment to ensure that all the files are placed in the correct sub-directories.

- Apache version 2.4 and higher

```
LogLevel debug rewrite:trace1
RewriteEngine on
```



```
RewriteRule ^/recordings/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*)  
/recordings/$2/$3/$4/$5/$1 [P,L]  
RewriteRule ^/screens/([A-Z0-9]*_(\d{4})-(\d\d)-(\d\d)_(\d\d).*) /screens/$2/$3/$4/$5/$1  
[P,L]  
RewriteRule ^/screens/([A-Za-z0-9]*_(\d{4})_(\d\d)_(\d\d)_(\d\d).*)  
/screens/$2/$3/$4/$5/$1 [P,L]
```

Important

For Apache version 2.4 and higher, rewrite logs are not written to a separate file (as in Apache 2.2) but to error_log file. To get the mod_rewrite specific log messages, you can use the following command:

```
tail -f error_log | fgrep '[rewrite:]'
```

Setting up the Load Balancer in a Multi-Tenant Environment

See also: [Setting up the Load Balancer in a Single-Tenant Environment](#)

Important

- The load balancer used for RWS must be configured with sufficient capacity to accommodate one persistent connection from each logged in agent with SR Service in addition to other RWS requests.
- Currently, Genesys does not provide instructions on how to set up load balancer for the Voice Processor. You can configure your own load balancing solution for multiple Voice Processor instances, if required.
- The architecture for load balancer cluster is supported on Red Hat Enterprise Linux 6 for HTTPD 2.2 only.

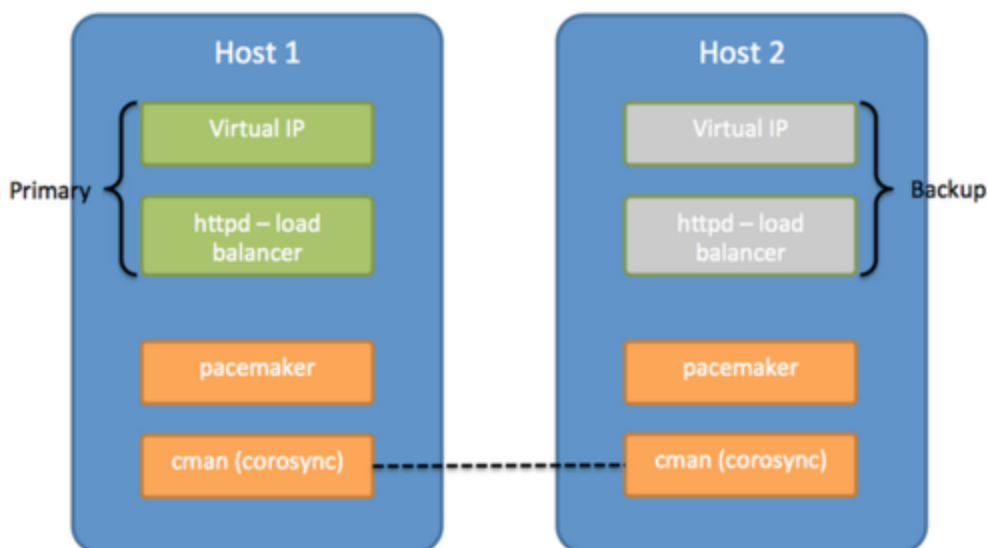
Red Hat Enterprise Linux 6 for HTTPD 2.2

Overview and Architecture

The solution uses a common Linux HA framework from <http://clusterlabs.org>. There are two components involved in this solution:

- Cman uses corosync internally to provide a platform for membership, messaging, and quorum among the hosts.
- Pacemaker is a cluster resource manager that controls where resources (processes) are executed. Pacemaker works with the processes like Apache httpd using resource agents to provide controls of the process such as start/stop/status.

The following diagram shows a primary/backup design to associate a single virtual IP address with httpd. Whenever the primary host fails, the virtual IP address and the httpd process can be automatically fail over to the backup host.



As a simple two host primary/backup solution, the hosts must be deployed on the same subnet that allows UDP multicast. This solution provides the same reliability as a network that hosts the two machines handling the virtual IP address.

Deploying the Load Balancer

Important

For load balancers used for Recording Processors, warm standby functionality must be disabled.

Prerequisites

- Red Hat Enterprise Linux 6 with the High Availability Add-On, for HTTPD 2.2

Tip

Network Manager can be enabled as part of the OS installation. To disable Network Manager, see [Red Hat documentation](#).

Installing the OS

Install the required software using the following command:

```
yum -y install httpd pacemaker cman pcs ccs resource-agents
```

Setting up the HTTP Load Balancer

Please note that any URL setup for the various GIR components described in the [Multi-Tenant Deployment](#) should now point to the respective loadbalancer URLs, such as

- RPS URL: <loadbalancer URL>/t1/rp/api
- htcc.baseurl should point to the RWS loadbalancer URL: <loadbalancer URL>/t1
- rcs.base_uri should point to <loadbalancer URL>/t1/rcs

Important

Only GIR releases post-8.5.210.02 with WDE support multi-tenancy.

On both servers, create the following files:

- Create /etc/httpd/conf.d/serverstatus.conf, and add the following text:

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

For each tenant, create a separate /etc/httpd/conf.d/loadbalancer_tenantN.conf file. Use the Include directive within the main /etc/httpd/conf/httpd.conf to include each tenant configuration:

```
Include /etc/httpd/conf.d/loadbalancer_tenantN.conf
```

In addition, provide each tenant with a separate balancer rule, ProxyPass and the following URI conventions:

- Interaction Recording Web Services
 - http://loadbalancer/t1/api
 - http://loadbalancer/t1/internal-api
- Recording Processor
 - http://loadbalancer/t1/rp
- Recording Crypto Server
 - http://loadbalancer/t1/rcs
- Interaction Receiver

- `http://loadbalancer/t1/interactionreceiver`
- WebDAV Server
 - `http://loadbalancer/t1/recordings`
- For each tenant, create `/etc/httpd/conf.d/loadbalancer_tenantN.conf`, and add the following text:

Important

If your existing configuration already includes the loadbalancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the `_` (underscore) characters. For example, if the application name is RCS 1, set the **route** value to `RCS_1`.

loadbalancer_tenantN.conf

```
# Interaction Recording Web Services for tenant 1
<Proxy balancer://tlrws>
BalancerMember http://tlrws1:8080 route=TlRWS1
BalancerMember http://tlrws2:8080 route=TlRWS2
BalancerMember http://tlrws3:8080 route=TlRWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://tlrws/api
ProxyPass /t1/internal-api balancer://tlrws/internal-api

# RP for tenant 1
<Proxy balancer://tlrp>
BalancerMember http://tlrp1:8889
BalancerMember http://tlrp2:8889
</Proxy>
ProxyPass /t1/rp/api balancer://tlrp/api

# RCS for tenant 1
<Proxy balancer://tlrcs>
BalancerMember http://tlrcs1:8008 disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://tlrcs2:8008 disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
ProxyPassReverseCookiePath "/rcs" "/t1/rcs"
</Proxy>
ProxyPass /t1/rcs balancer://tlrcs/rcs

# Interaction Receiver for tenant 1
```

```

<Proxy balancer://tlsm>
BalancerMember http://tlir1
BalancerMember http://tlir2
</Proxy>
ProxyPass /t1/interactionreceiver balancer://tlsm/interactionreceiver

# WebDAV for tenant 1
<Proxy balancer://tlwebdav>
BalancerMember http://tlwebdav1
BalancerMember http://tlwebdav2 status=H
</Proxy>
ProxyPass /t1/recordings balancer://tlwebdav/recordings
ProxyPass /t1/dest2 balancer://tlwebdav/dest2

```

Configuring TLS for the HTTP Load Balancer

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: /etc/pki/tls/certs/localhost.crt
- Key: /etc/pki/tls/private/localhost.key

2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **/etc/httpd/conf.d/ssl.conf** file and modify the following lines:

- SSLCertificateFile /etc/pki/tls/certs/localhost.crt
- SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

3. To enable https for the proxy, edit the **/etc/httpd/conf.d/ssl.conf** file and add the following option:
SSLProxyEngine on

4. Direct the load balancer to the proper https locations. For example:

```

<Proxy balancer://tlrws>
BalancerMember https://tlrws1:8080 route=TLRWS1
BalancerMember https://tlrws2:8080 route=TLRWS2
BalancerMember https://tlrws3:8080 route=TLRWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://tlrws/api
ProxyPass /t1/internal-api balancer://tlrws/internal-api

```

Setting Up Pacemaker and Cman

Important

Perform the following commands using the **root** user.

Disable Autostart for Httpd

Pacemaker manages the startup of httpd. Disable httpd from chkconfig services using the following command:

```
chkconfig httpd off
```

Setting Up the Hosts File

Make sure there is a hostname for both servers and that the hostname is resolvable on both hosts, either using DNS or /etc/hosts file. ip1 and ip2 are used as the hostnames thereafter.

```
# /etc/hosts
# ... keep the existing lines, and only append new lines below
192.168.33.18 ip1
192.168.33.19 ip2
```

Setting Up the Cluster

Run the following command on each host to create the cluster configuration:

```
ccs -f /etc/cluster/cluster.conf --createcluster webcluster
ccs -f /etc/cluster/cluster.conf --addnode ip1
ccs -f /etc/cluster/cluster.conf --addnode ip2
ccs -f /etc/cluster/cluster.conf --addfencedev pcmk agent=fence_pcmk
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect ip1
ccs -f /etc/cluster/cluster.conf --addmethod pcmk-redirect ip2
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk ip1 pcmk-redirect port=ip1
ccs -f /etc/cluster/cluster.conf --addfenceinst pcmk ip2 pcmk-redirect port=ip2
ccs -f /etc/cluster/cluster.conf --setcman two_node=1 expected_votes=1
echo "CMAN_QUORUM_TIMEOUT=0" >> /etc/sysconfig/cman
```

Start the Service

Start the cman and pacemaker services on each host using the following command:

```
service cman start
service pacemaker start
chkconfig --level 345 cman on
chkconfig --level 345 pacemaker on
```

(Optional) Setting Up UDP Unicast

This solution relies on UDP multicast to work, but can also work with UDP unicast. Edit the **/etc/cluster/cluster.conf** file and insert an attribute to the <cman> tag as follows:

```
...
<cman transport="udpu" two_node="1" expected_votes="1"/>
...
```

Restart both servers for the changes to take effect.

Setting Cluster Defaults

Run the following on one of the servers.

```
pcs property set stonith-enabled=false
pcs property set no-quorum-policy=ignore
pcs resource defaults migration-threshold=1
```

Configure the Virtual IP Address and Apache httpd

Run the following on one of the servers.

For the first command below, `nic=eth0` refers to the network interface that brings up the virtual IP address. Change `eth0` to the active network interface your environment uses.

Change `<Virtual IP>` in the first command below to your virtual IP assigned to this load balancer pair.

```
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=<Virtual IP> nic=eth0 cidr_netmask=32
op monitor interval=30s
pcs resource create webserver ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf
statusurl="http://localhost/server-status" op monitor interval=30s
pcs resource meta webserver migration-threshold=10
pcs constraint colocation add webserver virtual_ip INFINITY
pcs constraint order virtual_ip then webserver
```

Maintaining Pacemaker

The following commands help you with the maintenance operations for pacemaker.

To check the status of the cluster:

```
pcs status
```

To clear resource errors (for example, because of incorrect configuration):

```
pcs resource cleanup <resourcename>
```

A resource name is either `virtual_ip` or `web server` (for example, `pcs resource cleanup webserver`).

To check the status of the resources in the cluster:

```
crm_mon -o -l
```

Red Hat Enterprise Linux 8 for HTTPD 2.4

Deploying the Load Balancer

Important

For load balancers used for Recording Processors, warm standby functionality must be disabled.

Prerequisites

- Red Hat Enterprise Linux 8 with the High Availability Add-On, for HTTPD 2.4

Tip

Network Manager can be enabled as part of the OS installation. To disable Network Manager, see [Red Hat documentation](#).

Installing the OS

Install the required software using the following command:

```
yum -y install httpd
```

Setting up the HTTP Load Balancer

Please note that any URL setup for the various GIR components described in the [Multi-Tenant Deployment](#) should now point to the respective loadbalancer URLs, such as

- RPS URL: <loadbalancer URL>/t1/rp/api
- htcc.baseurl should point to the RWS loadbalancer URL: <loadbalancer URL>/t1
- rcs.base_uri should point to <loadbalancer URL>/t1/rcs

Important

Only GIR releases post-8.5.210.02 with WDE support multi-tenancy.

On both servers, create the following files:

- Create /etc/httpd/conf.d/serverstatus.conf, and add the following text:

```
<Location /server-status>  
  SetHandler server-status
```

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

For each tenant, create a separate `/etc/httpd/conf.d/loadbalancer_tenantN.conf` file. Use the `Include` directive within the main `/etc/httpd/conf/httpd.conf` to include each tenant configuration:

```
Include /etc/httpd/conf.d/loadbalancer_tenantN.conf
```

In addition, provide each tenant with a separate balancer rule, ProxyPass and the following URI conventions:

- Interaction Recording Web Services
 - `http://loadbalancer/t1/api`
 - `http://loadbalancer/t1/internal-api`
- Recording Processor
 - `http://loadbalancer/t1/rp`
- Recording Crypto Server
 - `http://loadbalancer/t1/rcs`
- Interaction Receiver
 - `http://loadbalancer/t1/interactionreceiver`
- WebDAV Server
 - `http://loadbalancer/t1/recordings`
- For each tenant, create `/etc/httpd/conf.d/loadbalancer_tenantN.conf`, and add the following text:

Important

If your existing configuration already includes the loadbalancer rules in the `/etc/httpd/conf/httpd.conf`, skip this step.

The following lines starting with **BalancerMember** refer to the URL to the servers for Interaction Recording Web Services, Recording Processor, Recording Crypto Server, Interaction Receiver, and WebDAV server.

For Recording Crypto Server, the **route** value must be set to the application name of the Recording Crypto Server instance, where the " " (space) characters in the name are replaced with the _ (underscore) characters. For example, if the application name is RCS 1, set the **route** value to `RCS_1`.

loadbalancer_tenantN.conf

```
# Interaction Recording Web Services for tenant 1
<Proxy balancer://tlrws>
BalancerMember http://tlrws1:8080 route=T1RWS1
BalancerMember http://tlrws2:8080 route=T1RWS2
BalancerMember http://tlrws3:8080 route=T1RWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://tlrws/api
ProxyPass /t1/internal-api balancer://tlrws/internal-api

# RP for tenant 1
<Proxy balancer://tlrp>
BalancerMember http://tlrp1:8889
BalancerMember http://tlrp2:8889
</Proxy>
ProxyPass /t1/rp/api balancer://tlrp/api

# RCS for tenant 1
<Proxy balancer://tlrcs>
BalancerMember http://tlrcs1:8008 disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember http://tlrcs2:8008 disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
ProxyPassReverseCookiePath "/rcs" "/t1/rcs"
</Proxy>
ProxyPass /t1/rcs balancer://tlrcs/rcs

# Interaction Receiver for tenant 1
<Proxy balancer://tlism>
BalancerMember http://tlir1
BalancerMember http://tlir2
</Proxy>
ProxyPass /t1/interactionreceiver balancer://tlism/interactionreceiver

# WebDAV for tenant 1
<Proxy balancer://tlwebdav>
BalancerMember http://tlwebdav1
BalancerMember http://tlwebdav2 status=H
</Proxy>
ProxyPass /t1/recordings balancer://tlwebdav/recordings
ProxyPass /t1/dest2 balancer://tlwebdav/dest2
```

Configuring TLS for the HTTP Load Balancer

1. On the WebDAV server, run the following command to install SSL:

```
yum install mod_ssl
```

The certificate/key pair is automatically generated:

- Certificate: /etc/pki/tls/certs/localhost.crt

-
- Key: /etc/pki/tls/private/localhost.key
2. To use your own certificate/key pair, either update the files automatically generated (as above), or edit the **/etc/httpd/conf.d/ssl.conf** file and modify the following lines:
 - SSLCertificateFile /etc/pki/tls/certs/localhost.crt
 - SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
 3. To enable https for the proxy, edit the **/etc/httpd/conf.d/ssl.conf** file and add the following option:
SSLProxyEngine on
 4. Direct the load balancer to the proper https locations. For example:

```
<Proxy balancer://tlrws>
BalancerMember https://tlrws1:8080 route=TLRWS1
BalancerMember https://tlrws2:8080 route=TLRWS2
BalancerMember https://tlrws3:8080 route=TLRWS3
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e; path=/t1"
env=BALANCER_ROUTE_CHANGED
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPass /t1/api balancer://tlrws/api
ProxyPass /t1/internal-api balancer://tlrws/internal-api
```

Additional Feature Configuration

Audio Tones

The following section outlines the general configuration for audio tones.

Media Server

The following table describes the options required for audio tones when using Media Server:

Section Name	Parameter Name	Description
Conference	record_recorddnhearstone	Specifies whether the RecordDN (Party A) hears the repeating tone.
Conference	record_otherdnhearstone	Specifies whether the OtherDN (Party B) hears the repeating tone.

Media Server allows you to configure whether the recording also gets the audio tone. When the audio tone is injected into the call, Media Server distinguishes between what the participant hears and what the participant says. The above two configuration parameters affect what the participant hears.

Section Name	Parameter Name	Description
Conference	record_chan2source	<p>Specifies the recorded media that represents the first participant (Record DN) in the recording session.</p> <ul style="list-style-type: none">recorddnsaysotherdnhears <p>If the Other DN is configured to receive consent and you want the consent to be recorded, set the value to otherdnhears.</p>
Conference	record_otherdnhearstone	<p>Specifies the recorded media that represents the second participant (Other DN) in the recording session.</p> <ul style="list-style-type: none">otherdnsays

Section Name	Parameter Name	Description
		<ul style="list-style-type: none">recorddnhears <p>If the Record DN is configured to receive consent and you want the consent to be recorded, set the value to recorddnhears.</p>

Enable Call Recording

Call recording can be enabled through three methods:

1. **Full-time recording or Total recording**—A specific DN is configured to enable recording for all calls for the specific DN.
2. **Selective Recording**—Record a party in the call is determined at a route point and the recording starts as soon as the call is established.
3. **Dynamic Recording**—Start/stop/pause/resume a recording call can be requested by an agent at any time after the call is established using Interaction Workspace.

Once a recording has started, there are two conditions where the recording stops:

1. When the party being recorded leaves the call, or when the customer drops the call. For example, when the recording applies to the agent in the call and the call is transferred to a second agent. The recording is stopped when the agent leaves the call. Note that the second agent can have recording enabled and the same call gets recorded with a second call recording segment.
2. When dynamic recording control requests the recording to be stopped.

Important

If using Workspace Desktop Edition for the agent desktop, the agent can hide the status of the recording. This functionality can be enabled through Workspace role configuration. For more information, see the [Setting Up Agents on the System](#) in the Workspace Desktop Edition documentation.

Genesys Interaction Recording Options Reference

Recording Processor Script

The following describes the options for the Recording Processor Script application.

[+] config_server Section

application_name

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the application name of the Recording Processor Script.

hostname

Default Value: <ip address>

Valid Values: Any string

Changes Take Effect: After restart

Specifies the IP address of the Configuration Server host.

port

Default Value: 2020

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the port of the Configuration Server host.

username

Default Value: default

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to connect to the Configuration Server.

password

Default Value: password

Valid Values: Any string
Changes Take Effect: After restart

Specifies the password used to connect to the Configuration Server.

backup_host

Default Value: Empty
Valid Values: Any string
Changes Take Effect: After restart

Specifies the IP address of the backup Configuration Server.

backup_port

Default Value: Empty
Valid Values: Any integer
Changes Take Effect: After restart

Specifies the port for the backup Configuration Server.

config_server_encoding

Default Value: Empty
Valid Values: UTF-8
Changes Take Effect: After restart

Specifies whether to enable decoding of the multibyte usernames from Configuration Server. This parameter is optional. If left empty, RPS might post the multibyte usernames incorrectly, resulting in two usernames being displayed for calls in SpeechMiner. You need to enable this option in the following circumstance:

- Configuration Server is configured for multiple languages (UTF-8).
- Microsoft SQL Server is used for the Genesys Interaction Recording (GIR) ICON database.
- The agent username contains characters encoded as multiple bytes in UTF-8.

[+] processing Section

check_agent_prev_state

Default Value: 0
Valid Values: 0 (no), 1 (yes)
Changes Take Effect: After restart

Specifies whether or not Recording Processor Script will wait for the agent to complete After Call Work (ACW) and to no longer be in the ACW state before retrieving the ACW data from the ICON Database.

Note: Enabling this setting can cause delays in processing calls with ACW data.

post_acw_delay

Default Value: 0

Valid Values: 0, or any positive integer

Changes Take Effect: After restart

The minimum amount of time (in seconds) that Recording Processor Script will wait for After Call Work (ACW) data to be processed before retrieving ACW data from the ICON database. If this setting is enabled, it causes RPS to wait (one or more times) for a duration that is approximately equal to **process_wait_seconds** until **post_acw_delay** seconds have passed before collecting ACW data. The purpose of this setting is to ensure all ACW data is collected for all calls.

encoding

Default Value: utf_8

Valid Values: Value defined in [Codec registry and base classes](#)

Changes Take Effect: After restart

Expected encoding type for strings queried from the ICON database. Refer to [Codec registry and base classes](#) to obtain the value that is applicable to your ICON database.

failed_folder_path

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the folder to save failed recordings to.

mode

Default Value: active

Valid Values: active, backup

Changes Take Effect: After restart

Specifies the High Availability Mode of the Recording Processor instance.

post_uri

Default Value: http://<active_rp_ip>:<port>/api/contact-centers/%s/recordings/

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the URL to be used by a backup node to send metadata to an active node (for HA).

get_from_httpc_before_posting

Default Value: 1

Valid Values: 0 (no), 1 (yes)

Changes Take Effect: After restart

Specifies whether Recording Processor is to fetch from data from Interaction Recording Web Services

(or Web Services if you're using version 8.5.210.02 or earlier) before sending a POST.

`acw_threshold_minutes`

Default Value: 5

Valid Values: 0 or any positive integer

Changes Take Effect: After restart

Specifies the maximum time, in minutes, that Recording Processor Script will wait in After Call Work (ACW) mode. If this time is exceeded, Recording Processor Script will skip collecting the ACW customized data from ICON. If set to 0, Recording Processor Script will not collect ACW customized data from ICON.

`enable_acw`

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to enable After Call Work data. If set to 0, collecting ACW data is disabled.

`backup_failed_metadata`

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to back up the messages that fail to POST correctly to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner to the `<recording processor dir>\failed` folder.

`include_unknown_agent`

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

When this option is set to true (default), and if the agent's username is missing, then the username field will be populated as "UNKNOWN" in the eventHistory metadata. If this option is set to false, and if the agent's username is missing, then the username field will not be updated in the eventHistory metadata. This parameter is not available in the default configuration file. If you want to set this parameter to false, add it manually.

[+] client Section

`certs`

Default Value: tests/cacerts.txt

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the path to the Certificate Authority certificates used to validate the SSL connections to Workspace Web Edition and/or SpeechMiner.

`http_timeout`

Default Value: 20

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the HTTP timeout duration, in seconds, for the Recording Processor Script when sending messages to Workspace Web Edition and/or SpeechMiner.

[+] wait_times Section

`message_wait_seconds`

Default Value: 20

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the internal delay, in seconds, to wait before checking message queue.

`process_wait_seconds`

Default Value: 10

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the internal delay, in seconds, to wait between message processing attempts.

[+] metadata Section

`region`

Default Value: `region1`

Valid Values: Any string

Changes Take Effect: After restart

Specifies the region for the metadata.

`call_end_threshold_minutes`

Default Value: 30

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the length of time, in minutes, after a call ends before Recording Processor Script sends the call's metadata to SpeechMiner.

[+] rp_server Section

port

Default Value: 8889

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the port of the Recording Processor Script (RPS) REST server.

addr

Default Value: 0.0.0.0

Valid Values: Any valid IP address or 0.0.0.0

Changes Take Effect: After restart

Specifies the IP address of the Recording Processor Script (RPS) REST server.

ssl_certificate

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the path to the file that contains the SSL certificate required to support the HTTPS server.

ssl_private_key

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the path to the PEM file that contains the SSL private key required to support the HTTPS server.

Important

This file should **NOT** be password protected.

[+] htcc Section

base_uri

Default Value: http://<htcc_ip>:<port>

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the Base URI for accessing the Interaction Recording Web Services (or Web Services if

you're using version 8.5.210.02 or earlier) API.

get_uri

Default Value: /api/v2/ops/contact-centers/%s/recordings

Valid Values: Any valid URI reference

Changes Take Effect: After restart

Specifies the URI suffix used to retrieve metadata from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). This is appended to the base_uri. %s is replaced internally by the Contact Center ID. Normally it is sufficient to change the base_uri and this option does not need to be changed.

post_uri

Default Value: /internal-api/contact-centers/%s/recordings

Valid Values: Any valid URI reference

Changes Take Effect: After restart

Specifies the URI suffix used to send metadata to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). This is appended to the base_uri. %s is replaced internally by the Contact Center ID. Normally it is sufficient to change the base_uri and this option does not need to be changed.

username

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

password

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

csrfp

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to enable the Cross Site Request Forgery (CSRF) protection mode. This option must be enabled (set to 1) if Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has CSRF enabled.

[+] speechminer Section

post_uri

Default Value: Empty

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the URL used to send metadata to the SpeechMiner Interaction Receiver.

Note: This should not typically be specified; instead the information configured in the IVR Profile is used for SpeechMiner connectivity. Refer to the Recording Destinations section on the Recording Tab, described in the Step #4 table in the [IVR Profile](#) section.

username

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the SpeechMiner operating account.

Note: This should not typically be specified; instead the information configured in the IVR Profile is used for SpeechMiner connectivity. Refer to the Recording Destinations section on the Recording Tab, described in the Step #4 table in the [IVR Profile](#) section.

password

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the SpeechMiner operating account.

Note: This should not typically be specified; instead the information configured in the IVR Profile is used for SpeechMiner connectivity. Refer to the Recording Destinations section on the Recording Tab, described in the Step #4 table in the [IVR Profile](#) section.

disable_ssl_certificate_validation

Default Value: 1

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to disable validation of the SpeechMiner certificate when establishing a TLS connection to the SpeechMiner Interaction Receiver.

[+] persistence Section

table_name

Default Value: No default value

Valid Values: Any string

Changes Take Effect: After restart

Specifies the table name that the Recording Processor Script uses to store data.

`db_filename`

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the filename (full path or relative path) used for the sqlite3 database file.

[+] logfile Section

`max_log_file_size_mb`

Default Value: 21

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the maximum size, in MB, of the Recording Processor Script log file.

`logfile_path`

Default Value: No default value

Valid Values: Any valid file path

Changes Take Effect: After restart

Specifies the filename (full path or relative path) of the Recording Processor Script log file.

`level`

Default Value: INFO

Valid Values: DEBUG, INFO, WARNING, ERROR, CRITICAL

Changes Take Effect: After restart

Specifies the level or severity of the events to track in the log file. For more information about these logging levels, see the [Python documentation](#).

`log_backup_count`

Default Value: 100

Valid Values: Any integer (a non-zero integer is recommended)

Changes Take Effect: After restart

Specifies the number of log files to back up. When **log_backup_count** is a non-zero integer and the current log file reaches **max_log_file_size_mb**, the system closes the current log file and creates a new log file with the current time appended to the filename. For example, when **log_backup_count** is set as 5, the system will have backup for a maximum of 5 log files after the current log file size reaches **max_log_file_size_mb**.

Recording Crypto Server

Application-Level Options

The following describes the options for the Recording Crypto Server application.

[+] general Section

archive.enable

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Specifies whether to enable archiving for recordings.

samesite.enable

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Specifies whether the **SameSite=None** and **Secure** cookie attributes are set during screen recording playback from the SpeechMiner browser application.

Important

Before enabling this option, ensure that the connection between the SpeechMiner browser application and Recording Crypto Server is configured to use HTTPS. If you set the value of this option to `true` and are using HTTP, the cookie will not be returned by the browser.

gwsAuthUri

Default Value: Empty string

Valid Values: Any URL path

Changes Take Effect: After restart

This parameter is not supported. Do not change the default value.

[+] htcc Section

baseurl

Default Value: `https://htcchost:8080`

Valid Values: Any URL path

Changes Take Effect: After restart

Specifies the base URL for the connection to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). This parameter is dependent on the Interaction Recording Web Services (Web Services) server protocol (http, or https), port, and URL suffix.

internalUrlPrefix

Default Value: /api/v2

Valid Values: Any string

Changes Take Effect: After restart

Controls the prefix added to requests sent to Interaction Recording Web Services to retrieve recording files. By default, or if a value other than **disable** is specified, RCS will concatenate the **baseurl**, **internalUrlPrefix**, and the **mediaPath** returned by RWS as the request URL. If the **internalUrlPrefix** value is set to **disable**, RCS will use the **mediaUri** from the metadata instead when fetching the recordings from RWS.

domain

Default Value: Empty string

Valid Values: Any string

Changes Take Effect: After restart

Specifies the domain of the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) contact center. This is the domain ID set for the contact center within Interaction Recording Web Services (Web Services).

user

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the name of the user for the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection.

password

Default Value: opspassword

Valid Values: Any positive integer

Changes Take Effect: After restart

Specifies the password of the user for the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) connection.

max-sr-playback-connections

Default Value: 50

Valid Values: Any string

Changes Take Effect: After restart

Specifies the maximum number of HTTP connections between Recording Crypto Server and Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for

screen recording playback.

contactcenterid

Default Value: Empty string

Valid Values: Any string

Changes Take Effect: After restart

Specifies the contact center ID value in the RCS requests sent to Interaction Recording Web Services (RWS). If this value is not specified, the contact center ID information is derived from the **/api/v2/ops/contact-centers** request sent to RWS.

trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). For more information, see [Configuring TLS connection to Interaction Recording Web Services](#).

[+] cors Section

allowed-origins

Default Value: Empty

Valid Values: Any comma-separated list

Changes Take Effect: After restart

Specifies the allowed origins list that is attached in the HTTP response Access-Control-Allow-Origins header, sent to a cross-origin request.

allowed-headers

Default Value: X-Requested-With, Content -

Type, Accept, Origin, Cookie, authorization, ssid, url, ContactCenterId, X-CSRF-TOKEN, Range

Valid Values: Any comma-separated list

Changes Take Effect: After restart

Specifies the allowed headers list that is attached in the HTTP response Access-Control-Allow-Headers header, sent to a cross-origin request.

allowed-methods

Default Value: GET, POST, PUT, DELETE, OPTIONS

Valid Values: Any comma-separated list

Changes Take Effect: After restart

Specifies the allowed methods list that is attached in the HTTP response Access-Control-Allow-

Methods header, sent to a cross-origin request.

allow-credentials

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Specifies the value sent in the Access-Control-Allow-Credentials header of the HTTP response to a cross-origin request.

[+] keystore Section

max-read-attempts

Default Value: 5

Valid Values: -1 or a positive integer. When using -1, RCS will retry endlessly.

Changes Take Effect: After restart

Specifies the maximum number of attempts to read keystore during startup.

read-interval

Default Value: 1

Valid Values: 1-30

Changes Take Effect: After restart

Specifies time interval, in seconds, between the keystore read attempts.

[+] log Section

suppress-debug-loggers

Default Value: Empty string

Valid Values: Any package name. The package name can contain the wildcard (*) character. For example, org.apache.http.wire.*.

Changes Take Effect: After restart

Suppresses the debug logs by Jetty for the specified package names.

Tenant-Level Options

The following describes the options for the Tenant.

[+] recording.archive Section

interval

Default Value: 1

Valid Values: 0-30

Changes Take Effect: After restart

Specifies how often, in days, the archiving process runs. If set to 0, archiving is disabled for the tenant.

retentiontime

Default Value: 60

Valid Values: Between 1 and 5*365 (5 years)

Changes Take Effect: After restart

Specifies how long (in days) to maintain the recordings in the system before they are archived by RCS.

outputfolder

Default Value: archive

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the destination folder where the archived recordings are stored.

speechminerurl

Default Value: https://<host or IP address of the SpeechMiner host>/speechminer

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the SpeechMiner database destination where the recording metadata is stored.

user

Default Value: archiveuser

Valid Values: Any string

Changes Take Effect: After restart

Specifies the SpeechMiner username that is required for authentication.

password

Default Value: changeit

Valid Values: Any string

Changes Take Effect: After restart

Specifies the SpeechMiner password that is required for authentication.

speechminer-trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to SpeechMiner Interaction Receiver. For more information, see [Configuring TLS connection to SpeechMiner Interaction Receiver](#).

Recording Plug-in for GAX

The following describes the options for the Plug-in for GAX (Genesys Administrator Extension).

Tenant-Level Options

[+] recording Section

`rcsurl`

Default Value: `https://rcshost:8080`

Valid Values: Any URL

Changes Take Effect: After restart

Specifies the URL to the Recording Crypto Server.

`htcc_base_url`

Default Value: Empty

Valid Values: Interaction Recording Web Services load balancer URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; `http://<Interaction Recording Web Services IP Address>:8081`

Changes Take Effect: After restart

Specifies the HTCC Base URL.

`trusted_ca_rcs`

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Recording Crypto Server (RCS).

`trusted_ca_rws`

Default Value: `true`

Valid Values: `true`, `false`

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Interaction Recording Web Services (RWS).

GAX-Level Options

[+] rcs Section

htcc_base_url

Default Value: Empty

Valid Values: Interaction Recording Web Services load balancer URL (or Web Services server URL if you're using version 8.5.210.02 or earlier); for example; http://<Interaction Recording Web Services IP Address>:8081

Changes Take Effect: After restart

Specifies the HTCC Base URL.

trusted_ca_rcs

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Recording Crypto Server (RCS).

trusted_ca_rws

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Configures whether or not the certificate will be validated when making a secure outbound connection to Interaction Recording Web Services (RWS).

htcc_http_timeout

Default Value: 20

Valid Values: Integers > 0

Changes Take Effect: After restart

Specifies the HTTP timeout value (in seconds) for HTCC API.

httpclient.cookiepolicy

Default Value: BEST_MATCH

Valid Values: rfc2109, rfc2965, BEST_MATCH, BROWSER_COMPATIBILITY, NETSCAPE, IGNORE_COOKIES

Changes Take Effect: After restart

Allows a user to specify which cookie policy to use for HTCC to work around the load balancer settings.

rccs_keepalive_interval

Default Value: 60

Valid Values: Integers > 0

Changes Take Effect: After restart

The interval in seconds the Plugin will be sending a request to RCS to keep the user session alive.

htcc_keepalive_interval

Default Value: 60

Valid Values: Integers > 0

Changes Take Effect: After restart

The interval in seconds the Plugin will be sending a request to HTCC API to keep the user session alive.

show_cert_with_upload_privilege

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Configures whether or not the GAX requires the **RECORD_KEY_UPLOAD** privilege to show the **Screen Recording Certificates** menu.

Recording Muxer Script

The following describes the options for the Recording Muxer Script.

[+] htcc Section

base_uri

Default Value: Empty

Valid Values: Any URL

Changes Take Effect: After restart

Specifies the host and port of the Interaction Recording Web Services server (or Web Services server if you're using version 8.5.210.02 or earlier) (for example, https://<web_services_host>:<web

`services port>/).`

`contact_center_id`

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the unique identifier of the contact center.

`username`

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

`password`

Default Value: ops

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) operating account.

`trusted-ca`

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). For more information, see [Configuring TLS connection to Interaction Recording Web Services](#).

[+] rcs Section

`base_uri`

Default Value: Empty

Valid Values: Any valid URL

Changes Take Effect: After restart

Specifies the host and port of the Recording Crypto Server (for example, `https://<Recording Crypto Server host>:<Recording Crypto Server port>`).

username

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to access the Recording Crypto Server account belonging to the contact center specified by the **contact_center_id** option in the **htcc** section.

password

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to access the Recording Crypto Server account belonging to the contact center specified by the **contact_center_id** option in the **htcc** section.

trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to Recording Crypto Server (RCS). For more information, see [Configuring TLS connection to Recording Crypto Server](#).

[+] processing Section

auto_clean_temp_folder

Default Value: 1

Valid Values: 1, 0

Changes Take Effect: After restart

If set to 1, Recording Muxer Script will automatically delete the recording files from the temp folder.

batch_read_screen_recording_metadata

Default Value: 1

Valid Values: Bulk API = 1 / GWS request = <>1

Changes Take Effect: After restart

Determines how screen recording metadata is received. The new algorithm reads multiple screen recordings metadata in one request. The previous algorithm reads one request at a time.

check_matching_username

Default Value: 1

Valid Values: 1, 0

Changes Take Effect: After restart

If set to 1, Recording Muxer Script will mux a screen recording with a matching call recording only when the screen recording contains the same username value as in the call recording metadata. Note that Recording Muxer Script will always mux the recordings regardless of the **check_matching_username** value if the username is "UNKNOWN", undefined, or empty in the call recording metadata.

clean_temp_folder_timeout

Default Value: 43200

Valid Values: Integer

Changes Take Effect: After restart

Determines how often the recording files are cleaned up in the temp folder.

split_window_enabled

Default Value: 0

Valid Values: 1, 0

Changes Take Effect: After restart

Setting this parameter to 1 enables splitting the window into sub-intervals in order to improve Elasticsearch performance when querying RWS. If set to 0, this feature is disabled.

max_interval_minutes

Default Value: 30

Valid Values: 0 to total minutes between `window_past` and `window_past_older_than` values.

Changes Take Effect: After restart

If **split_window_enabled** is set to 1, **max_interval_minutes** specifies the maximum duration of the sub-interval in minutes.

muxer_type

Default Value: Empty

Valid Values: primary, backup

Changes Take Effect: After restart

Determines the primary Recording Muxer Script instance or backup Recording Muxer Script instance. If you select not to use Sharding, the system ignores this value.

muxer_id

Default Value: -1

Valid Values: A non-negative integer starting with 0 (every Muxer ID is incremented by 1)

Changes Take Effect: After restart

A unique Muxer ID. If you select not to use Sharding, the value should be empty or -1.

The Recording Muxer Script backup instance ID is automatically calculated as follows: $(id + 1) \% \text{total number of muxers}$

total_muxers

Default Value: Empty

Valid Values: $\max(\text{muxer_id}) + 1$

Changes Take Effect: After restart

The total number of Recording Muxer Script instances deployed (excluding the backup). If **muxer_id** is less than 0, the system ignores this value.

window_past

Default Value: 720

Valid Values: Any integer

Changes Take Effect: After restart

Specifies how many minutes into the past to search for those call recordings that are to be combined with the screen recordings.

window_past_older_than

Default Value: 5

Valid Values: Any integer

Changes Take Effect: After restart

Specifies how many minutes into the past to search for those call recordings that are to be combined with the screen recordings. The age of the call recording is determined by the current time and the endTime metadata. If left empty, the window_past parameter is used to search for call recordings. If not empty, then both the window_past and the window_past_older_than parameters are used to search for call recordings.

min_poll_interval

Default Value: 5

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the duration, in seconds, for the Recording Muxer Script to poll Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for the new recordings to be multiplexed.

ffmpeg

Default Value: ffmpeg

Valid Values: Any valid relative or absolute file path

Changes Take Effect: After restart

Specifies the ffmpeg executable with full directory path. If ffmpeg is specified in the executable path,

ffmpeg is sufficient to run the command.

ffprobe

Default Value: ffprobe

Valid Values: Any valid relative or absolute file path

Changes Take Effect: After restart

Specifies the path to the ffprobe executable file. If ffprobe is specified in the file path, ffprobe is sufficient to run the command.

openSSL

Default Value: openssl

Valid Values: Any valid relative or absolute file path

Changes Take Effect: After restart

Specifies the path to the openSSL executable file. If openSSL is specified in the file path, openSSL is sufficient to run the command.

encrypt_always

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to force the combined call and screen recordings to be uploaded as encrypted even if the source screen recording is not encrypted. Set to 0 if the screen recording is encrypted. Set to 1 if the screen recording was not encrypted.

temp_dir

Default Value: Empty

Valid Values: Any valid path

Changes Take Effect: After restart

Specifies the absolute path to the directory for storing the temporary files.

save_temp_file

Default Value: 0

Valid Values: 0, 1

Changes Take Effect: After restart

Specifies whether to delete the temporary files when the processing is complete. If set to 0, the temporary files are deleted when no longer needed. If set to 1, the temporary files are not deleted.

[+] webDav Section

username

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the username used to allow read and write access to the WebDAV storage server.

password

Default Value: Empty

Valid Values: Any string

Changes Take Effect: After restart

Specifies the password used to allow read and write access to the WebDAV storage server..

trusted-ca

Default Value: false

Valid Values: true, false, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format.

Changes Take Effect: After restart

Configures TLS certificate validation when making a secure outbound connection to WebDAV. For more information, see [Configuring TLS connection to WebDAV](#).

[+] advanced Section

worker_threads

Default Value: 4

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the number of parallel processing threads to allow multiple call recordings and screen recording pairs to be combined in parallel.

pagination

Default Value: 100

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the number of records that are returned per page.

query_slice_size

Default Value: 100

Valid Values: < 0

Changes Take Effect: After restart

Defines the maximum number of call recording records whose screen recordings should be queried.

call_recording_extra_query_string

Default Value: If left empty, its value will be defaulted internally to `userData=SRSScreenRecordingStateStarted>anAndScroll=true`, if RWS version `>= 8.5.201.14`, otherwise, "" (empty string).

Valid Values: `callerPhoneNumber`, `dialedPhoneNumber`, `userName`, `userData`

Changes Take Effect: After restart

Specifies what the Recording Muxer Script is to query for with the Interaction Recording Web Services API (or Web Services API if you're using version 8.5.210.02 or earlier) along with `startTime` and/or `endTime` that are determined by `"window_past"` and/or `"window_past_older_than"` configurations. When non-empty (for example, `userData=SRSScreenRecordingStateStarted&userName=some_user`), the Recording Muxer Script is run against all the records found with the GET `<RWS_URL>/api/v2/ops/contact-centers/%s/recordings?startTime=%s&endTime=%s&<call_recording_extra_query_string>` query. If the result returns the `nextUri` attribute, the link will be followed with the records to run against. Once the records are traversed and processed, the Recording Muxer Script will exit.

call_recording_query_string

Default Value: Empty

Valid Values: `callerPhoneNumber`, `dialedPhoneNumber`, `startTime`, `endTime`, `userName`, `userData`

Changes Take Effect: After restart

Specifies what the Recording Muxer Script is to query for with the Interaction Recording Web Services API (or Web Services API if you're using version 8.5.210.02 or earlier). When non-empty (for example, `startTime=0&endTime=1000`), the Recording Muxer Script is run against all the records found with the GET `<RWS_URL>/api/v2/ops/contact-centers/%s/recordings?<call_recording_query_string>` query. If the result returns the `nextUri` attribute, the link will be followed with the records to run against. Once the records are traversed and processed, the Recording Muxer Script will exit. If left empty, the Recording Muxer Script periodically issues a query with the parameters generated internally to look at last N hours of records, combines the records, and never exits.

max_overlap_allowed

Default Value: 0

Valid Values: Any integer

Changes Take Effect: After restart

Specifies how much overlap time, in milliseconds, needs to occur before deciding to truncate the overlapping duration. When the muxed file's `startTime` and `stopTime` overlaps with the newly uploaded recording the muxed file's audio or video needs to be truncated to discard the previously silence/black filled duration.

video_padding_slice_length_ms

Default Value: 5000

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the duration, in milliseconds, of the prepended padded video slice. Set this parameter if the

video starts later than the audio.

mark_screen_recording_label

Default Value: 1

Valid Values: 1, 0

Changes Take Effect: After restart

If set to 1, Recording Muxer Script will automatically apply the label "screenRecording" to the associated call recording metadata after muxing.

[+] logfile Section

max_log_file_size_mb

Default Value: 21

Valid Values: Any integer

Changes Take Effect: After restart

Specifies the maximum size, in megabytes, of the Recording Muxer Script log file before starting a new log file. If set to 0, the Recording Muxer Script log file will not roll over.

logfile_path

Default Value: Empty

Valid Values: Any valid path

Changes Take Effect: After restart

Specifies the path to the log file. If left empty, the working directory is used.

level

Default Value: INFO

Valid Values: DEBUG, INFO, WARNING, ERROR, CRITICAL

Changes Take Effect: After restart

Specifies the level or severity of the events to track in the log file.

Migrate Genesys Interaction Recording from a Single Tenant to a Multi-Tenant Deployment

This page describes the deployment steps required to migrate Genesys Interaction Recording from a single-tenant deployment to a multi-tenant deployment.

Important

- Before performing the upgrade procedure, verify that a multi-tenant Configuration Server is deployed in your environment.
- The steps on this page should be only performed when you want to add a tenant to a deployment in which a single tenant was previously configured using the existing **Single Tenant** configuration instructions.
- All the steps on this page should be only performed for each new tenant. The existing installation and configuration will continue to be used for the existing tenant only.
- GIR does not support WWE when configured with a multi-tenant deployment.

To successfully migrate GIR from a single-tenant deployment to a multi-tenant deployment, you must perform the following procedures in the order presented:

For the following steps, please substitute:

- Tenant-specific RPS load balancer URL = <loadbalancer>/t1/rp
- Tenant-specific RCS load balancer URL = <loadbalancer>/t1/rcs
- Tenant-specific SMIR load balancer URL = <loadbalancer>/t1/interactionreceiver
- Tenant-specific WebDAV load balancer URL = <loadbalancer>/t1/recordings
- Tenant-specific Interaction Recording Web Services load balancer URL = <loadbalancer>/t1

...where <loadbalancer>/t1 refers to the load balancer which is described in [step 20 \(Load Balancing\)](#), and t1 is a tenant-specific identifier - subsequent tenants will use t2, t3, etc.

1. **Genesys Administrator Extension (GAX):** GAX is already installed in a single-tenant deployment. Therefore, when migrating from a single-tenant deployment to a multi-tenant deployment you do not have to install GAX.
2. **Interaction Recording Web Services (RWS)** (or **Web Services and Applications** if you're using version

8.5.210.02 or earlier)

In a multi-tenant deployment, each tenant must deploy a separate instance of Interaction Recording Web Services.

Perform the steps described on the [Interaction Recording Web Services \(RWS\) \(Web Services and Applications\)](#) pages.

When performing these steps, consider and perform the following instructions for each additional tenant added to the existing tenant:

WebDAV [+] **Show steps.**

- Each tenant must deploy a separate WebDAV Server instance. Follow the instructions in the [Deploy the WebDAV Storage Server](#) section.

Cassandra [+] **Show steps.**

Important

When migrating from a single-tenant to a multi-tenant deployment, you do not have to deploy Cassandra since it is already deployed in the single-tenant deployment. You only need to add a keyspace per tenant. The existing keyspace (**sipfs**) will continue to be used for the existing tenant only.

- Each tenant must have a separate keyspace for Interaction Recording Web Services (Web Services) on Cassandra. Follow the steps in either [Initializing Cassandra](#) or in the [Initializing Cassandra section in the Web Services and Applications Deployment Guide](#) (if you are using GIR version 8.5.210.02 or earlier) with the following exceptions:

CHANGED TEXT BEGINS

- When deploying the **ks-schema** file, replace the keyspace name from **sipfs** to a tenant-specific name such as **sipfs1** as follows:

```
CREATE KEYSPACE sipfs1 WITH replication = {'class': 'SimpleStrategy',
'replication_factor': '2'} AND durable_writes = true;
```

- After the keyspace is created, update the schema file **cf-schema.cql** by changing the first line from **sipfs** to the tenant-specific keyspace name **sipfs1** as follows:

```
CREATE TABLE sipfs1.accounts (
  key text,
  column1 text,
  value blob,
  PRIMARY KEY (key, column1)
) WITH COMPACT STORAGE
AND CLUSTERING ORDER BY (column1 ASC)
AND bloom_filter_fp_chance = 0.01
AND caching = '{"keys":"ALL", "rows_per_partition":"NONE"}'
AND comment = ''
AND compaction = {'class':
'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy'}
AND compression = {'sstable_compression':
'org.apache.cassandra.io.compress.SnappyCompressor'}
AND dclocal_read_repair_chance = 0.0
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.1
AND speculative_retry = 'NONE';
```

CHANGED TEXT ENDS

Elasticsearch **[+]** **Show steps.**

- In a multi-tenant deployment, the recommended deployment approach for Elasticsearch is a standalone Elasticsearch cluster shared across multiple tenants, with each tenant owning separate indexes. The minimum number of Elasticsearch nodes that should be deployed in a High Availability (HA) environment is 3. To deploy Elasticsearch properly for a multi-tenant deployment, refer to either the [Elasticsearch](#) section on the Configuring Features page or to the Prerequisites section in the Web Services and Applications Deployment Guide (if you're using GIR version 8.5.210.02 or earlier).

Interaction Recording Web Services **[+]** **Show steps.**

- You must create a separate Interaction Recording Web Services (Web Services) instance for each tenant (Note: Each instance can be a cluster of 2 or more High-Availability nodes):
1. If you are using version 8.5.210.02 or earlier, follow the steps in the [Installing](#) section of the *Web Services and Applications Deployment* guide. Otherwise, follow the instructions in the [Installing](#) page in this guide. **Note:** The RWS instructions are different depending on whether you are using Interaction Recording Web Services by itself, or Interaction Recording Web Services together with Web Services.
 2. You must create a separate Cluster application for each tenant (note that this is shared per tenant between Interaction Recording Web Services and Web Services if you are using both services), and a separate Node application per tenant per server instance.
 - For the Cluster application:
Change the name to `IRWS_Cluster_<tenant>` or `WS_Cluster_<tenant>`, depending for which service the cluster is being created, where `<tenant>` is the tenant name.
In the **Tenants** tab, click **Add**, select the tenant object that you want to configure for Interaction Recording Web Services (or Web Services) and click **OK**.
 - For the Node application:
For a standalone deployment of Interaction Recording Web Services, make the connection to the `IRWS_Cluster_<Tenant>` application and name the application `IRWS_Node_<Tenant>`, where `<tenant>` is the tenant name.
For Web Services (when using version 8.5.210.02 or earlier) or a deployment where Interaction Recording Web Services is being used together with Web Services, make the connection to the `WS_Cluster_<Tenant>` application and name the application `WS_Node_<Tenant>` (when using version 8.5.210.02 or earlier), or `IRWS_Node_<Tenant>` (when using Interaction Recording Web Services together with Web Services) where `<tenant>` is the tenant name.
 3. For each Interaction Recording Web Services (Web Services) tenant instance, the **Jetty genconfig/application.yaml** must have the following parameters:
 - `keyspace:` `<Cassandra keyspace assigned to this tenant as per "Create separate keyspace">`
 - `nodes:` `<the shared Cassandra nodes>`
 4. For each Interaction Recording Web Services (Web Services) tenant instance, the **Jetty genconfig/application.yaml** must have the following parameters (if you are using Web Services and Application version 8.5.201.09 or earlier modify the `server-settings.yaml` and not `application.yaml`):
 - `externalApiUrlV2:` `http://<tenant-specific Interaction Recording Web Services load balancer URL>/api/v2`
 - `internalApiUrlV2:` `http://<tenant-specific Interaction Recording Web Services load balancer URL>/internal-api`
-

- undocumentedExternalApiUrl: http://<tenant-specific Interaction Recording Web Services load balancer URL>/internal-api
 - applicationName: <the WS_Node_<Tenant> name assigned for this tenant>
 - crClusterName: <Elasticsearch cluster name as per cluster.name>
 - crossOriginSettings/allowedOrigins: <tenant-specific Interaction Recording Web Services Servers>, <tenant-specific SpeechMiner Web Servers>
5. Configure **Voice Recordings**: Follow the same instructions as for the single-tenant deployment. Whenever Interaction Recording Web Services server (Web Services server) is specified, replace it with the tenant-specific Interaction Recording Web Services server (Web Services server) instance. When reference is made to the storage credentials, use the tenant-specific WebDAV Server credentials as configured in the WebDAV Server section above.
 6. Configure **Screen Recordings**: Follow the same instructions as the single tenant instructions. Whenever Interaction Recording Web Services server (Web Services server) is specified replace it with the tenant-specific Interaction Recording Web Services server (Web Services server) instance. When reference is made to the storage credentials, use the tenant-specific WebDAV Server credentials as configured in the WebDAV Server section above.
3. **SIP Server**
- In a multi-tenant deployment, each tenant must deploy a separate SIP Server instance, each with its own tenant-specific Switch object.

Important

A new SIP Server application object must be created for each new tenant. The existing SIP Server will continue to be used for the existing tenant.

The GIR deployment instructions for SIP Server are the same as for a single tenant and must be performed for each additional tenant added to the existing tenant.

4. **Genesys Voice Platform (GVP)**
- GVP is a shared resource for all tenants. Follow the instructions in this link to deploy the **Resource Manager** and **Media Control Platform** that are shared for all tenants. When creating the tenant-specific **IVR** profile, change the steps as follows:

Important

A new voice platform profile must be created for each new tenant. The existing voice platform profile associated with the Environment tenant will be used for the existing tenant only.

[+] Show steps.

In **Step #1** on the **Genesys Voice Platform (GVP)** page:

- Under **GAX**, switch the view to the tenant you want to configure.
- Navigate to **Configuration > System > Configuration Manager** and under **Voice Platform**, select **Voice Platform Profiles** and click **New**.

In **Step #3** on the **Genesys Voice Platform (GVP)** page:

- Change **recordingclient.callrec_authorization** to **rp_username** and **re_password**.

In **Step #4** on the **Genesys Voice Platform (GVP)** page:

- In the **Recording** tab, following the same instructions, but change the following parameters with tenant-specific information:
 - Storage destination—`http://<Tenant-specific WebDAV load balancer URL>`
 - Storage HTTP Authorization header—`<Tenant WebDAV Server authorization>`
 - Recording Processor—URI—`http://<Tenant-specific RPS load balancer URL>/api/contact-centers/<Contact Center Domain Name>/recordings/`
 - SpeechMiner Interaction Receiver—`http://<Tenant-specific SMIR load balancer URL>`
 - SpeechMiner Interaction Receiver Authorization header—`<Tenant Interaction Receiver authorization>`

5. **Interaction Concentrator (ICON)**

When following the instructions on this page, use a tenant-specific instance of ICON and ICON DB.

Important

A new ICON instance and a new ICON DB instance must be created for each new tenant. The existing ICON and ICON DB instances associated with the Environment tenant will be used for the existing tenant only.

6. Depending on the component you are using between Voice Processor and Recording Processor Script, follow the instructions below:

Voice Processor

Follow the instructions on this page except for the following for each additional tenant.

Important

A new Voice Processor instance must be created for each new tenant. The new instance must be configured to reference the appropriate tenant specific component instances. The existing Voice Processor instance will be used for the existing tenant only.

- a. Deploy separate instances of the Voice Processor for each tenant.
- b. Configure the following settings in the **settings-override.yml**:

- **rwsBaseUri**
- **nodeRpsDb**

Recording Processor Script (RPS)

Follow the instructions on this page except for the following for each additional tenant.

Important

A new Recording Processor (RP) instance must be created for each new tenant. The new RP instance must be configured to reference the appropriate tenant specific component instances. The existing RP instance will be used for the existing tenant only.

- a. Deploy separate instances of the Recording Processor for each tenant.
 - b. Replace the following configuration parameters with tenant-specific instances of Interaction Recording Web Services (or Web Services and Applications if you're using version 8.5.210.02 or earlier) and ICON DB: **[+] Show parameters.**
 - `htcc.base_uri`—Set this to the Tenant-specific Interaction Recording Web Services load balancer URL.
 - `htcc.username`
 - `htcc.password`
 - `icon_db_servers`
7. **Recording Crypto Server (RCS)**
When deploying RCS, use a tenant-specific instance of RCS.

Important

A new RCS instance must be created for each new tenant. The existing RCS instance will be used for the existing tenant only.

Follow the instructions on this page for each additional tenant, except for the following:

[+] Show steps.

- In the **RCS application object**, replace the following parameters with tenant-specific instances of Interaction Recording Web Services (or Web Services and Applications if you're using version 8.5.210.02 or earlier):
 - `htcc.baseurl`—Set to the Tenant-specific Interaction Recording Web Services load balancer URL
 - `htcc.user`
 - `htcc.password`
 - `cors.allow-origins`—Set to the Tenant-specific Interaction Recording Web Services load balancer URL.
 - On the **Annex** tab for the specific tenant object:
 1. Create a new section called **Recording**.
 2. Create a parameter called: `rcsurl`.
 3. Set its value to the Tenant-specific RCS load balancer URL.
 4. In the `[recording.archive]` section, set `speechminerurl` = `<Tenant-specific SMIR load balancer URL>`.
8. **Recording Plug-in for GAX**
Execute the **Solution Deploy SPD** file for each additional tenant, to create the appropriate tenant

Access Groups, Roles, and Permissions. For additional information, refer to the [Example Solution SPD File](#) page.

Follow the instructions in the "Multi-Tenant Environment" subsection within the "Configure for Screen Recording" section.

9. Deploying Encryption

Follow the instructions on this page and perform the following for each additional tenant: **[+] Show steps.**

1. Upload tenant certificates:

- a. Log into GAX using a user account belonging to the tenant.
- b. Follow the instructions about how to configure the encryption of voice recordings in the [For Call Recordings](#) section.

3. For each tenant, configure the decryption proxy by following the steps in the [Setting up the Decryption Proxy](#) section.

Note: Replace **Web Services** and **RCS URI** with the tenant-specific addresses.

10. Screen Recording Service

Follow the instructions on this page.

When using the command line to install the SR Service, change the **/server** parameter in the setup to point to the Interaction Recording Web Services tenant instance. For example: `setup.exe /server=<Tenant-specific Interaction Recording Web Services load balancer URL>`.

11. Screen Recording Service - Advanced Configuration

Follow the instructions on this page for each additional tenant and replace the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) instance with an Interaction Recording Web Services (Web Services) tenant instance.

For WDE, follow [these WDE instructions](#) and set the following:

`screen-recording.htcc.uri`: <Tenant-specific Interaction Recording Web Services load balancer URL>

12. Recording Muxer Script

Follow the instructions on this page to deploy tenant-specific instances of Muxer to each additional tenant.

Replace the following parameters with tenant-specific values: **[+] Show parameters.**

- `htcc.base_uri`—Set this to the Tenant-specific Interaction Recording Web Services load balancer URL
- `htcc.contact_center_id`
- `htcc.username`
- `htcc.password`
- `rccs.base_uri`—Set set this to the Tenant-specific RCS load balancer URL.
- `rccs.username`
- `rccs.password`
- `webdav.username`
- `webdav.password`

13. Interaction Recording Options Reference

Refer to this page for a description of the configuration options.

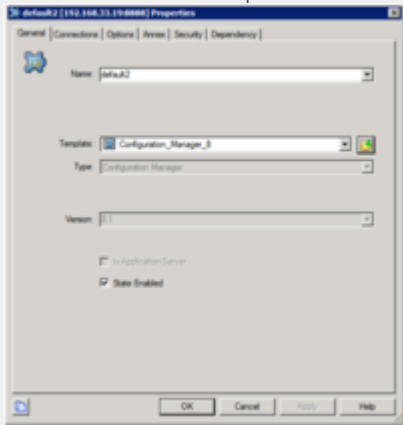
14. **User Access**

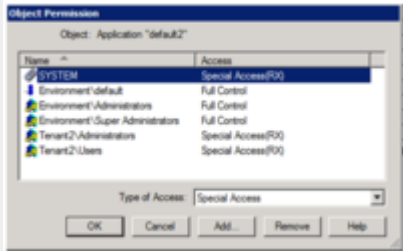
Follow the instructions on this page for each additional tenant and consider the following: **[+] Show notes.**

Environment Tenant

Deployment Steps	Description
1	<p>Create a new Configuration Manager object for the Environment tenant (used for SpeechMiner access), so that only users associated with the Environment tenant can access the SpeechMiner instance configured for the Environment tenant.</p> <div>Important Create a new Configuration Manager Application object for the Environment tenant for SpeechMiner access only if it does not already exist.</div> <p>The new Configuration Manager object must be restricted to Environment tenant users only (that is, Environment\Administrators and Environment\Users) and super administrators.</p>

Additional Tenants:

Deployment Steps	Description
1	<p>To restrict logging into the SpeechMiner UI, a new Configuration Manager object must be created for each tenant.</p> <p>Back up the default Configuration Manager object, since this object is accessible by all users from all tenants.</p> <p>For the tenant-specific Configuration Manager object, the permission must be restricted to tenant users (for example, TenantN\Administrators and TenantN\Users), and super administrators. For example for CME:</p> 

Deployment Steps	Description
	
2	The agent hierarchy for each tenant must be configured to ensure the agents for each tenant are on a different agent hierarchy. While the previous step prevents a user from connecting to a SpeechMiner UI associated with a tenant the specific user is not allowed to use, the agent hierarchy ensures a unified hierarchy for all users in the Management Framework. For example, recording.agent_hierarchy for users in Tenant 1 would start with /Tenant1. Sub-teams could be a child of /Tenant1.
3	Users of each tenant will be assigned to the appropriate Roles in each tenant created by the Recording SPD. The Environment tenant should not be expected to have the Roles created by the SPD.
4	Users of each tenant will be assigned to the appropriate Access Groups in each tenant (but not the Environment tenant). Although the Recording SPD always creates a "/" for each tenant, it is recommended (per step #2) to create a /TenantX Access Group object and assign it to the appropriate tenant users.

15. Speech and Text Analytics (SpeechMiner)

When following the instructions on this page, create a separate SpeechMiner instance for each additional tenant.

To do this, you must configure the following items for each tenant: **[+] Show items.**

Environment Tenant

SpeechMiner Server Configuration

Deployment Steps	Description
1	Run SMConfig for the environment tenant, and set the Database to the Environment tenant-specific SpeechMiner database instance.
2	On the Recording tab, change the User Application Name to be the name of the Configuration Manager application object used for SpeechMiner purposes for the Environment tenant, as per the User Access section above. Note: The default Configuration Manager

Deployment Steps	Description
	application object must no longer be used for this purpose.

Additional Tenants
SpeechMiner Database

Deployment Steps	Description
1	A shared SQL Server instance can be used by all SpeechMiner tenants. All SpeechMiner machines must be configured to connect to the same SQL Server and create a separate SpeechMiner database for each tenant.
2	A separate SpeechMiner database must be configured for each tenant. When installing the database portion of the SpeechMiner installation, create a new database for each tenant.

SQL Server Reporting Services

Deployment Steps	Description
1	A shared SQL Server Reporting Service can be used by all SpeechMiner tenants. All SpeechMiner machines must be configured to connect to the same SQL Server and create a separate SpeechMiner database for each tenant.

SpeechMiner Server Configuration

Deployment Steps	Description
1	<p>For each tenant, create separate application objects for SpeechMiner components:</p> <ul style="list-style-type: none">• <Speechminer prefix>_ClientApplications• <Speechminer prefix>_InteractionReceiver• <Speechminer prefix>_Platform• <Speechminer prefix>_Web <p>Where <Speechminer prefix> is a name assigned for the tenant. This name is used later in step #7.</p>
2	For each tenant in the SpeechMiner server, create separate UNC paths for the data directory. The data directory will hold files for Index, plus additional SpeechMiner files for the tenant.
3	After successfully installing the SpeechMiner software on each server, run SMConfig on each server, and set the Database to the tenant-specific SpeechMiner database instance. For

Deployment Steps	Description
	example, speechminer855_sm2 is the database name.
4	<p>On the Sites & Machines page:</p> <ul style="list-style-type: none">• Configure the paths to folders for the tenant created in step #2• Configure machines for this tenant for Web, Interaction Receiver, and Index• Configure the primary/backup configuration server IP:port
5	<p>On the Report Deployment page:</p> <ol style="list-style-type: none">1. Enter the server name for the shared SQL Server Reporting Services and deploy the MRSLibrary and reports for this tenant.2. After deploying the reports, use Internet Explorer to make sure the connection to the database source is configured. The default location is http://<SSRS server>/reports.3. Click on the database name that was chosen for this tenant.4. Click on SME to access the data source properties.5. Verify that the data source has a proper connection.6. Click Test Connection to validate, and click Apply to save the settings.
6	Follow existing deployment instructions for License, Services, Audio, and Index.
7	<p>On the Recording tab, each tenant has different settings:</p> <ul style="list-style-type: none">• Configuration• Tenant - The tenant name as per the configuration server.• Application Name - The SpeechMiner prefix for the application objects as per step #1.• Users Access Group - The tenant-specific Access Group that contains the list of SpeechMiner users for the tenant.• User Application Name - The name of the new Configuration Manager application object for the tenant as per step #1 in the Additional Tenants subsection of the User Access section above.

Deployment Steps	Description
	<ul style="list-style-type: none">• Interaction Receiver - The RP Authorization.• RP Authorization - Authorization for the tenant-specific RP.• Playback• RCS URI = <Tenant-specific RCS load balancer URL>• External RCS URI = <Tenant-specific RCS load balancer URL>• HTCC URL = <Tenant-specific Interaction Recording Web Services load balancer URL>

16. Security (TLS)

Follow the instructions on this page for each additional tenant.

17. Media Lifecycle Management

Follow the instructions on this page for each additional tenant and verify that the Interaction-Receiver settings group points to a tenant-specific Interaction Receiver. In the SpeechMiner section, use tenant Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and tenant Interaction Receiver instances. Set the SpeechMiner Interaction Receiver URL to the Tenant-specific SMIR load balancer URL when enabling Interaction Recording Web Services to contact Interaction Receiver.

18. Recording Storage Folder Hierarchy

Follow the instructions on this page for each additional tenant using a tenant WebDAV server instance.

19. Feature Configuration

Follow the instructions on this page for each additional tenant.

Note: **Audio Tones** are applicable on a per-tenant basis.

20. Load Balancing

Follow the instructions on this page.

Architecture and Features

The overall recording architecture is designed to adapt to different target solutions, while the core of the basic call flow remains consistent throughout all target solutions. Click on the following links to read more about the Genesys Interaction Recording (GIR) architecture and its features.

- [GIR Core Components](#)
- [Multi-Site Call Transfers](#)
- [Typical CallFlows](#)
- [The Recording Model](#)
- [Screen Recording](#)
- [Media Life Cycle Management](#)
- [Using Multiple Storage Locations for Screen Recording](#)
- [High Availability](#)
- [Security and Encryption](#)
- [Archiving and Metadata](#)
- [Access Control for Users](#)
- [How the T-Library Works](#)
- [User Interfaces](#)
- [Geo-Location](#)
- [AudioTones](#)

Genesys Interaction Recording Components

The following components are involved in the Genesys Interaction Recording architecture, and their roles are described in each of the solutions that follow.

- **Management Framework**—The foundation for all Genesys-based interaction management systems. Management Framework provides you with the following administration functions: Configuration, Access Control, Solution Control, Alarm Processing, Troubleshooting, and Fault Management.
- **Genesys Administrator and Genesys Administrator Extension**—The user-friendly interfaces that perform complex operations such as configuration, maintenance, and administrator of the contact center objects.
- **SIP Server**—The core competency of SIP Server is routing and call control, and SIP Server is responsible to initiate call recording by using media control to direct media towards Media Server.
- **Resource Manager**—A SIP Proxy that manages a pool of Genesys Media Servers and applies runtime policies such as ensuring call legs to the same conference are pinned to the same Media Server. Resource Manager also generates the call detail record (CDR) that allows correlation with individual call recordings.
- **Media Server**—Performs the actual file-based recording. Media Server is also responsible for negotiating the media between the endpoints, to minimize the need for transcoding, or to preserve security of the audio stream.
- **Reporting Server**—An optional component in this solution; this provides storage of CDR and call events for Resource Manager and Media Server, and provides a web service to provide the user (through Genesys Administrator) the ability to query CDR and other call event information.
- **Workspace Desktop Edition**—The smart-client application that provides agents and knowledge workers with non-intrusive access to the information, processes, and applications they need to perform their jobs more efficiently and to ensure increased customer satisfaction.
- **SpeechMiner**—SpeechMiner leverages recorded customer interactions (from any recording system) and analyzes each call for critical business topics and events. With unmatched accuracy, the system “listens” to conversations between customers and contact-center agents, precisely identifies the topics that were discussed, and categorizes what took place within each interaction. In the Genesys Interaction Recording solution context, the SpeechMiner analytics capability is disabled; however, it is limited to perform indexing, searching and playback of recordings.
- **Voice Processor**—A Node.js based microservice responsible for sending call recording metadata to the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner servers.
- **Recording Processor**—A Python script responsible for sending call recording metadata to the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) and SpeechMiner servers.
- **Recording Crypto Server**—Provides the Key Management System for the Genesys Interaction Recording solution.
- **Recording Plug-in for GAX**—Provides the user the ability to manage and administer recording certificates and policies.
- **Screen Recording Client**—Records screens at the direction of Interaction Recording Web Services

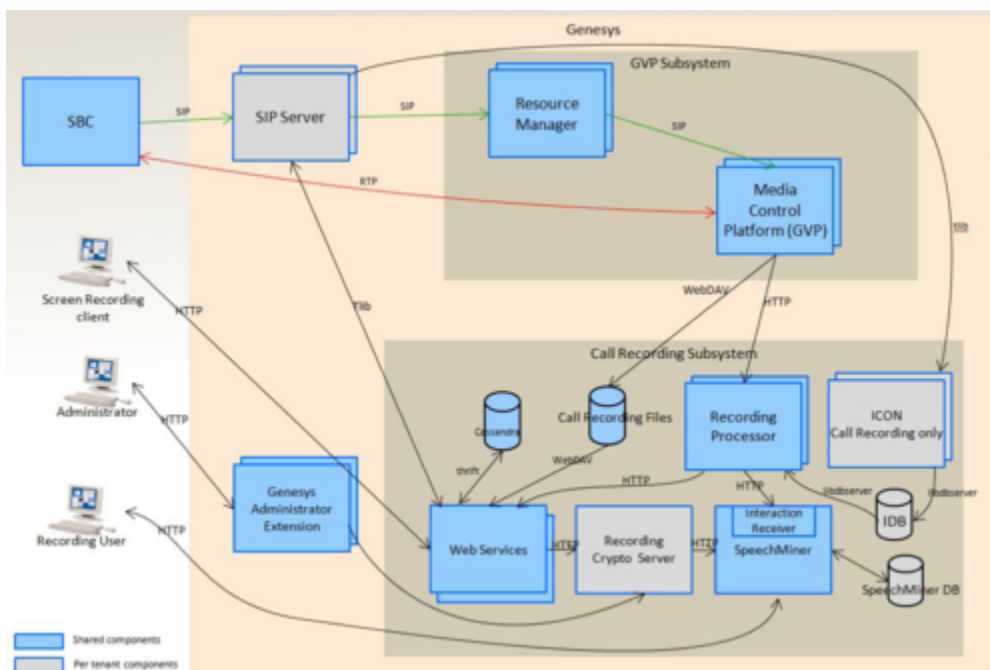
(Web Services), encrypts them using certificates indicated by Interaction Recording Web Services (Web Services), and then uploads them to Interaction Recording Web Services (Web Services).

- **Interaction Recording Web Services API (Web Services API)**—A REST API that provides a web client interface to access Genesys services.
- **Interaction Concentrator**—The server for the Interaction Database (or IDB) to store detailed reporting data from various sources in a contact center empowered with Genesys software.
- **Interaction Server**—The component that is responsible, in concert with the Routing components, to route interactions (non-voice) according to interaction workflows and routing strategies.

The following diagram illustrates these components in a premise deployment (click on the picture to increase the size):

Important

In the diagram below, the GIR Voice Processor can be used instead of the Recording Processor Script (RPS).



IVR Recording

The following table lists the DN types used with GVP and specifies how recording is supported for each DN type. There are two different ways to record a GVP-based IVR application:

- Record the entire IVR application, using the agent recording method with the SIP Server DN configuration.
- Record part or all of the IVR application with VoiceXML application-level control.

DN Type	Usage	Can be recorded using the agent recording method with SIP Server configuration	Can be recorded using VoiceXML recording control
VoIP Service DN (MSML)	PlayApplication treatment to execute a VoiceXML application.	No	Yes
Voice Treatment Port (VTP)	Legacy IVR ports for both inbound and outbound IVR calls	Yes	Yes
Trunk Group DN	Inbound GVP IVR calls and proactive notification (outbound GVP IVR calls)	Yes	Yes
Trunk DN	Inbound GVP IVR calls	Yes	Yes

Important

- Configuration of GVP is also required for IVR recording. For additional details, refer to the [IVR Profile](#).
- You must also configure the `sip-enable-ivr-metadata` option so that SIP Server can pass its Application name to MCP for IVR Recording. For details about configuring this option, see [Metadata Support for IVR Recording](#).

Controlling Recording with VoiceXML

The syntax to control recording within a VoiceXML application is described in [VoiceXML Syntax](#), and the scope of when recording occurs within an IVR application is described in [Recording Scope](#).

Important

This feature is applicable only for GIR and it is not applicable to environments with third-party recording solutions integrated with GVP.

VoiceXML Syntax

To control recording within a VoiceXML application, set the **dest** attribute of the **<log>** tag to **record**, anywhere that executable content is allowed. This value informs MCP that the contents of the **<log>** tag are commands, rather than text that should be logged. In the body of the tag, list the desired commands, delimited by semicolons. For example,

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns:gvp="http://<url>/vxml21-extension">
..
<log gvp:dest="record">
  <command>
    [additional commands;]
</log>
```

Where **<command>** can be one of the following:

Command	Description
start;	This command starts recording, or restarts call recording (in a new recording file), if it has already been started. All audio data for the call from this point forward will be recorded into the newly opened recording file. This will continue until the call terminates, or a subsequent start; command is issued, or until a stop; command is issued.
stop;	This command stops recording and terminates recordings in progress. Audio will be recorded up until the point that this command is executed. If no recording are in progress at the time, then this is a no-operation.
pause;	This command pauses the recording in progress. The recording file will continue to be recorded with silent audio until the recording is resumed (with a resume; command), stopped, or the call is terminated. If no recording is in progress, then this is a no-operation.
resume;	This command resumes a paused recording in progress. The recording file will no longer be recording silent audio. If no recording is in progress, then this is a no-operation.

Command	Description
additional commands;	parameter key=value can be used as additional commands.

Command	Description
	Allows VoiceXML to inject an additional media file parameter into the recording file. More than one parameter command can be used to include multiple parameters in the recording. key is inserted as a property in the parameters object of the recording media file metadata, with value as the value of the property.

Example

```
<?xml version="1.0"?>
<vxml version="2.1" xmlns:gvp="http://<url>/vxml21-extension" >
  <form>
    <var name="Result" expr="'success'"/>
    <block>
      <log >
        START RECORDING
      </log>
    </block>
    <block>
      <prompt>
        Prompt from external vxml - starting recording now
      </prompt>
    </block>
    <block>
      <log gvp:dest="record">
        start;
      </log>
    </block>
    <block>
      <log >
        RECORDING STARTED
      </log>
    </block>

    <block>
      <prompt>
        Prompt from external vxml - recording started now
      </prompt>
    </block>

    <block>
      <return namelist="Result"/>
    </block>

  </form>
  <catch event="connection.disconnect.hangup telephone.disconnect.hangup" >
    <assign name="Result" expr="'hangup'"/>
    <log>Message is <value expr="typeof(_message)"/>.</log>
    <if cond="typeof(_message) == 'undefined' || _message == null || _message ==
'undefined' || (!_message)">
      <assign name="ResultDetail" expr="_event"/>
    <else/>
      <assign name="ResultDetail" expr="_event + ' - ' + _message"/>
    </if>
    <return namelist="Result ResultDetail ReturnedIntent ReturnedConfidenceScore
ReturnedUtterance"/>
  </catch>
  <catch event="error" >
    <if cond="typeof(_message) == 'undefined' || _message == null">
```

```

                <assign name="ResultDetail" expr="_event"/>
            <else/>
                <assign name="ResultDetail" expr="_event + ' - ' + _message"/>
            </if>
            <assign name="Result" expr="'error'"/>
            <return namelist="Result ResultDetail ReturnedIntent ReturnedConfidenceScore
ReturnedUtterance"/>
        </catch>
    </vxml>

```

Recording Scope

The scope of the recording is within the scope of the VoiceXML session, and this is different from Full Call Recording using the agent recording method with the SIP Server DN configuration, where the scope of the recording is limited to the connection on the MCP. When the VoiceXML session terminates or **<exit>** or **<disconnect>** tags are called, the recording session is also considered terminated.

Each PlayApplication treatment will cause the MCP to generate a separate recording file. The recording files will be grouped together automatically by GIR as they share the same CallUUID.

The recording will be recorded from the perspective of the caller. It will follow the convention of the IVR Profile recording for mono or stereo recording. In stereo recording, the channels represent what the caller hears and what the caller says.

Access Control Considerations

GIR provides access control for recording files (refer to [Agent Hierarchy](#)), to allow recording files to be accessible only to specific GIR users. To enforce access control, each recording file is provided with an agent hierarchy or a set of partitions. Users must be a member of an Access Group with a matching agent hierarchy or partition to search and playback the recording.

IVR recordings have no agent hierarchy (as the IVR is not a user), and by default they have no partitions. This means that by default only users in the root access group (that have access to all recordings), or the default access group can access IVR recordings.

There are two ways to apply additional access control to IVR recordings:

1. By attaching the data key `GRECORD_PARTITIONS` to the call. In this case, all media files for the call (that is, the IVR segments and any agent segments) will receive the same set of partitions set in the attached data. For additional information, refer to the [Partitions](#) section.
2. You can individually apply a partition for a VoiceXML recording by using the additional command **parameter** and specifying the key **partitions**, with a value set to a comma-delimited list of partitions to apply to the recording. For example, to set partitions **/sales** and **/support** in VoiceXML:

```

// <?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
    <form>
        <block>
            <log gvp:dest="record">
                start;
                parameter partitions=/sales,/support;
            </log>

```

```
    </block>
    <!-- ...more VoiceXML code -->
</form>
</vxml>
```

Multi-Site Call Transfers

The Genesys Interaction Recording solution supports multi-site deployments with multiple SIP Servers and T-Servers in the call flow. For calls that are transferred across multiple switches, the solution has the ability to group together the segments of the call from different switches into a single recording.

From the user interface, the segments are grouped as a single recording, and each recording file can be played separately.

From the archive perspective, multiple segments of a call across switches will appear as a single recording-id, and each media file may refer to a different calluid.

For more information, see the [Multi-Site Call Scenarios](#) section in the *Genesys Events and Models Reference Manual*.

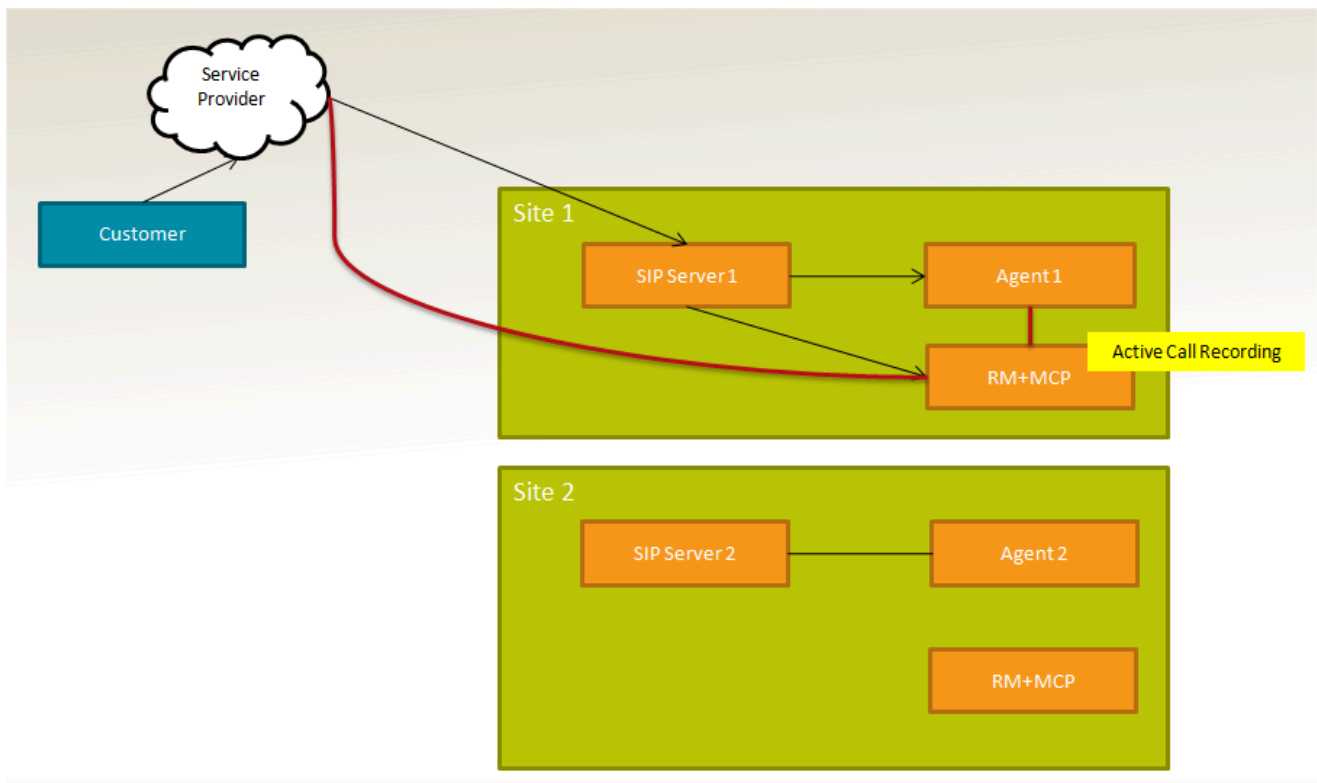
Multi-site Deployment Models

This section describes some of the multi-site deployment models that the Genesys Interaction Recording solution supports.

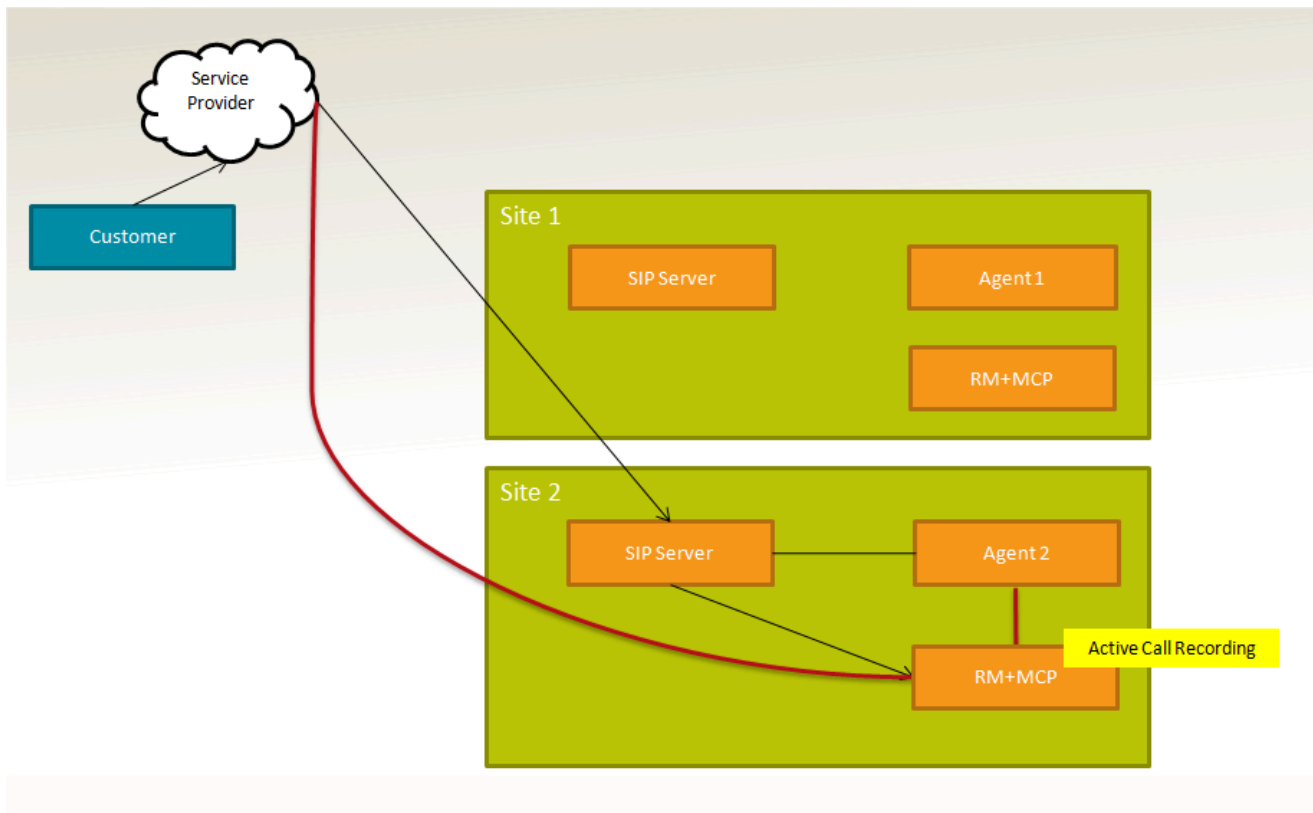
Tip

Click the diagrams to make them bigger and easier to read.

SIP Server to SIP Server Transfer/Conference



For premise deployments, multiple SIP Servers can be deployed in different locations. Each SIP Server queues the calls and distributes them to the agent. Recording is performed on the local SIP Server where the agent resides. The agent transfers the call to a Routing Point on the local SIP Server. The call is then distributed to an agent on another SIP Server.



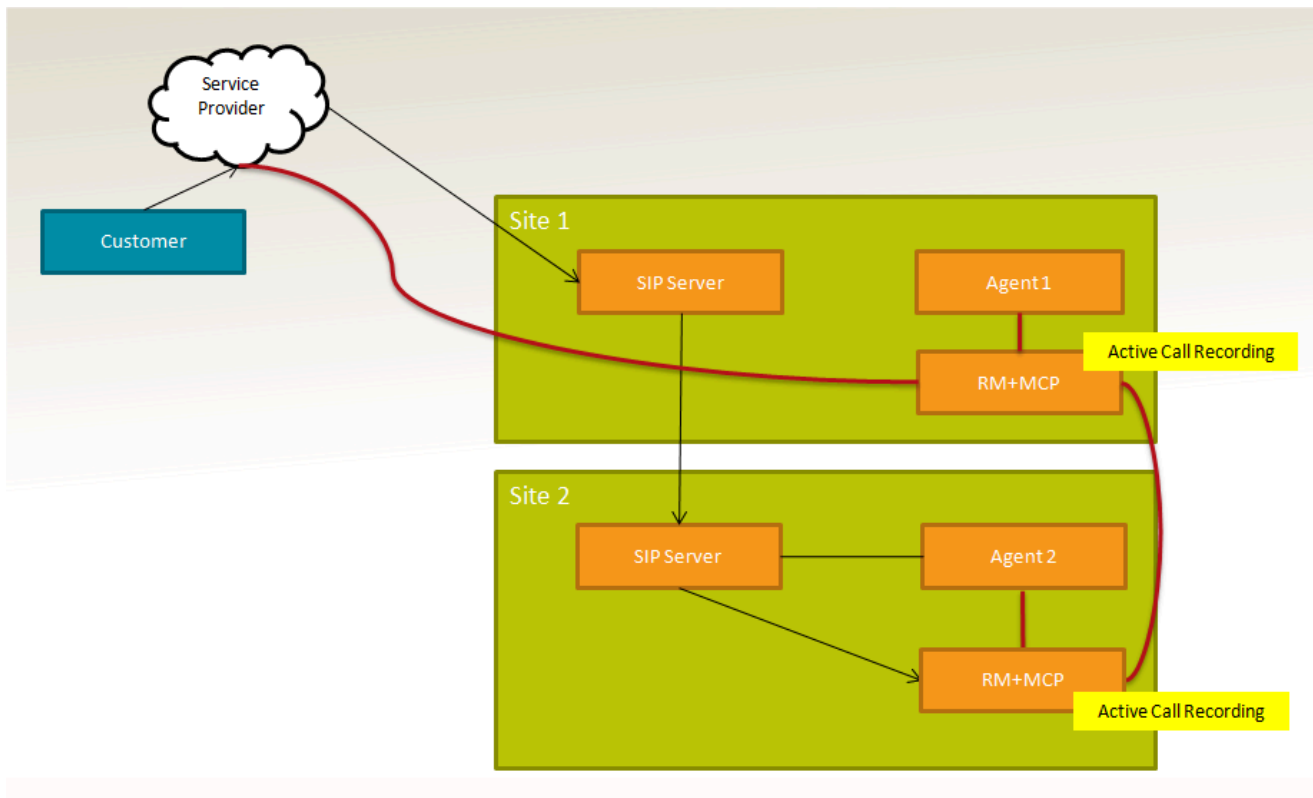
Assume that Agent1 is being recorded for all transfer cases.

Single-Step Transfer

When Agent1 transfers the call, the call is transferred to a Routing Point on local SIP Server1. The recording on Agent1 stops after the call is transferred. If Agent2 is configured to be recorded, the recording starts on Site 2. At the end of the call, there will be two recording files grouped together as a single recording.

Consult (Two-Step) Transfer

When Agent1 transfers the call, the call is transferred to a Routing Point on local SIP Server1. The call is distributed to Agent2 on SIP Server2. The recording starts on SIP Server2 for Agent2 (Agent1 and Agent2). Agent1 completes the transfer and SIP Server1 is removed from the signaling loop. The recording continues on Agent2 (Customer and Agent 2).



Single-Step Conference

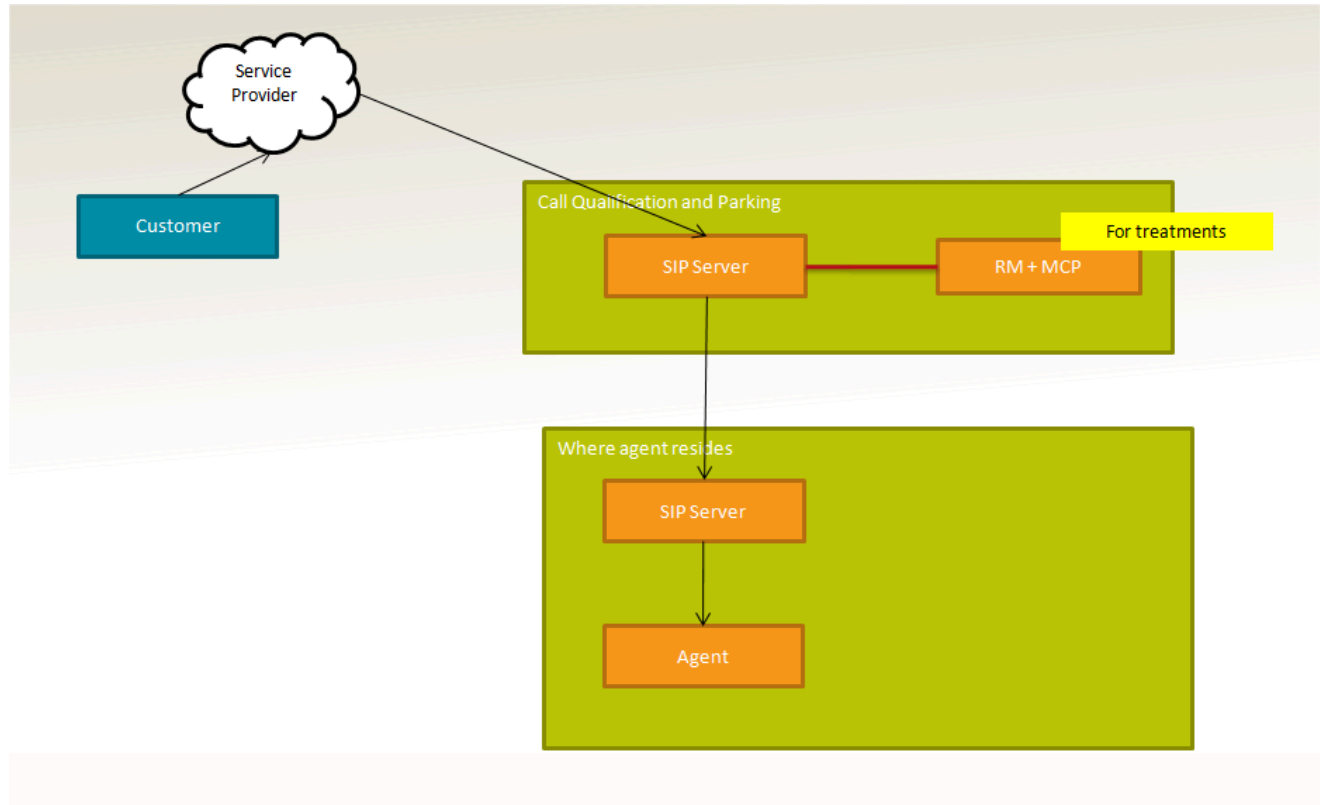
When Agent1 conferences a call, the call is transferred to a Routing Point on local SIP Server1. The call is distributed to Agent2 on SIP Server2. The call is conferenced through Media Control Platform (MCP) on SIP Server1. Recording starts on SIP Server2 for Agent2 (Customer, Agent1, and Agent2). Recording on Agent1 remains on SIP Server1 (Customer, Agent1, and Agent2). The recording remains on Agent2 on SIP Server2 (Customer and Agent2).

Two-Step Conference

The end result is similar to the two-step transfer. The recording for Agent2 remains on SIP Server2 after Agent 1 drops out of the call (Customer and Agent2).

Call Qualification

Centralized Call Qualification and Parking



In this scenario there are two SIP Servers:

- A SIP Server is dedicated for queuing.
- A SIP Server or T-Server for handling agents.

Calls are always transferred to the queuing SIP Server to wait for an agent, where media server applies treatments. When the agent is ready, the call is transferred to SIP/T-Server to the agent. The SIP Server for queuing can add recording functionality to record the call, thus providing a centralized recording solution instead of putting recording function at the agent SIP Server.

Call Flows

This section describes the following call flow types:

Tip

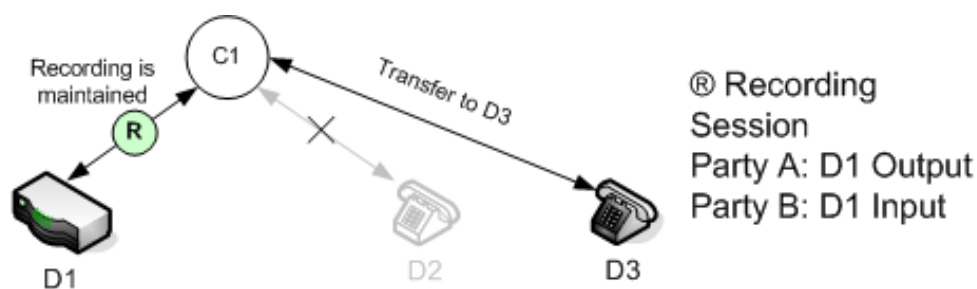
Click on the diagram to enlarge it.

Transfers

Transferring Calls

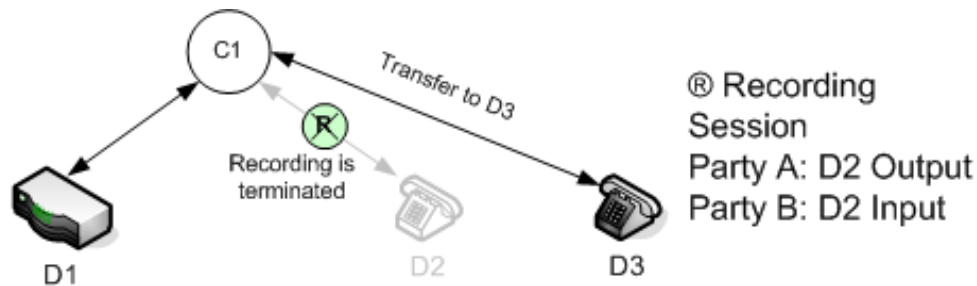
This section describes the two types of transfer (from the origination device, and from the termination device) that the Genesys Interaction Recording solution supports.

Recording from the Origination Device



When a call that is being recorded is transferred to another party, the recording can be affected differently depending on where the recording is initiated. The reason for differentiating the side of the connection is that call recording is "sticky" on the side of the connection that is chosen for recording. When the connection needs to transfer the call to another device, the call recording stays with the device. For example, if the connection is transferred from D2 to D3, the call recording is maintained if recording is initiated from the origination device, while the recording is terminated if the recording is initiated from the terminating device.

Recording from the Termination Device

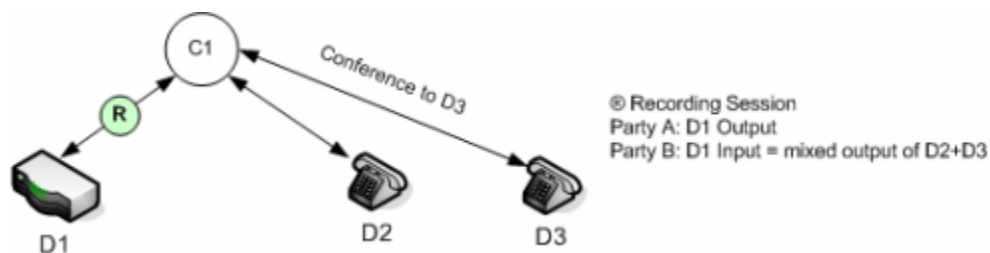


When the call is transferred to D3, the Recording Session is maintained and should expect a reINVITE to re-negotiate the media between D1 and D3. The media control dialog between SIP Server and Media Server is also maintained by only sending reINVITEs to the media control dialog.

Conferencing Calls

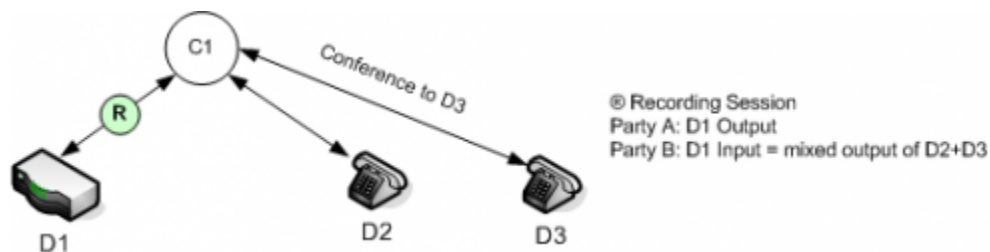
When a conference or monitoring supervisor requests to start recording, a participant can request the conference to be recorded through the recording service. You can reuse the existing SIP Server call structure and treat recording and conference as separate services, and the participant requesting recording will be recording the output of the conference.

Recording Conference



The recording is a mixed output of the customer and agent, plus a separate stream of the supervisor.

SIP Dialog



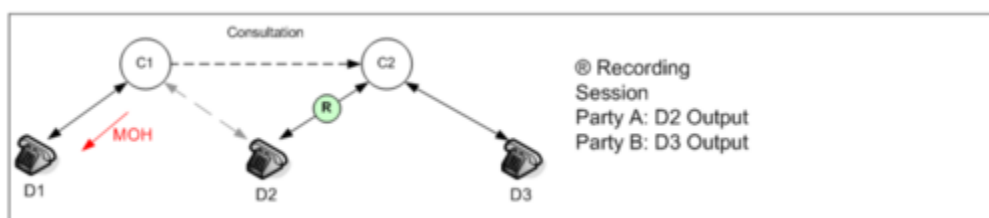
The structure of the SIP dialog when recording a conference.

Consultations

Consulting Calls

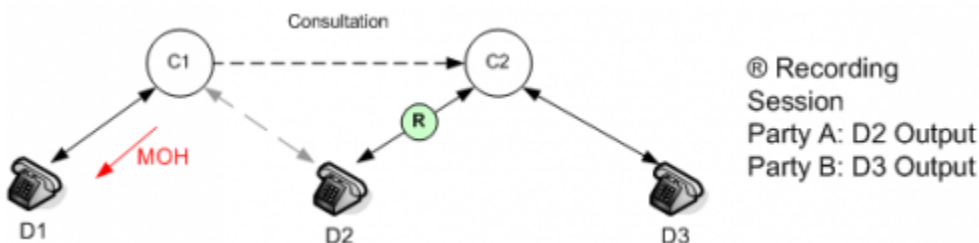
When the agent initiates a consultation call and call recording is enabled on the agent DN, the call recording to record the consultation session as well is allowed. This is recognized as a single-dialog consultation mode where there is only a single active SIP dialog on the device. Set the DN record-consult-calls option to true to allow recording of consultation calls. The following diagrams illustrate this scenario.

Before Consultation



The initial call when the customer (D1) is talking to the agent (D2).

During Consultation



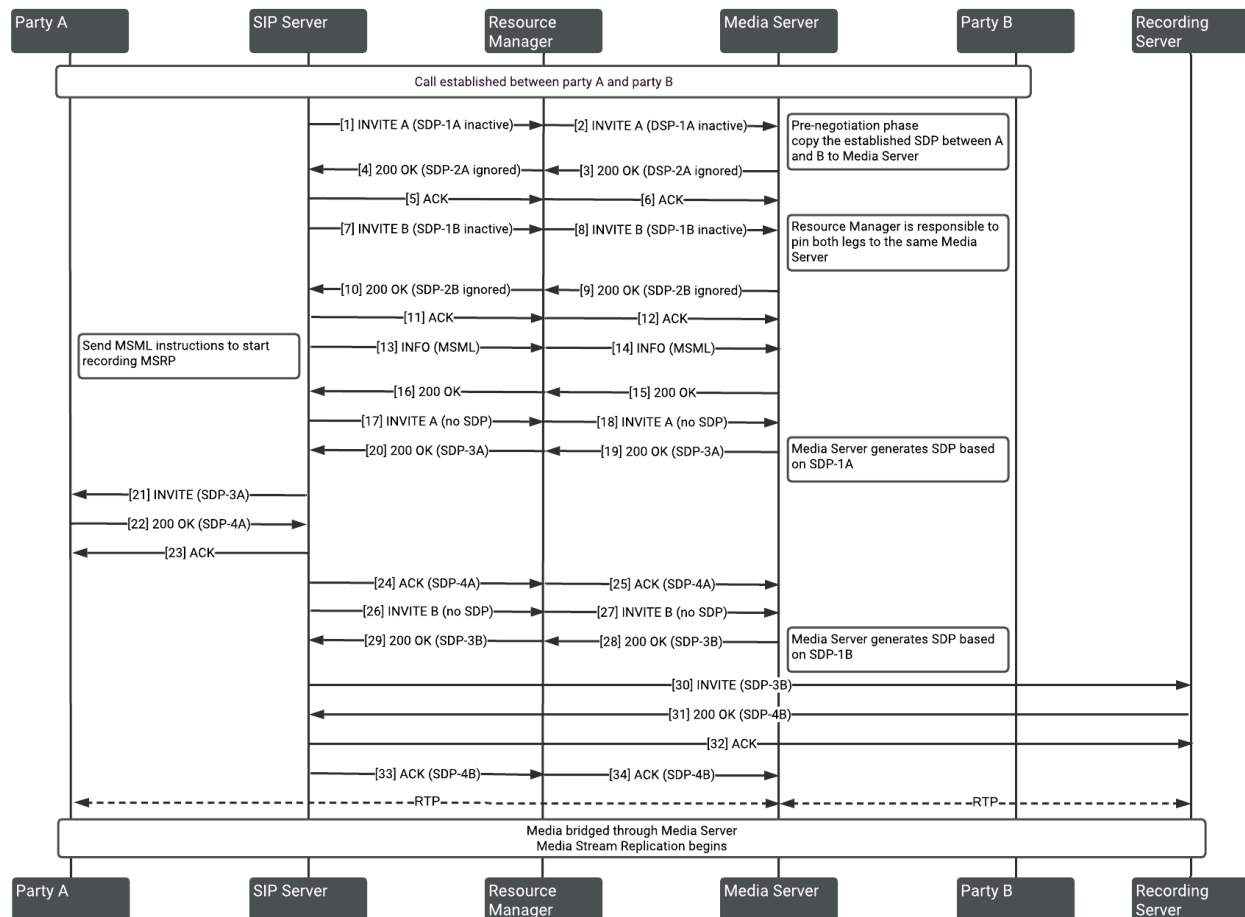
When the agent initiates a consultation to the supervisor (D3), the existing SIP dialog is retained and so is the Recording Session.

Important

As a current limitation for consultation calls, recording is not available on the consulted party, so a Recording Session cannot be started on D3.

Basic

Basic Call Flow



After Party A and Party B are connected and a recording request is made to SIP Server, SIP Server initiates two sessions, one session for each party, to Media Server. SIP Server first INVITEs with the Session Description Protocol (SDP) offer from the connected parties to Media Server, and a second reINVITE to Media Server to get an SDP offer from Media Server. The offer from SIP Server is sent to the connected parties to proxy the media through Media Server. Once the media is established, Media Server bridges the media between the parties and writes the recording to a file on the disk. The Recording Server fetches and indexes the recording after the call completes.

The Call Recording Model

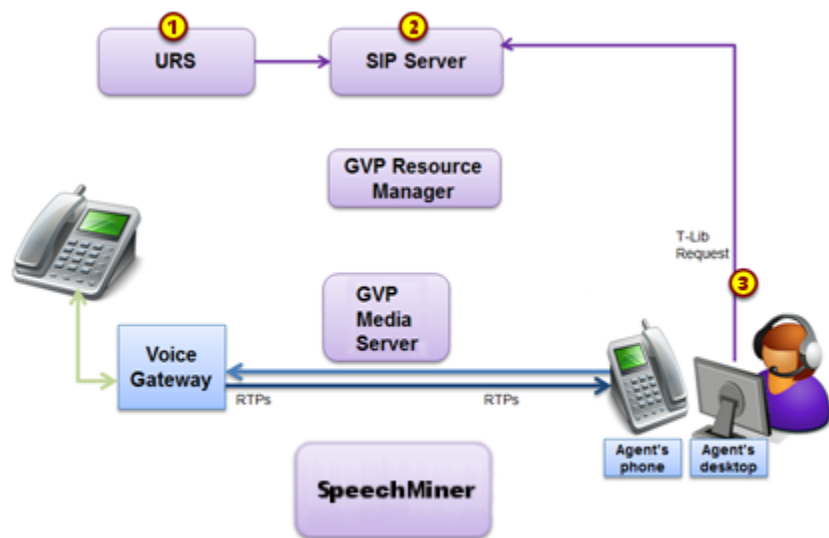
This section describes how call recording works. It also provides an overview of the recording model.

Call Recording proceeds in three stages:

- Initiating
- Progressing
- Ending

as described in the following sections.

Call Recording Initiated



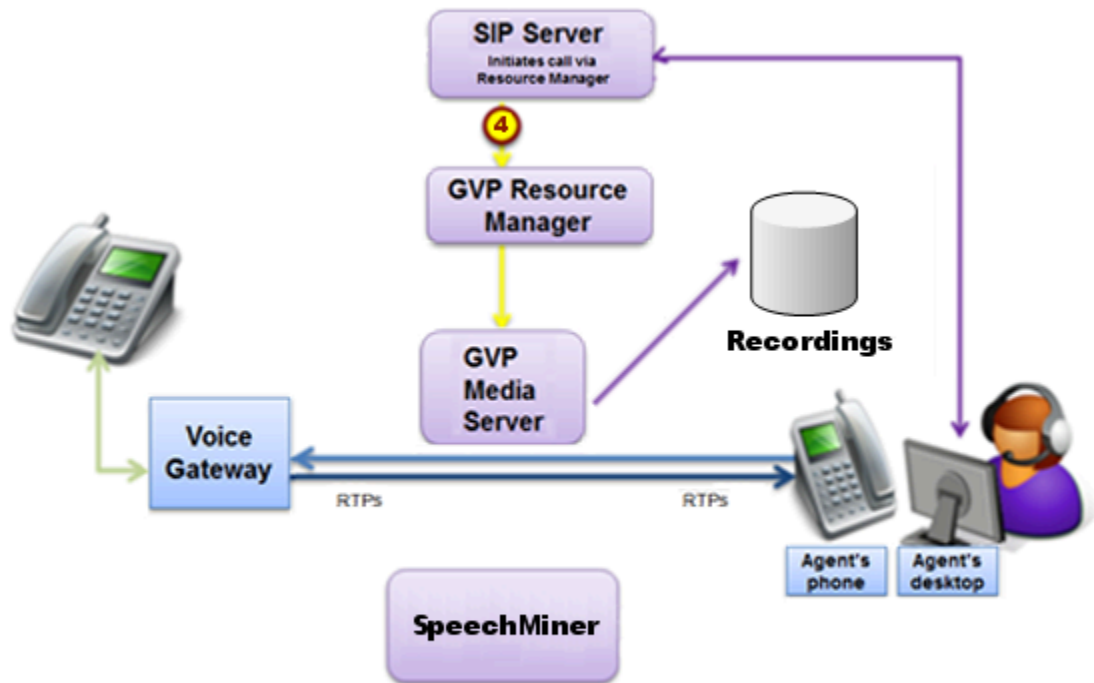
There are three ways that recording can be initiated. Once started, the recording process is the same, regardless of how it was initiated.

One of the following initiates recording:

- A recording can be initiated by a Routing Strategy (1).
- SIP Server initiates full time recording based on DN configuration (2).

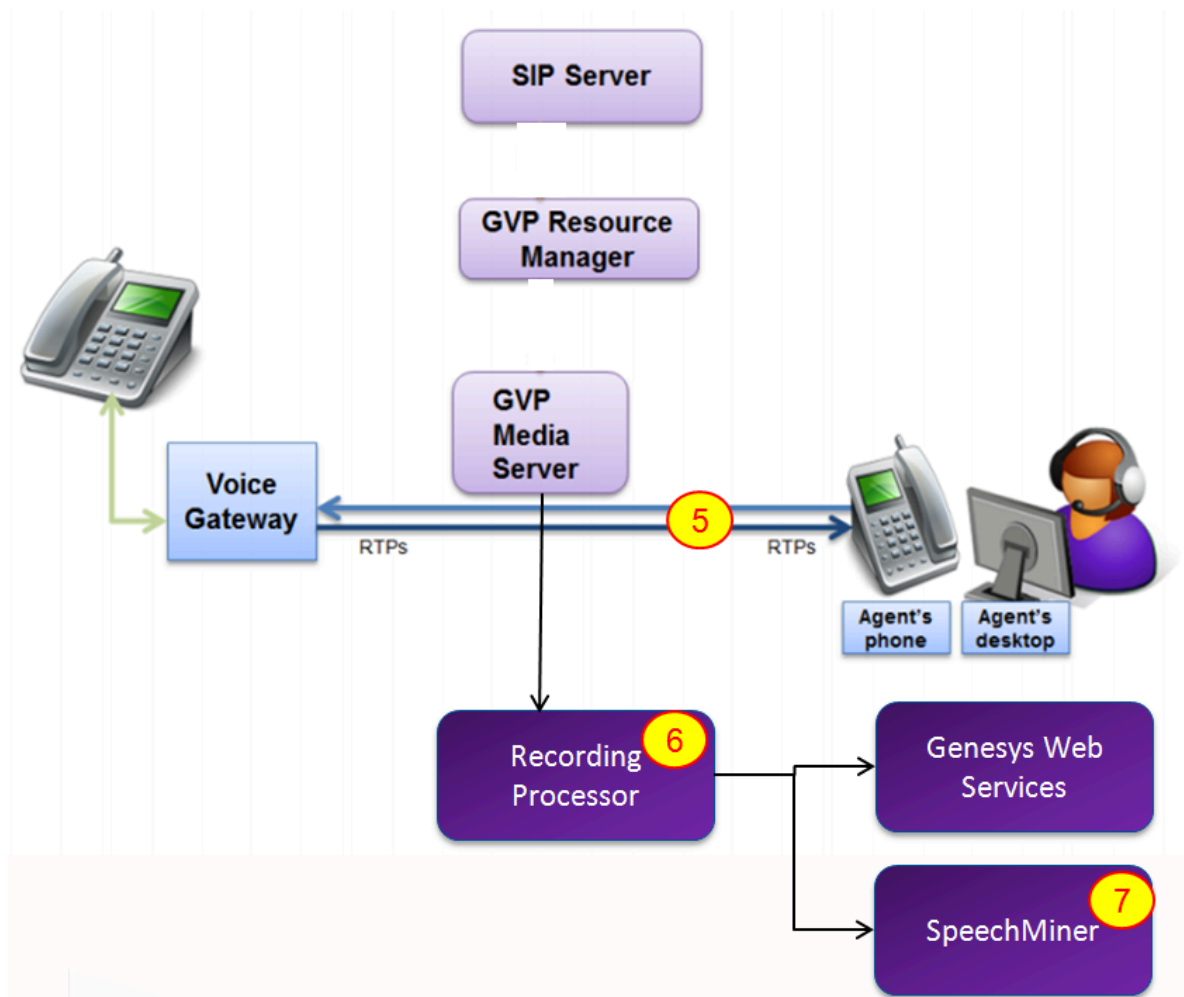
- Agent Desktop requests call recording through Interaction Workspace (3).

Recording in Progress



Using media control, SIP Server invites Media Server to bridge the media path between parties and at the same time record the call to a file (4).

Recording Ends



When the recording ends, SIP Server re-invites the session to have media sent directly between the parties. Media Server submits the recording to storage and to the Recording Processor (5).

Recording Processor then consolidates call events about the recording (for example, across switches) and submits the information to SpeechMiner and Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) (6).

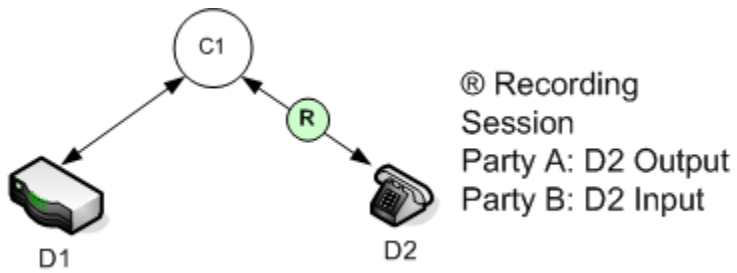
After indexing has been completed by SpeechMiner, the recording is made available through the SpeechMiner interface (7).

Call Recording Model

The basic model for initiating call recording is based on the SIP Server connection model. Recording is

initiated from the termination device; it is known as agent-side recording:

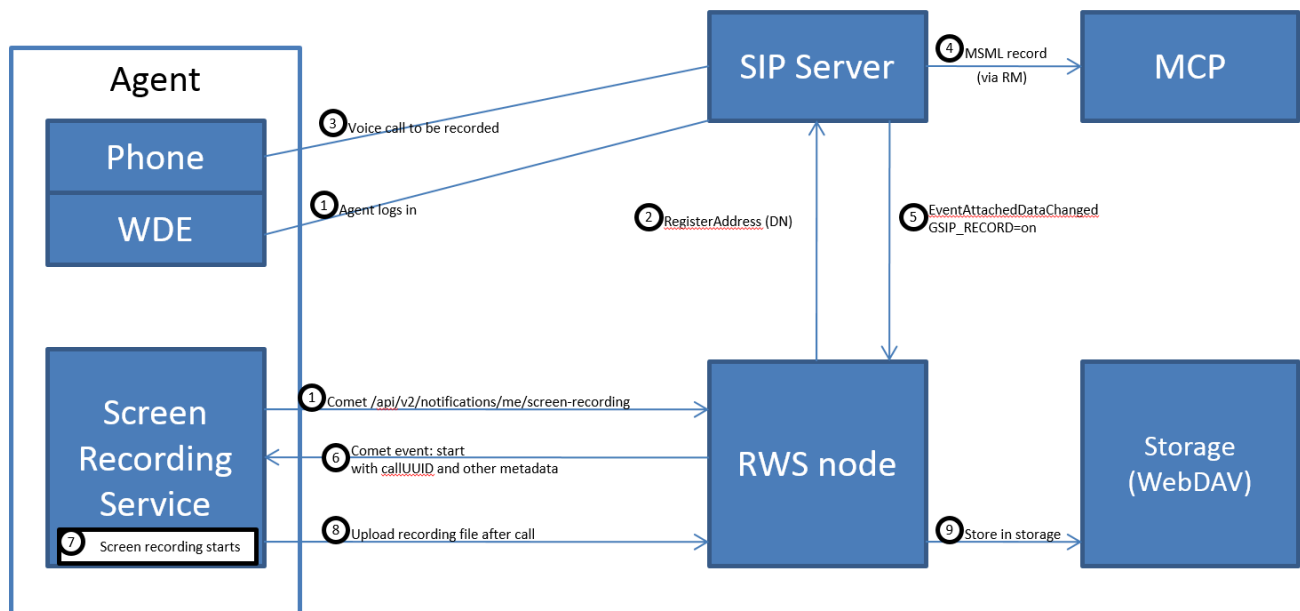
Agent-side Recording



Screen Recording Architecture

The Screen Recording Service integrates directly with Genesys Workspace Desktop Edition (WDE) and Genesys Workspace Web Edition (WWE). It can also be integrated with other agent applications – refer to [Screen Recording Service API](#) for further information on the integration mechanism. For additional information, see [Deploying the Screen Recording Service](#).

The following diagram illustrates Screen Recording architecture in a WDE integration:



1. The Agent logs into a PC where the Screen Recording Service is running as a Windows Service.
2. Interaction Recording Web Services (Web Services) registers the Agent's DN with SIP Server so that it will receive event notifications associated with that DN.
3. The Agent is engaged in a voice call for which recording is initiated.
4. SIP Server invokes Media Control Platform (MCP) to record the voice (audio) portion of the interaction.
5. SIP Server sends an event notification to Interaction Recording Web Services (Web Services) that recording has been initiated for the Agent's DN.
6. Interaction Recording Web Services (Web Services) determines that the Agent's screen is to be recorded and invokes the Screen Recording Service.
7. The Screen Recording Service records the Agent's screen.
8. After the call ends, the Screen Recording Service uploads a file of the recording to Interaction Recording Web Services (Web Services).
9. Interaction Recording Web Services (Web Services) stores the file in the recording storage location.

Mapping Screen Recordings to Interactions

Voice Interactions

As the Screen Recording Service can only generate one screen recording file at a time, the solution will map the screen recording file for the first voice call started for that agent (file with matching Agent ID).

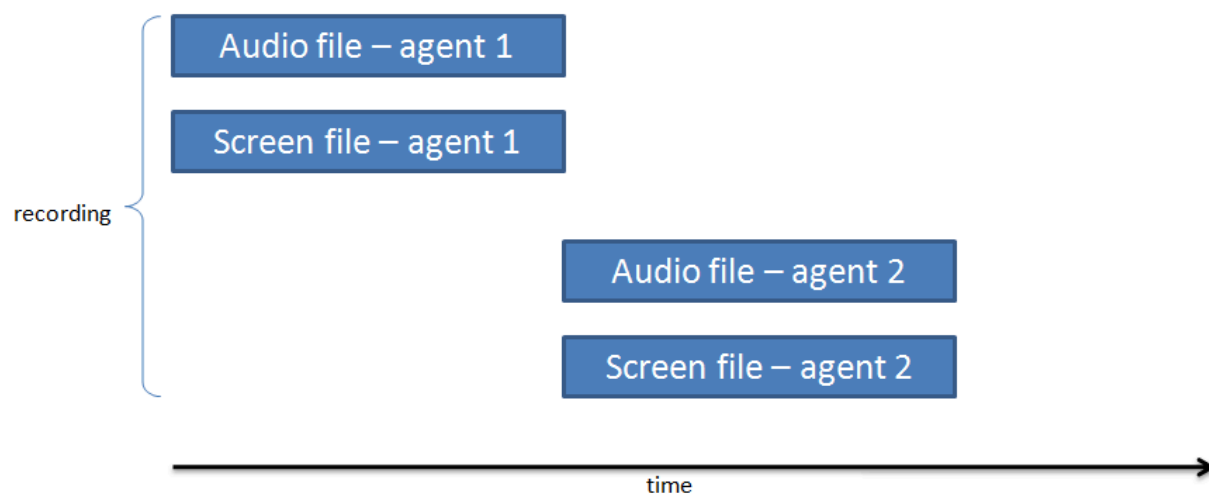
The following shows the transfer scenarios of how the screen recording is mapped to the voice calls.

Tip

Click on the diagrams to enlarge them.

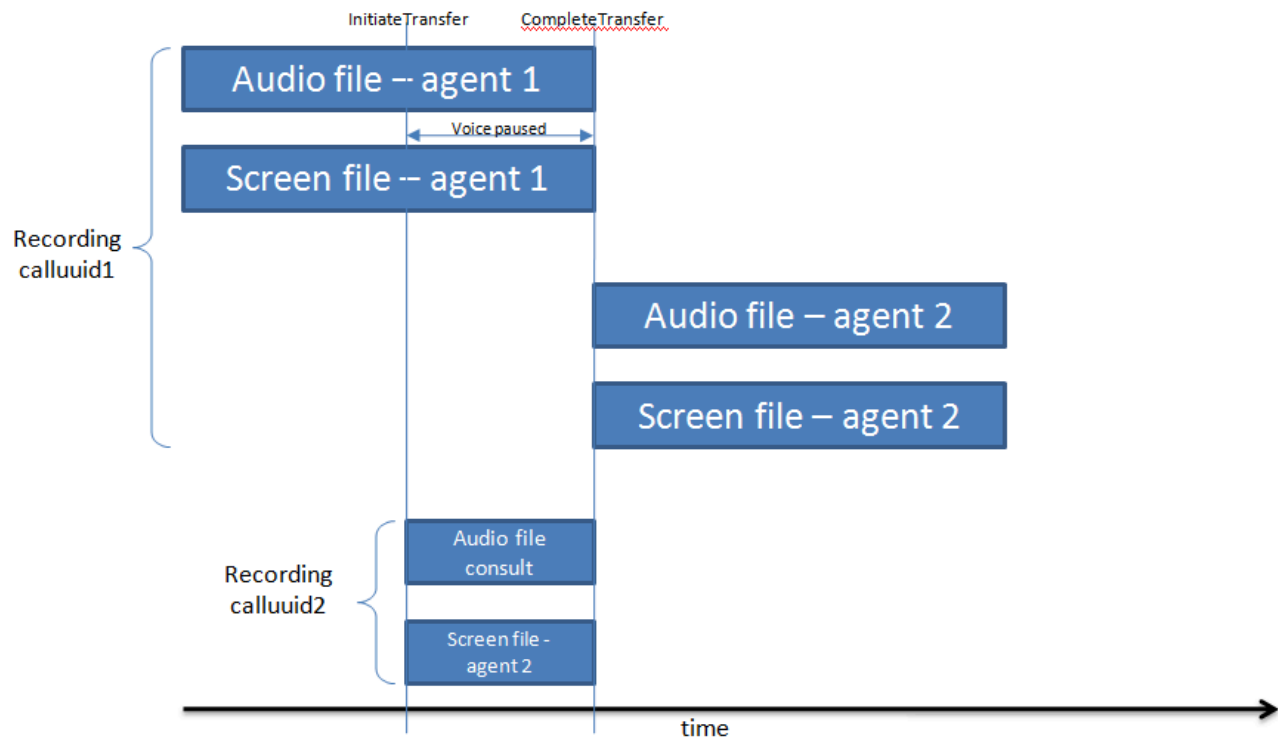
Single-Step Transfer

A simple customer-to-agent scenario:



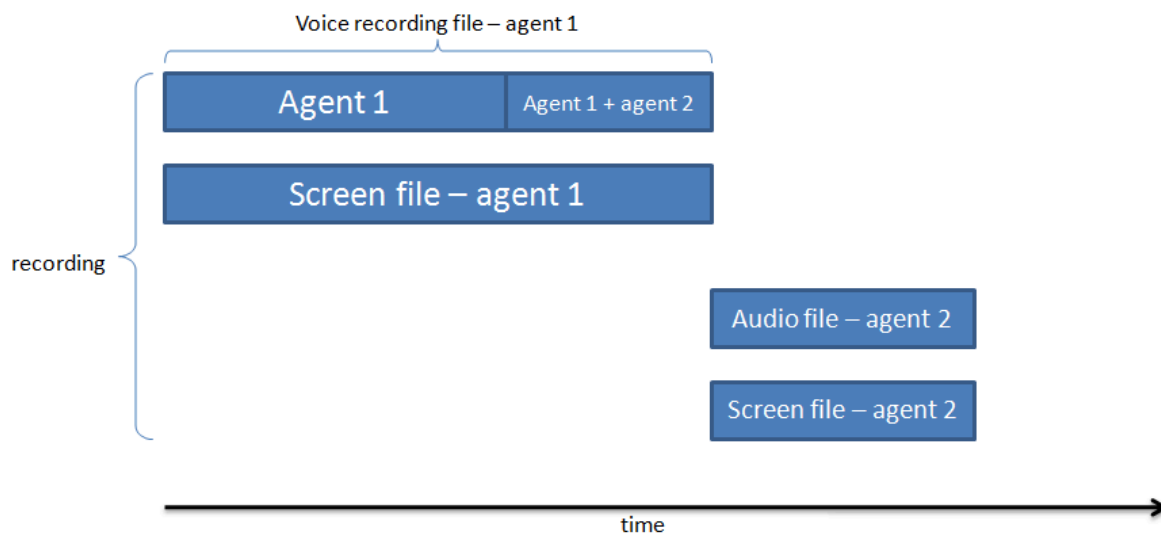
Two-Step Transfer

Two recordings with a separate recording during the consultation leg of the interaction:



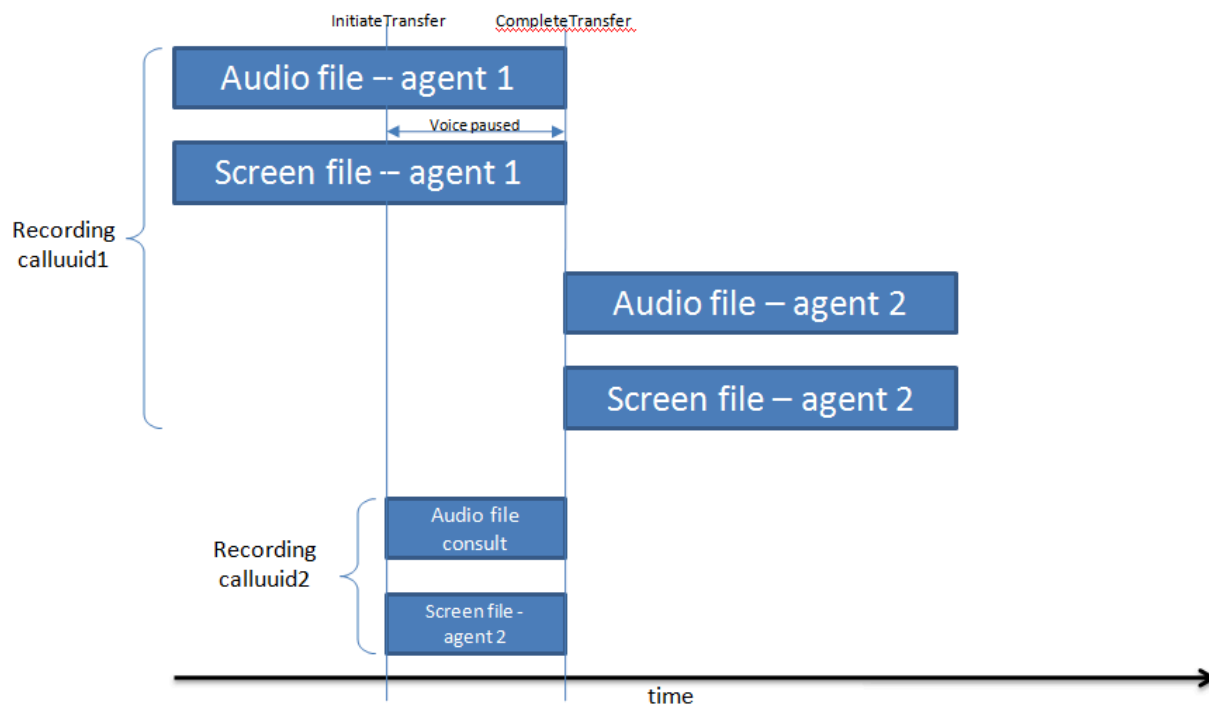
Single-Step Conference

The recording contains two segments:



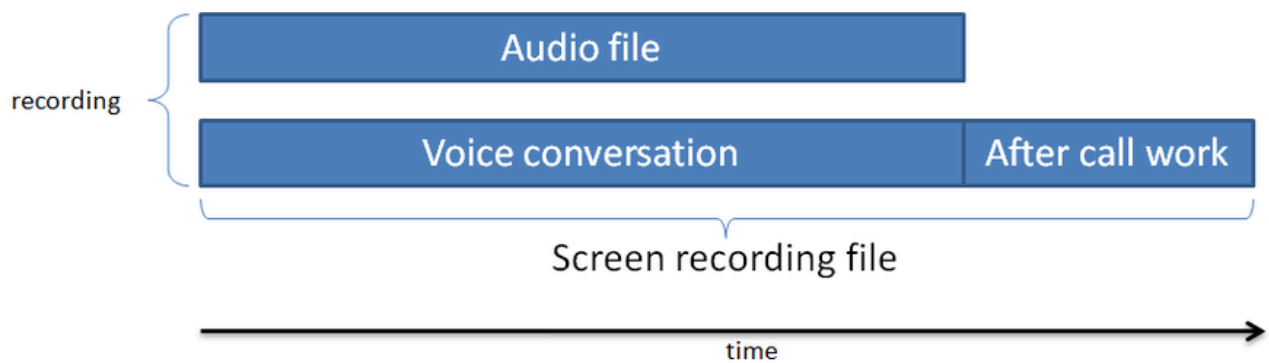
Two-Step Conference

Two recordings with a separate recording during the consultation leg of the interaction:



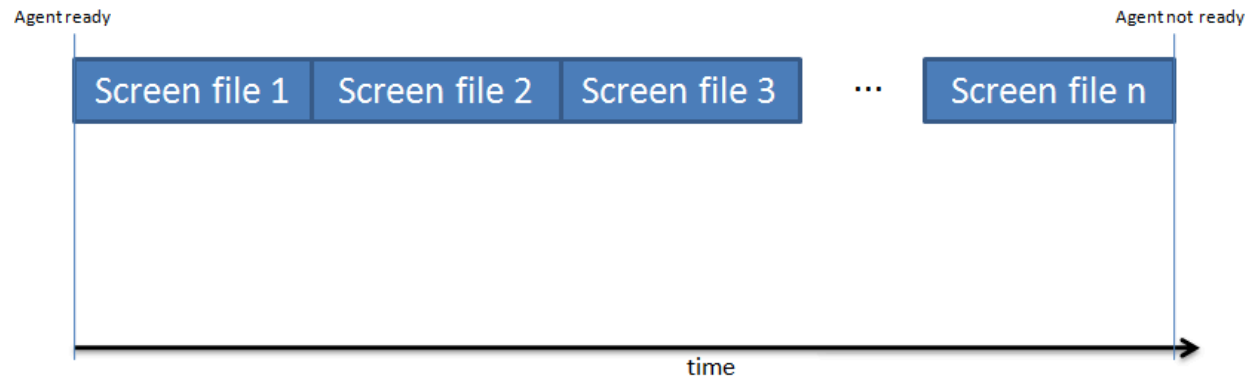
After Call Work

Single segment with after call work:



Non-voice Interactions

The following diagram illustrates the non-voice interaction recording started based on the configurable agent's state:



Multiple data center locations

Genesys Interaction Recording (GIR) can be deployed in multiple data center locations.

Before you can deploy GIR in multiple data center locations, you must consider the following:

- Recording storage locations
- Recording processing
- User interface
- Encryption management
- Media lifecycle management

Recording storage locations

Recording files can be saved in separate storage locations. The location of the recording files is directly related to the location of the agent handling the recording, so that compliance or bandwidth requirements between data centers is met. To enable each data center to record an interaction, each data center must be deployed with its own Media Control Platform (MCP) instance. Consider the following examples:

- There are data centers throughout the country where the agents reside, and Genesys server components are located in those data centers. To save network bandwidth between the data centers, all recording files must be stored in the same data center in which the recordings occur. A recording file is only transported to another data center whenever a client requests it for media file playback.
- There are data centers in different countries where the agents handle interactions. There are compliance requirements to ensure that interactions arriving in a country are recorded and stored in that country's data center only, and interactions arriving in another country are recorded and stored in that country's data center only. Your deployment model has Interaction Recording Web Services nodes installed in data centers per country.

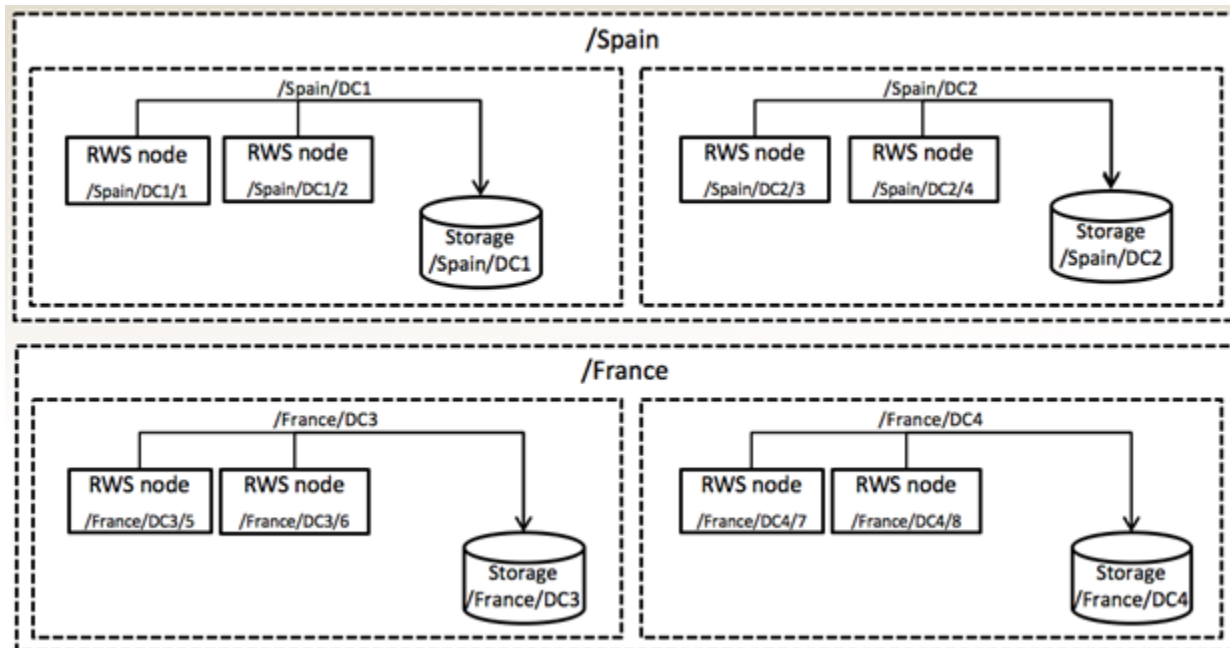
These examples require that Interaction Recording Web Services nodes be located in each data center with a local storage location. For those locations that do not have local storage configured, a centralized storage location can be used.

In the case of voice recordings, the storage location is determined based on the geo-location set in the SIP Server.

For screen recording, the mapping storage location based on the Interaction Recording Web Services nodes requires no provisioning of the location of the agent, since the connection of the Screen Recording Service to the Interaction Recording Web Services nodes represents the actual location in which the agent resides.

The following diagram is an example of how recording storage can be divided into multiple data centers. When studying this diagram consider the following:

- In a system where there is a data center in the same location as the SIP Server and GVP, recording and recording storage are located in the same data center. RWS and the Recording Crypto Server should also be deployed in all of these data centers.
- GIR recording can occur in any of these data centers. GIR operates in primary / backup across 2 such data centers. SpeechMiner and GIR components (such as Recording Processor Script or Voice Processor and Muxer), are deployed in these data centers.



Configure voice recording storage

To configure the voice recording storage locations, refer to [Configuring the Storage Credentials for Interaction Recording Web Services > Enable Storage](#).

Assign a recording IVR profile

A Recording IVR profile enables you to setup a separate voice recording storage location, per data center location, based on the SIP Server geo-location (see the content describing Geo-Location within the [SIP Server Deployment Guide](#)).

To setup a separate voice recording storage location, perform the steps in the [IVR Profile](#) section.

Configure screen recording storage

To configure the screen recording storage locations, refer to [Configuration for Screen Recordings > Configuring the Storage Credentials for Interaction Recording Web Services > Step #4 Enable storage for a single or multiple locations](#).

Recording processing

Each data center where recordings occur must be deployed with at least one Recording Processor Script or Voice Processor, to process recordings within the same data center. When the processing is complete, the recording metadata is written to the primary data center for indexing.

The Recording Processor Script or Voice Processor is deployed in the primary and backup data centers. SpeechMiner is also deployed in the primary and backup data centers, and these components are active in the primary data center while the backup data center only contains backup instances. Indexing of recording metadata also occurs in the primary data center.

For additional information, see [Deploying Recording Processor Script](#) or [Deploying Voice Processor](#).

The Recording Processor Script or Voice Processor should send recording metadata to the SpeechMiner Interaction Receiver in the primary data center according to the [IVR Profile configuration](#) above.

Screen Recording processing

The Screen Recording Service (SRS) is always connected to Interaction Recording Web Services (RWS). This connection occurs from the agent's location to a data center in which the SIP Server and RWS are deployed. This connection manages screen recording and the upload of screen recordings to the recording storage.

The Recording Muxer Script processes screen recordings through the primary data center. Muxer operates in the primary/backup data center and all active instances operate in the primary data center.

The following instructions are applicable if you are using Workspace Web Edition (WWE), or Workspace Desktop Edition (WDE) (versions earlier than 8.5.118.10), or Screen Recording Service (SRS) (versions earlier than 8.5.340.97):

- If a data center fails and agents need to use the backup data center for continuity of operations, the SRS must be reconfigured to connect to the Interaction Recording Web Services nodes located in a backup data center. Perform one of the following procedures to reconfigure the SRS for this purpose. Once the SRS is configured to connect to that remote data center, the recording files will be transported to the remote data center and saved locally at that data center.
 - Configure the agent desktop to re-submit the data center server address to the SRS, specifying the backup data center address.
 - Update DNS mapping of RWS to specify the backup data center.
 - Use an HTTP Load Balancer to redirect HTTP requests from the SRS to the backup data center where RWS is located.

The following instructions are applicable only if you are using WDE (version 8.5.118.10 or later) and SRS (version 8.5.340.97 or later):

- If a data center fails and agents need to use a backup data center for continuity of operations, SRS must be configured to connect to the Interaction Recording Web Services nodes located in a backup data center. Perform the following procedure to configure SRS for this purpose. Once SRS is configured to connect to a remote data center, SRS will automatically failover to RWS nodes in the backup data center from the primary data center and vice versa.

- Specify a value for the [interaction-workspace] **screen-recording.htcc.peer_uri** option in the WDE application object. For more information, refer to [Screen Recording Disaster Recovery](#).

Similar concepts apply if a custom agent desktop is used with integration through the SRS API as described in [Client Login API](#).

For eServices, RWS in both data centers connects to the Interaction Server in the primary data center. If a data center fails, or if the Interaction Server in a data center fails, the operator needs to make an HTTP POST request to RWS explicitly to switch over the Interaction Server in the primary data center to the backup data center for continuity of operations.

The following steps describe how to switch the Interaction Server being used by RWS between data centers:

1. Determine the genesys-environment ID using the following command with the ops username and password (ops:ops):

```
curl -u ops:ops http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services port>/api/v2/ops/genesys-environments; echo
```

Interaction Recording Web Services returns the following output:

```
{
  "statusCode": 0,
  "paths": [
    "/genesys-environments/<genesys-environment-ID>"
  ],
  "uris": [
    "http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services port>/api/v2/ops/genesys-environments/<genesys-environment-ID>"
  ]
}
```

2. Using a text editor, create a new file called **multimedia_switch** with the following content:

```
{
  "eServicesSite": "DC2" // set to site name as per application parameter siteName
}
```

3. Execute the following command:

```
curl -u ops:ops -X POST -d @multimedia_switch http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services Port>/api/v2/ops/genesys-environments/{genesys-environment-ID}/multimedia-switch --header "Content-Type: application/json"; echo
```

4. Verify that WDE is connected to same data-center:
 - a. Specify value for the option **[interaction-workspace] "disaster-recovery.eservices-site"** in the WDE application object. See: [Administrator operations in case of data center failover](#)
 - b. After the primary data center Interaction Server is brought back to working condition, make another POST to switch back to the Interaction Server in the primary data center.

Important

RWS will monitor the **eServicesSite** value every **drMonitoringDelay** seconds. These

values are configured in the **serverSettings** section in the **application.yaml** file. If there is a change in the **eServicesSite** value during monitoring, RWS will perform a switch over to the specified **eServicesSite**. For a definition of **drMonitoringDelay** refer to the [Configuration Options](#) page.

User interface

SpeechMiner works in active/backup mode across data centers.

To enable recordings from multiple data centers to be viewed from a single SpeechMiner user interface, assign a primary data center where SpeechMiner hosts the primary database and index.

Data center index update

Ensure that the indexes are up-to-date in all data centers.

SpeechMiner Index: Configure the **Daily Index Backup** and write the backup to the backup data center. Whenever the primary data center becomes unavailable, the backup index can be configured to be the primary instance in the backup data center. For additional information, see: [SMConfig > Index](#).

Elasticsearch Index: In the backup data center, write a periodic task (for example, cron job) that will execute the following re-indexing command using HTTP to RWS. It is recommended that you perform this task daily.

```
POST http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/recordings
```

If you are using screen recording you must also perform a re-indexing command for the screen recording index.

```
POST http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/screen-recordings
```

For additional information about how to obtain the contact center ID, refer to [Configuring Features](#).

For example:

1. Using a text editor, create a new file called `update_index` with the following content:

```
{
  "operationName": "forceIndex",
  "from": "1369272257713",
  "to": "1369275857713",
```

```
    "purgeOld":false
}
```

2. Execute the following command:

```
{
  curl -u ops:ops -X POST -d @update_index
  http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
  Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/recordings
  --header "Content-Type: application/json"; echo
}
```

3. If you are using screen recordings, you must also execute the following command:

```
{
  curl -u ops:ops -X POST -d @update_index
  http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services
  Port>/api/v2/ops/contact-centers/<contact center ID (in hex format)>/screen-recordings
  --header "Content-Type: application/json"; echo
}
```

Attribute	Type	Mandatory	Description
operationName	String	yes	forceIndex
from	long	yes	Milliseconds since epoch. The starting timestamp from which recordings are indexed and rounded to the nearest hour.
to	long	no	Milliseconds since epoch. The ending timestamp until recordings are indexed and rounded to the nearest hour.
purgeOld	boolean	no	The flag that specifies if the old index should be deleted prior to re-indexing. This is necessary when the updated RWS version needs an index with an updated schema. Default value = false

Due to the rounding of time units in the API, the re-indexing command interval is a minimum of one hour.

Encryption management

When multiple data centers are deployed, you must ensure that the Recording Crypto Server (RCS) **keystore** file is replicated to all the RCS instances in all the data centers.

To learn how to replicate the RCS keystore file, refer to the [Deploying Recording Crypto Server](#) page.

Media lifecycle management

Media Lifecycle Management is location specific and can be configured separately for each data center. Refer to [Configuring Media Lifecycle Management](#) to configure multiple regions.

Media Lifecycle Management

The Genesys Interaction Recording solution allows you the flexibility to manage your recording files. You can use the Recording Scheduler function of Genesys Administrator Extension to schedule purge and backup rules on the Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) node paths.

These tasks are described as follows:

- Purge—Delete call recording and screen recording metadata and media files from the WebDAV Server, Cassandra database, and SpeechMiner database.
- Backup—Back up call recording and screen recording metadata and media files from the WebDAV Server and the Cassandra database to the specified backup folder (see [Locations](#)). Note that call or screen recordings that have been backed up and then purged from the GIR system cannot be played back through SpeechMiner. These should be played with your own media player.

For more information about configuring MLM, see [Configuring Media Life Cycle Management](#).

Why would I want to use the Recording Lifecycle Scheduler?

Use the Recording Lifecycle Scheduler tool, that is installed with the [Recording Plugin](#), to create the purge and backup tasks. The following sections provide a few examples on how to set up specific tasks.

Daily Backup of Recordings

Click [here](#) for the instructions on how to use the Recording Scheduler.

Recording Lifecycle Scheduler Rules

Recording Lifecycle Scheduler Nodes > New

RWS Node Path *

Rules

Quick Filter

	Name	Enabled	Schedule	Tasks
<input type="checkbox"/>	Daily Backup	<input checked="" type="checkbox"/>	Every day at 22:00:00	backup, range: 1 to 2 days old, voice, Parameters: location=file://BackUpStorageFolderforVoiceRecordings, executeOnScreenRecording=true,

Click to enlarge diagram

For example, you want to back up yesterday's voice and their associated screen recordings at the end of each day, so you can keep copies in case the primary storage folder is unavailable when you need to listen to a recording.

As shown in this Rule example, create the following Backup task:

1. **FilterType** = Voice
 - **Min Age** = 1
 - **Max Age** = 2
 - **Location** = file://<BackUpStorageFolderforVoiceRecordings>
 - Select: **Include Screen Recordings**
2. Select a time that will not impact your daily business activities. Remember to take into account that the **Repeat Daily At:** parameter is in UTC time.

Purging Old Files

Click [here](#) for the instructions on how to use the Recording Scheduler.

The screenshot shows the 'Recording Lifecycle Scheduler Rules' interface. At the top, there's a breadcrumb 'Recording Lifecycle Scheduler Nodes > New' and a 'Save' button. Below this is a form for 'RWS Node Path' with the value '/GSWNodePath'. A 'Rules' section contains a table with one rule:

<input type="checkbox"/>	Name	Enabled	Schedule	Tasks
<input type="checkbox"/>	PurgeOldFiles	✓	Every day at 02:00:00	purge, range: 365 days or older, voice. Parameters: executeOnScreenRecording=true,

There is a 'Quick Filter' search bar and icons for delete, add, and refresh in the top right of the rules table.

Click to enlarge diagram

For example, you want to delete any voice recording that is more than a year old. As shown in this Rule example, create the following Purge task:

1. **FilterType** = Voice
 - **Min Age** = 365
 - **Max Age** = Leave empty
 - Select: **Include Screen Recordings**
2. Select a time that will not impact your daily business activities. Remember to take into account that the **Repeat Daily At:** parameter is in UTC time.

Backing Up and Purging Files

Click [here](#) for the instructions on how to use the Recording Scheduler.

Recording Lifecycle Scheduler Rules
Save

Recording Lifecycle Scheduler Nodes > New

RWS Node Path *

Rules
+
↺

<input type="checkbox"/>	Name	Enabled	Schedule	Tasks
<input type="checkbox"/>	Backup and Purge	✓	Every day at 02:00:00	backup, range: 365 days or older, voice, Parameters: location=\\file://BackUpStorageFolderforVoiceRecordings; executeOnScreenRecording=true, purge, range: 365 days or older, voice,

Click to enlarge diagram

For example, you want to backup and then delete any recording that is more than a year old. As shown in this Rule example, create the following two tasks in this order:

- Type** = Backup
 - FilterType** = Voice
 - Min Age** = 365
 - Max Age** = Leave empty
 - Location** = file://<BackUpStorageFolderforVoiceRecordings>
- Type** = Purge
 - FilterType** = Voice
 - Min Age** = 365
 - Max Age** = Leave empty

Important

The **FilterType**, **MinAge**, **MaxAge** (or any filter) parameters must always be the same between these two tasks.

Warning

If you select **Next Task**, the purge will execute even if the backup was unsuccessful. You might lose your recordings.

Select a time that will not impact your daily business activities. Remember to take into account that the **Repeat Daily At:** parameter is in UTC time.

Warning

The Backup task must come before the Purge task, if you want to backup the files.

Important

If an Interaction Recording Web Services (Web Services) node fails or is not running then the rule and tasks are assigned to it, the rule will not be executed.

Locations

The term *location* is used to describe the specific place to store recording files. There are two types of locations used in Genesys Interaction Recording:

- **Node (location) and node path (location-based hierarchy):** A node represents a specific Interaction Recording Web Services (Web Services) instance. For example, if you have three Interaction Recording Web Services (Web Services) instances installed, you have three nodes. A node can be identified using a node path. The node path is specified in the Interaction Recording Web Services (Web Services) `application.yaml` file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the `server-settings.yaml` file instead). This node path must be unique for each Interaction Recording Web Services (Web Services) instance. Sometimes the node is called a 'location'. Every rule (set of tasks) in Media Lifecycle Management is assigned to a specific node. So, if you want to run a rule with a purge/backup task, you assign it to a specific node path. At the designated time, this node (and only this one) runs the purge task. These node paths represent locations or regions; therefore, node path settings are sometimes called *location-based* settings.
- **Backup folder for Media Lifecycle Management:** When Interaction Recording Web Services (Web Services) performs a backup task, the resulting backup is stored in the specified path. The folder where the files are stored is sometimes called the *backup location/folder* or *archive location/folder*.

For more information and examples that describe how to use and configure a location, see [Interaction Recording Web Services Settings Groups](#).

Scalability and High Availability

This topic discusses options/possibilities for scalability and high availability for the Genesys Interaction Recording solution.

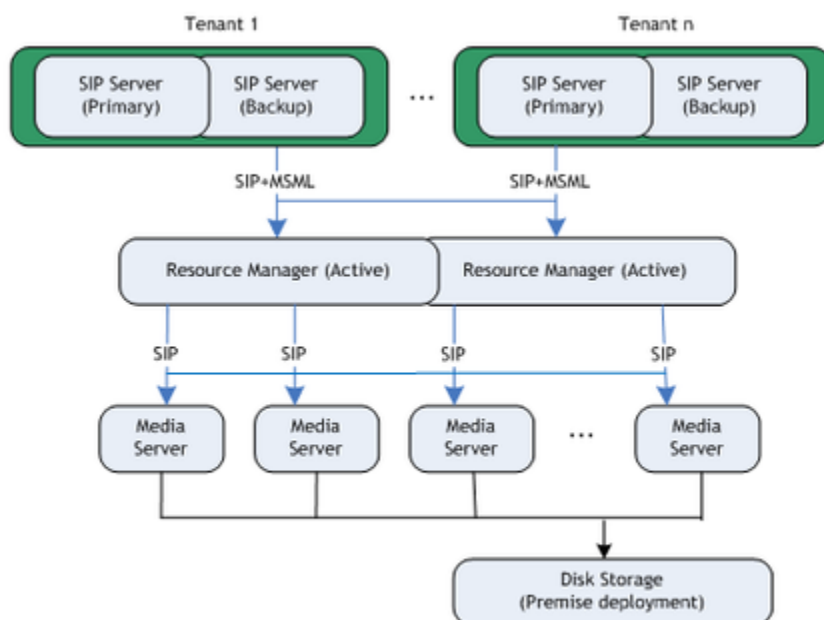
Tip

Click on the diagrams to enlarge them.

<tabber> Scalability=

Scalability

SIP Server uses the same unified media server instances to provide any new media services, and in this case, the recording service.



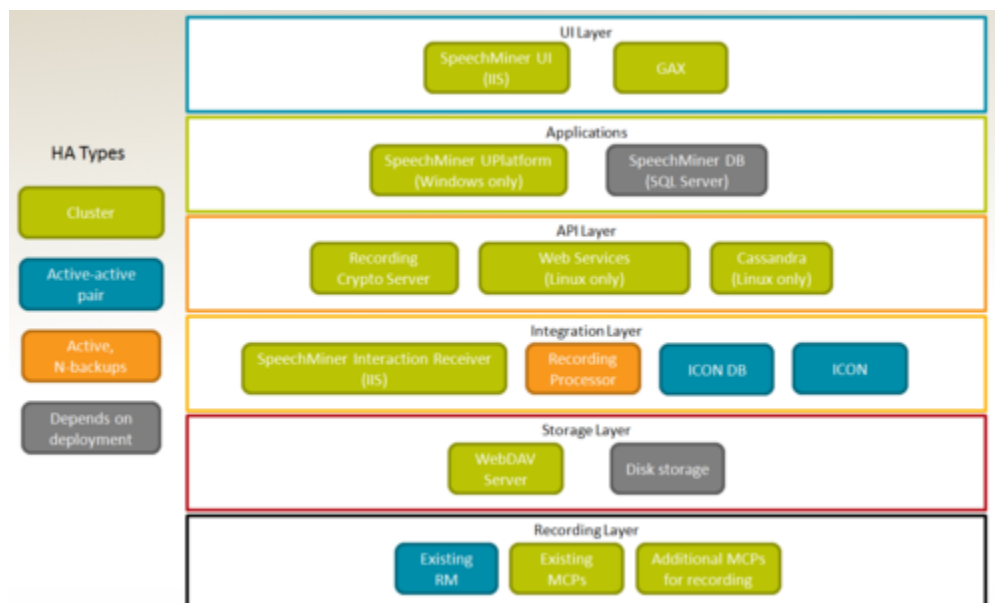
A media server (MCP) cluster is managed by a pair of Resource Managers running in Active-Active mode. Resource Manager identifies the tenant coming from SIP Server and selects the appropriate IVR Profile for the call. Each MCP instance in the cluster is able to handle call recording for any tenant based on the IVR Profile information.

MCP writes call recording to Amazon S3 or disk storage depending on the deployment type. Amazon S3 being a cloud service is already designed to scale. For disk storage, the premise deployment use a disk storage mechanism such as a disk array that can be scaled for the cluster of media servers. For more information on public keys and encryption, see the [Genesys Media Server Deployment Guide](#).

[-] High Availability=

High Availability

This section describes supported High Availability and failover modes for the Genesys Interaction Recording components. The following diagram illustrates HA architecture:



SIP Server

SIP Server can be deployed in an active/hot-standby pair; whenever the primary instance fails, the hot-standby instance will take over and will have knowledge of all established sessions. If a recording session has been established with a recording server, a failure of SIP Server will not affect the operation of the communication and recording sessions.

Resource Manager

Resource Manager is a SIP Proxy that can operate in either active/hot-standby or active/active pair. When there is a failure at the Resource Manager, the remaining instance of Resource Manager will become active and continue to accept incoming requests. The remaining instance will remember the affinity of the recording sessions with Media Server as well as the recording server.

Media Control Platform

When the media is bridged through the Media Control Platform (MCP), MCP becomes a single point of failure for the duration of the communication session. If MCP fails, Resource Manager notifies SIP Server about the failure so that SIP Server can take alternative action on the call. SIP Server first joins the endpoints together to ensure communication session is not interrupted. SIP Server then attempts to record the call again by initiating a new recording session with the same parameters. RM most likely will select another MCP instance while the failed MCP is unavailable.

Web Services and Cassandra

Web Services and Cassandra runs in a N+1 cluster. For more information, see [Initializing the Cassandra database](#).

Recording Processor

The Recording Processor runs in both active-active, and active-backup modes.

Recording Crypto Server

The Recording Crypto Server runs in active-active pair mode.

SpeechMiner

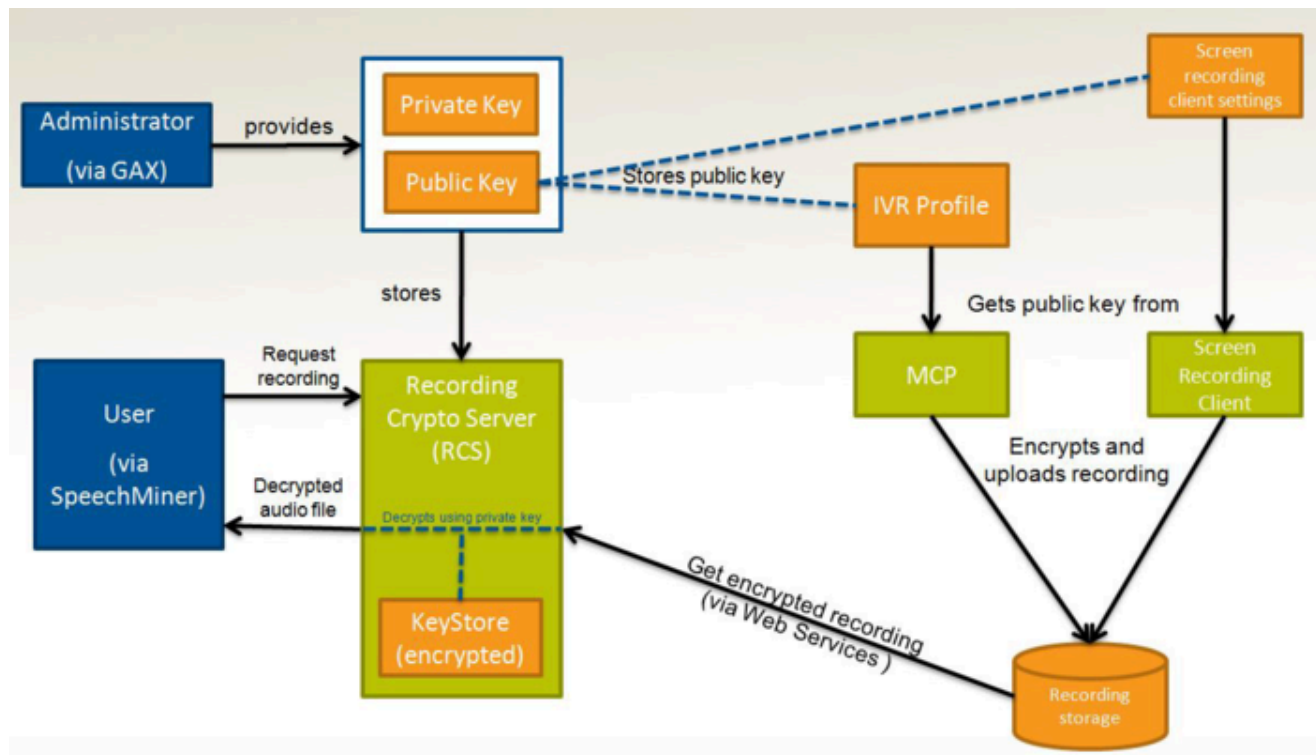
SpeechMiner requires reliable shared storage that can be accessed (read/write) by all SpeechMiner components. The shared folders are mapped as an UNC path on the SpeechMiner machines and all SpeechMiner machines must refer to the same UNC path.

- SpeechMiner UI—You can deploy N+1 active instances.
- SpeechMiner Interaction Receiver—You can deploy N+1 active instances.
- SpeechMiner UPlatform—You can deploy active/standby instances of the Scheduler task on UPlatform. When the active instance fails, the standby instance automatically takes over to become the active instance.

Security and Encryption

A key management system allows users to decrypt call and screen recordings for playback purposes.

The following diagram provides a quick overview on how the encryption keys are managed:



Security Keys and Storage

The administrator for the customer tenant provides two keys:

1. Private key—For decrypting audio files.
2. Public key—For encrypting audio files.

Private Key Storage

The Recording Crypto Server (RCS) uses Java Cryptography Architecture (JCA) to store and access private keys in the Java KeyStore.

Key Size Limit

In older versions of Java 8, the default installation limits key sizes to 128 bits. Larger key sizes can be enabled by installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. For information about installing JCE, see [Deploying Recording Crypto Server](#). Java 8u161 and later versions support the unlimited policy by default.

Public Key Storage

Public keys for voice recordings are stored in the IVR profile for the tenant. The Media Control Platform (MCP) reads the certificate from this IVR Profile. A customer may include multiple private/public keys for a tenant; however, private keys per user are not necessary.

Since MCP is shared across multiple tenants, there must be a single place to store all certificates that are accessible. Configuration Server stores all certificates (text format) as parameters in the IVR Profile for call recording. MCP loads all call recording IVR Profiles and receive updates for the IVR Profiles. When Resource Manager forwards a recording request to MCP, it inserts the IVR Profile DBID so MCP can look up the list of certificates to perform encryption.

The certificates are stored in the IVR Profile Annex tab in the recording-certificates section. For information about configuring encryption, see [Encryption](#)

Public keys for screen recordings are stored in the screen-recording-encryption settings of Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). For more information about storing these keys, see [Deploying Interaction Recording Web Services](#) or [Deploying Workspace Web Edition and Web Services for GIR](#).

Encrypting Recordings

MCP encrypts call recordings in two parts:

1. The audio file itself is encrypted with a session key using AES encryption.
2. MCP posts the encrypted audio data to WebDAV storage and sends metadata to the Recording Processor. The session key is encrypted (using the GIR public key) and stored in a PKCS7 envelope. This envelope is included in the media file's call recording metadata.

Important

It is your responsibility to store your private keys and certificates, including the expired ones.

You should also backup your keystore, keystore password, certificates and private keys in a secure location offsite to protect against site level disasters. When Genesys Interaction Recording encryption is enabled, loss of the keystore and private key would result in loss of recording files.

For more information, see [Encrypting and Provisioning Certificates](#).

File Format

The Genesys Interaction Recording solution supports RSA certificates and keys for the recording encryption. Certificates and keys are uploaded from files in PEM format. Certificates files must be X.509 PEM format. This is the default format generated by OpenSSL.

The following formats are supported for private key files:

- OpenSSL RSA private key format. The PEM file must start with -----BEGIN RSA PRIVATE KEY-----.
- PKCS#8 private key format. The PEM must start with -----BEGIN PRIVATE KEY-----.
- PKCS#8 encrypted private key format. The PEM file must start with -----BEGIN ENCRYPTED PRIVATE KEY-----.

For an example of encryption using OpenSSL, see [Sample Encryption Using OpenSSL](#).

Certificate Validation

Certificate validation is performed when certificates are uploaded to the system and upon each use of a certificate to encrypt a recording. The validation consists of checking the certificate signatures from the recording certificate up to the root CA certificate, and checking that the certificate "not valid before" date has passed and "current" date is within the "not valid before" and "not valid after" dates.

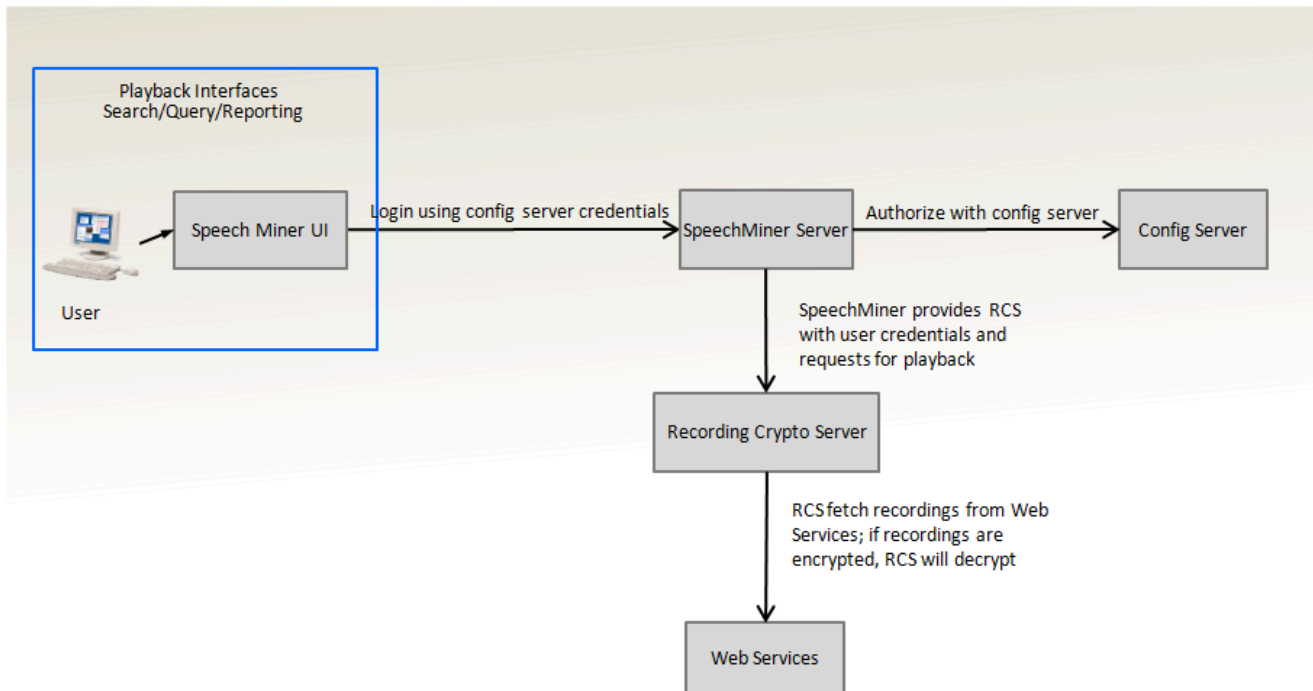
A certificate that is not yet valid can be imported, but an error will occur if it used for a recording.

For configuration of the Root CA for certificate upload, see the The certificates are stored in the IVR Profile Annex tab in the recording-certificates section. For information about configuring encryption, see [Encryption](#).

For configuration of the Root CA for the certificates used for each recording, see [Configuring GVP](#).

Decrypting Recordings

If the user wants to play back a specific recorded call, the SpeechMiner Server checks for the user's permission and makes a request to the Recording Crypto Server to fetch and decrypt the recording. The Recording Crypto Server retrieves the call metadata and fetches the recording from Interaction Recording Web Services (Web Services). The recording files are decrypted using the private key, and the audio is sent back to the SpeechMinder Server to be played in a browser.



Playback of Archived Files

The archiving function provides a zip file containing multiple encrypted recording files. Each encrypted file is in the PKCS#7 format.

Each encrypted file can be played back with the following OpenSSL command:

```
openssl smime -decrypt -inform DER -in <encrypted_file>.bin -inkey <private_key_file> -out <output_file>
```

where:

<encrypted_file> is the file to be decrypted

<private_key_file> is the private key that you used to initially configure file encryption

<output_file> is the file to be written after decryption has taken place

Archiving and Metadata

The Genesys Interaction Recording solution uses the [Recording Crypto Server](#) and [Media Lifecycle Management](#) system to back up store recording files.

Recording Crypto Server Archive Structure

The Recording Crypto Server, by default, is scheduled to generate an archive file every day based on the retention period. Each time the archival process is run, the output file is saved to a compressed file. The compressed file contains the recording files and the metadata. The following directory structure is the result of the expanded compressed file:

```
{contact-center-id}
  {tenant-name}
    {year-month-day}
      {recording-id}
        recording.json
        {mediaFiles}
```

- **contact-center-id**—The unique identifier of the contact center that is generated by Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier).
- **tenant-name**—The name of the tenant.
- **year-month-day**—The directory where the recording files are stored. There is a separate directory generated for each calendar day.
- **recording-id**—The unique recording identifier for each recording.
- **recording.json**—The recording metadata.
- **mediaFiles**—One or more recording files for the recording. The file name is referenced in the metadata. If encryption is enabled for this tenant, the recording is also encrypted in PKCS7 format. For an example of how to decrypt an encrypted recording file, see [Decrypting Call Recordings](#).

Media Lifecycle Management Archive Structure

This section explains the archive structure that results when you use the [Recording Lifecycle Scheduler](#) component of Genesys Administrator Extension to create zipped and unzipped backup tasks.

Important

Multiple rules that create backups should not be created with all of the following information configured identically:

- minAge and maxAge filter specs
- SS MM HH schedule time
- Filter type (for example, Voice)
- Location (for example, file://myArchives or blank)

Because these rules will all attempt to write to an archive file with the same file name. You can provide schedule times or location diversity to avoid this limitation.

Each zipped backup task exports the recording files to one compressed file, with the full path: `<archive export directory>/<Tenant Name>-<Filter Type>-<Max Age Filter>-<Min Age Filter>-<yyyyMMddHHmmss>.zip`

Each unzipped backup task exports the recording files to one folder, with the full path: `<archive export directory>/<Tenant Name>-<Filter Type>-<Max Age Filter>-<Min Age Filter>-<yyyyMMddHHmmss>`

Important

If the `useFullPathInMediaFileBackup` option in the **application.yaml** file is set to true on the node performing the backup task, unzipped backup tasks cannot point to a remotely mounted Windows directory. By default, the `useFullPathInMediaFileBackup` option is set to false.

[+] Show descriptions of the parameters.

- `<archive export directory>` is determined first by the task's parameters ["location"] setting. If missing or blank, it defaults to the server setting, `backgroundScheduledMediaOperationsSettings.backupExportURI`.
- `<Tenant Name>` is always "Environment".
- `<Filter Type>` is based on the task filter that exported this archive: "voice" or "screen".
- `<Min Age Filter>` is based on the task filter that exported this archive: 0 if was not set.
- `<Max Age Filter>` is based on the task filter that exported this archive: 0 if was not set.
- `<yyyyMMddHHmmss>` is the UTC time when the task started its execution.

`useFullPathInMediaFileBackup: false`

If the `useFullPathInMediaFileBackup` option in the **application.yaml** file is set to false on the node

performing the backup task (the default), the structure of the compressed file and the directory structure of the folder when using an unzipped backup is as follows:

```
<contact center id>
+---<Tenant Name>
+---<yyyy>
+---<MM>
+---<dd>
+---<recording id>
+---meta-data.json
+---<MediaFiles>
+---screen
+---<screen recording id>
+---meta-data.json
+---<MediaFiles>
```

[+] Show descriptions of the parameters.

- <Contact center id> is the contact center id for which the archiving is performed
- <Tenant Name> is always "Environment"
- <yyyy> is the year component in the startTime parameter of the recording id
- <MM> is the month component in the startTime of the recording id
- <dd> is the day component in the startTime of the recording id
- <recording id> is the recording id of the recording being archived
- meta-data.json is a file containing the recording metadata of the id in JSON format
- <MediaFiles> are one or more recording files for the recording. The file name is referenced in the metadata. If encryption is enabled for this tenant, the recording is also encrypted in PKCS7 format. For an example of how to decrypt an encrypted recording file, see [Decrypting Call Recordings](#).
- screen is a sub-folder under the <recording id> where the recording id is that of the call recording. When the task filter is voice and executeOnScreenRecording is true, both call recordings and screen recordings may be archived. The screen recordings that are archived because of the association with the <call recording id>, goes under the screen sub-folder.

The following entries can have multiple instances under the same parent entry:

- <yyyy>
- <MM>
- <dd>
- <recording id>
- <MediaFiles>

useFullPathInMediaFileBackup: true

If the useFullPathInMediaFileBackup option in the **application.yaml** file is set to true on the node performing the backup task, the structure of the compressed file and the directory structure of the folder when using an unzipped backup is as follows:

```

<contact center id>
+---<Tenant Name>
+---<yyyy>
+---<MM>
+---<dd>
+---<recording id>
+---meta-data.json
+---<path from the originalMediaDescriptor>
+---screen
+---<screen recording id>
+---meta-data.json
+---<path from the originalMediaDescriptor>

```

[+] Show descriptions of the parameters.

- <Contact center id> is the contact center id for which the archiving is performed
- <Tenant Name> is always "Environment"
- <yyyy> is the year component in the startTime parameter of the recording id
- <MM> is the month component in the startTime of the recording id
- <dd> is the day component in the startTime of the recording id
- <recording id> is the recording id of the recording being archived
- meta-data.json is a file containing the recording metadata of the id in JSON format
- <path from the originalMediaDescriptor> is the archived media file. The filename is the value of the "mediaFiles[n]["originalMediaDescriptor"]["path"] JSON attribute:

```

"originalMediaDescriptor": {
  "storage": "WebDAV",
  "path":
"/TPDLFSN0J13EJF50V3EB2182R000000V_1001_1001_1003_1001_2014-11-05_18-11-44__AgentVoice1_2014_11_05_10_12_03
  "data": {
    "storagePath": "http://eservices:8085/recordings"
  }
},

```

For the above example, the filename will be "TPDLFSN0J13EJF50V3EB2182R000000V_1001_1001_1003_1001_2014-11-05_18-11-44__AgentVoice1_2014_11_05_10_12_03.mp4_aH3XSa0_S_q9RVun6EnW0A.mp4". Note that the "/" (slash character) at the start of the path is removed from the name.

```

"originalMediaDescriptor":{
  "path":"http://esx36-ip21-167.gws.genesys.com/recordings/func-test-gir-sched-6331c0e2-1303-44ff-9b9e-ae5964155e56.mp3",
  "storage":"WebDAV"
}

```

For the above example, the filename retains the path value—for example, "http://...", but will be converted to "http_/esx36-ip21-167.gws.genesys.com/recordings/func-test-gir-sched-6331c0e2-1303-44ff-9b9e-ae5964155e56.mp3" by an extraction tool that operates on Windows. This only applies to a zipped backup.

- screen is a sub-folder under the <recording id> where the recording id is that of the call recording. When the task filter is voice and executeOnScreenRecording is true, both call recordings and screen recordings may be archived. The screen recordings that are archived because of the association with the <call recording id>, goes under the screen sub-folder.

The following entries can have multiple instances under the same parent entry:

- <yyyy>
- <MM>
- <dd>
- <recording id>
- <path from the mediaDescriptor>

Important

Genesys recommends that you do not use WinRAR to extract from a zipped archive because it strictly adheres to the Windows 260 character path limit, even if the extract to path is preceded by \\?. Use 7-Zip instead.

For more information about Media Lifecycle Management, click [here](#).

Call Recording Metadata

Depending on which component you are using between Voice Processor and Recording Processor Script, recording metadata is built differently. The Voice Processor builds the recording metadata from GIM, and from the metadata from MCP for each recording file. The Recording Processor builds the recording metadata from ICON, and from the metadata from MCP for each recording file. The overall structure is in JSON format with the following mandatory core properties:

- id—The recording identifier.
- callerPhoneNumber—The caller DN (ANI).
- dialedPhoneNumber—The dialed number (DNIS).
- startTime—The start time of the voice call
- stopTime—The end time of the voice call.
- screenRecording—Indicates whether or not the call recording has one or more associated screen recordings.
- region—The physical location of the recording.
- mediaFile—A list of recording files associated with this recording.
- eventHistory—A list of call events associated with this recording.

The overall structure can also have the following optional core properties:

- callType—Identifies the call as either Unknown, Inbound, Outbound, Internal, or Consult.

mediaFile

mediaFile Properties

The following table describes the mediaFile properties.

Property	Data Type	Description	Required
mediaUri	string	Specifies the Interaction Recording Web Services (Web Services) URI for the media file.	Yes
mediaPath	string	Specifies the Interaction Recording Web Services (Web Services) relative path for the media file.	Yes
startTime	datetime	Specifies the start time of the media file.	Yes
stopTime	datetime	Specifies the stop time of the media file. If MCP fails, this value will be the same as the startTime.	Yes
mediaID	string	Specifies the media file name for the media file that is used by clients to refer to the same media file. MCP ensures that this value is globally unique.	Yes
type	string	Specifies the MIME type of the media file.	Yes
duration	string	Specifies the time duration of the media file.	No
size	string	Specifies the size, in bytes, of the media file.	Yes
tenant	string	Specifies the tenant that the recording belongs to.	Yes
ivrprofile	string	Specifies the IVR Profile name that serviced the recording.	Yes
parameters	object—The properties are parameters.	Specifies the list of additional metadata information provided by SIP Server and the client applications. The properties are:	Yes

Property	Data Type	Description	Required
		<ul style="list-style-type: none"> • sipsAppName • ani • dnis • dateTime • calluuid • connid • agentId • id • recordDn • record 	
masks	array of objects—Each object contains the time and type property.	Specifies the time stamps of the pause/resume periods if the recording is masked by a client application.	No
pkcs7	string	Specifies the PKCS7 envelope (in PEM, base 64 string format) if the media file is encrypted.	No
certAlias	array of strings	Specifies a list of aliases to the encryption certificates if the media file is encrypted.	No (Yes if the pkcs7 property is present)
partitions	array of strings	Specifies a list of partition names for the media file.	Yes
accessgroups	array of strings	Specifies the access groups identified agent associated with the recording.	Yes
channels	number	Specifies whether the recording audio is capture in mono (1) or stereo (2).	Yes

eventHistory

eventHistory Properties

The following table describes the eventHistory properties.

Property	Data Type	Description	Required
occurredAt	datetime	Specifies the start time of the event.	Yes
calluuid	string	Specifies the call UUID that the event belongs to.	Yes
event	string	Specifies the event type: <ul style="list-style-type: none"> • Joined • Left • data 	Yes
contact	object	Specifies the the contact information of the caller who joined or left the recording if the event is Joined or Left.	No
data	object	The attached data included in the recording if the event is data.	No

Example Metadata

Metadata Format

The following code snippet illustrates the metadata format:

```
{
  "mediaFiles": [
    {
      "duration": "32662",
      "ivrprofile": "DefaultIVRProfile",
      "mediaDescriptor": {
        "data": {
          "bucket": "bucket2"
        },
        "path": "prefixA/029UDMMJ90A0L8151CGQH5AES000003_2015-06-24_20-19-03-00710158-100043FF-00000001.mp3",
        "storage": "storagelocation"
      },
      "mediaId": "029UDMMJ90A0L8151CGQH5AES000003_2015-06-24_20-19-03-00710158-100043FF-00000001.mp3",
    }
  ]
}
```

```

    "parameters": {
      "agentId": "1001",
      "ani": "1000",
      "callUuid": "029UDMMJ90A0L8151CGQHG5AES000003",
      "connId": "007002684ae93003",
      "dateTime": "2015-06-24T20:19:03Z",
      "dnis": "1002",
      "id": "029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-19-03",
      "partitions": "salesasdfsaf",
      "recordDN": "1001",
      "rp.speechminer_auth": "user:password",
      "rp.speechminer_uri": "http://sm/interactionreceiver",
      "sipsAppName": "SIPS_vagrant.genesys.com"
    },
    "size": "129888",
    "startTime": "2015-06-24T20:19:03Z",
    "stopTime": "2015-06-24T20:19:36Z",
    "tenant": "Environment",
    "type": "audio/mp3"
  }
}
]
}

```

[+] See another example.

```

{
  "statusCode": 0,
  "recording": {
    "id": "029UDMMJ90A0L8151CGQHG5AES000003",
    "callerPhoneNumber": "1000",
    "dialedPhoneNumber": "1002",
    "startTime": "2015-06-24T20:18:31.000+0000",
    "stopTime": "2015-06-24T20:19:36.000+0000",
    "mediaFiles": [
      {
        "mediaUri": "http://vagrant.genesys.com:8081/api/v2/recordings/029UDMMJ90A0L8151CGQHG5AES000003/play/ed362c5b-4025-41ef-ac25-a7541a8d4f09.mp3",
        "mediaPath": "/recordings/029UDMMJ90A0L8151CGQHG5AES000003/play/ed362c5b-4025-41ef-ac25-a7541a8d4f09.mp3",
        "startTime": "2015-06-24T20:18:31.000+0000",
        "stopTime": "2015-06-24T20:18:42.000+0000",
        "callUUID": "029UDMMJ90A0L8151CGQHG5AES000003",
        "mediaId": "029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-18-30-00710158-100043FD-00000001.mp3",
        "type": "audio/mp3",
        "duration": "11300",
        "tenant": "Environment",
        "ivrprofile": "DefaultIVRProfile",
        "size": "44928",
        "parameters": {
          "dnis": "1002",
          "connId": "007002684ae93003",
          "ani": "1000",
          "dateTime": "2015-06-24T20:18:30Z",
          "recordDN": "1000",
          "agentId": "1000",
          "rp.speechminer_uri": "http://sm/interactionreceiver",
          "rp.speechminer_auth": "user:password",
          "sipsAppName": "SIPS_vagrant.genesys.com",
          "id": "029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-18-30",
          "username": "agent1000",
          "partitions": "sales",
          "callUuid": "029UDMMJ90A0L8151CGQHG5AES000003"
        }
      }
    ]
  }
}

```

```

    },
    "partitions": ["sales"],
    "accessgroups": [
        "/team1"
    ]
},
{
    "mediaUri": "http://vagrant.genesys.com:8081/api/v2/recordings/
029UDMMJ90A0L8151CGQHG5AES000003/play/47693aa9-b098-46fb-b8cf-bbc494f59d68.mp3",
    "mediaPath": "/recordings/029UDMMJ90A0L8151CGQHG5AES000003/play/
47693aa9-b098-46fb-b8cf-bbc494f59d68.mp3",
    "startTime": "2015-06-24T20:19:03.000+0000",
    "stopTime": "2015-06-24T20:19:36.000+0000",
    "callUUID": "029UDMMJ90A0L8151CGQHG5AES000003",
    "mediaId":
"029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-19-03-00710158-100043FF-00000001.mp3",
    "type": "audio/mp3",
    "duration": "32662",
    "tenant": "Environment",
    "ivrprofile": "DefaultIVRProfile",
    "size": "129888",
    "parameters": {
        "dnis": "1002",
        "connId": "007002684ae93003",
        "ani": "1000",
        "dateTime": "2015-06-24T20:19:03Z",
        "recordDN": "1001",
        "agentId": "1001",
        "rp.speechminer_uri": "http://sm/interactionreceiver",
        "rp.speechminer_auth": "user:password",
        "sipsAppName": "SIPS_vagrant.genesys.com",
        "id": "029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-19-03",
        "username": "agent1001",
        "partitions": "salesasdfsaf",
        "callUuid": "029UDMMJ90A0L8151CGQHG5AES000003"
    },
    "partitions": ["salesasdfsaf"],
    "accessgroups": [
        "/team1"
    ]
}
],
"eventHistory": [
{
    "occurredAt": "2015-06-24T20:18:19.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "eventId": "2015-06-24 20:18:19.792_029UDMMJ90A0L8151CGQHG5AES000003",
    "event": "Data",
    "data": {
        "added": {
            "From:tag": "003C43D0-0F8B-158B-B223-0CFAA8C0AA77-7"
        }
    }
},
{
    "occurredAt": "2015-06-24T20:18:32.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "eventId": "2015-06-24 20:18:32.009_029UDMMJ90A0L8151CGQHG5AES000003",
    "event": "Data",
    "data": {
        "added": {
            "GSIP_RECORD": "ON",
            "GSIP_REC_FN": "029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-18-30"
        }
    }
}
]

```

```

    }
  },
  {
    "occurredAt": "2015-06-24T20:18:42.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "eventId": "2015-06-24 20:18:42.259_029UDMMJ90A0L8151CGQHG5AES000003",
    "event": "Data",
    "data": {
      "updated": {
        "From:tag": "003C43D0-0F8B-158B-B223-0CFAA8C0AA77-9"
      },
      "deleted": {
        "GSIP_RECORD": "ON"
      }
    }
  },
  {
    "occurredAt": "2015-06-24T20:19:05.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "eventId": "2015-06-24 20:19:05.284_029UDMMJ90A0L8151CGQHG5AES000003",
    "event": "Data",
    "data": {
      "added": {
        "GSIP_RECORD": "ON"
      },
      "updated": {
        "GSIP_REC_FN": "029UDMMJ90A0L8151CGQHG5AES000003_2015-06-24_20-19-03"
      }
    }
  },
  {
    "occurredAt": "2015-06-24T20:18:04.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "contact": {
      "type": "User",
      "phoneNumber": "1000",
      "userName": "agent1000",
      "firstName": "Firstname1000",
      "lastName": "lastname"
    },
    "event": "Joined"
  },
  {
    "occurredAt": "2015-06-24T20:18:44.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "contact": {
      "type": "User",
      "phoneNumber": "1001",
      "userName": "agent1001",
      "firstName": "Firstname1001",
      "lastName": "lastname"
    },
    "event": "Joined"
  },
  {
    "occurredAt": "2015-06-24T20:18:20.000+0000",
    "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
    "contact": {
      "type": "User",
      "phoneNumber": "1002",
      "userName": "agent1002",
      "firstName": "Firstname1002",

```

```

        "lastName": "lastname"
      },
      "event": "Joined"
    },
    {
      "occurredAt": "2015-06-24T20:18:42.000+0000",
      "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
      "contact": {
        "type": "User",
        "phoneNumber": "1000",
        "userName": "agent1000",
        "firstName": "Firstname1000",
        "lastName": "lastname"
      },
      "event": "Left"
    },
    {
      "occurredAt": "2015-06-24T20:19:36.000+0000",
      "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
      "contact": {
        "type": "User",
        "phoneNumber": "1001",
        "userName": "agent1001",
        "firstName": "Firstname1001",
        "lastName": "lastname"
      },
      "event": "Left"
    },
    {
      "occurredAt": "2015-06-24T20:19:36.000+0000",
      "calluuid": "029UDMMJ90A0L8151CGQHG5AES000003",
      "contact": {
        "type": "User",
        "phoneNumber": "1002",
        "userName": "agent1002",
        "firstName": "Firstname1002",
        "lastName": "lastname"
      },
      "event": "Left"
    }
  ],
  "screenRecording": true,
  "region": "us"
}

```

Screen Recording Metadata

The overall structure of the screen recording metadata is in JSON format with the following core properties:

- **id**—The recording identifier and the start time of the recording.
- **callRecordingId**—The associated call recording identifier, if the screen recording is associated with a call recording.
- **startTime**—The start time of the screen recording.
- **stopTime**—The end time of the screen recording.

- **mediaFile**—A list of recording files associated with this recording.
- **eventHistory**—A list of call events associated with this recording.
- **nonDelete**—Indicates whether or not the screen recording is protected from deletion.
- **region**—The physical location of the recording.

mediaFile

mediaFile Properties

The following table describes the mediaFile properties.

Property	Data Type	Description	Required
mediaUri	string	Specifies the Interaction Recording Web Services (Web Services) URI for the media file.	Yes
mediaPath	string	Specifies the Interaction Recording Web Services (Web Services) relative path for the media file.	Yes
playPath	string	Specifies the relative path to play back a recording file.	Yes
startTime	datetime	Specifies the start time of the media file.	Yes
stopTime	datetime	Specifies the stop time of the media file.	Yes
mediaId	string	Specifies the media file name for the media file that is used by clients to refer to the same media file.	Yes
type	string	Specifies the MIME type of the media file.	Yes
duration	string	Specifies the time duration of the media file.	No
size	string	Specifies the size, in bytes, of the media file.	Yes
parameters	object—The properties are parameters.	Specifies the list of additional metadata information provided by SIP Server and the client applications. The properties are:	Yes

Property	Data Type	Description	Required
		<ul style="list-style-type: none"> agentID callUuid contact monitors muxed_mediaIds region virtualHeight virtualWidth originalVirtualHeight originalVirtualWidth 	

Example Metadata

Metadata Format

The following code snippet illustrates the metadata format:

```
{
  "id": "02A1MCKSD0CDNBRP1CGQHG5AES000003_2017-03-30_17-00-18",
  "callRecordingId": "02A1MCKSD0CDNBRP1CGQHG5AES000003"
  "startTime": "2017-03-30T17:00:18.000+0000",
  "stopTime": "2017-03-30T17:02:09.000+0000",
  "mediaFiles": [
    {
      "mediaUri": "http://vagrant.genesys.com:8081/api/v2/recordings/029UDMMJ90A0L8151CGQHG5AES000003/play/47693aa9-b098-46fb-b8cf-bbc494f59d68.mp4",
      "mediaPath": "/screen-recordings/02A1MCKSD0CDNBRP1CGQHG5AES000003_2017-03-30_17-00-18/content/47693aa9-b098-46fb-b8cf-bbc494f59d68.mp4",
      "playPath": "/screen-recordings/02A1MCKSD0CDNBRP1CGQHG5AES000003_2017-03-30_17-00-18/content/6a3082c4-6ff9-4133-84de-b6333d36505d.mp4",
      "startTime": "2017-03-30T17:00:18.000+0000",
      "stopTime": "2017-03-30T17:02:09.000+0000",
      "mediaId": "Jz0Vt5zQS5mcnS8ibDzFsA_02A1MCKSD0CDNBRP1CGQHG5AES000003_5d1c6dc1d96e4a178d672ddd019ac753_2017_03_30_17_00_19",
      "type": "video/mp4",
      "duration": "0:01:51",
      "size": "2594464",
      "parameters": {
        "muxed_mediaIds": [
          "02A1MCKSD0CDNBRP1CGQHG5AES000003_5d1c6dc1d96e4a178d672ddd019ac753_2017_03_30_17_00_19",
          "02A1MCKSD0CDNBRP1CGQHG5AES000003_2017-03-30_17-00-18-0071018D-1014A2D5-00000001.mp3"
        ]
      },
      "agentID": "1000",
      "virtualHeight": "749",

```

```
    "contact": {
      "userName": "agent1000",
      "lastName": "lastname",
      "firstName": "Firstname1000"
    },
    "virtualWidth": "1920",
    "region": "us",
    "originalVirtualHeight": "1920",
    "originalVirtualWidth": "4920",
    "callUuid": "02A1MCKSD0CDNBRP1CGQHG5AES000003",
    "monitors": [
      {
        "name": "monitor_0",
        "primary": true,
        "originalPositions": "[0, 0, 1024, 768]",
        "actualPositions": "[0, 0, 399, 299]"
      }
    ]
  },
  "eventHistory": [],
  "nonDelete": true,
  "region": "us"
}
```

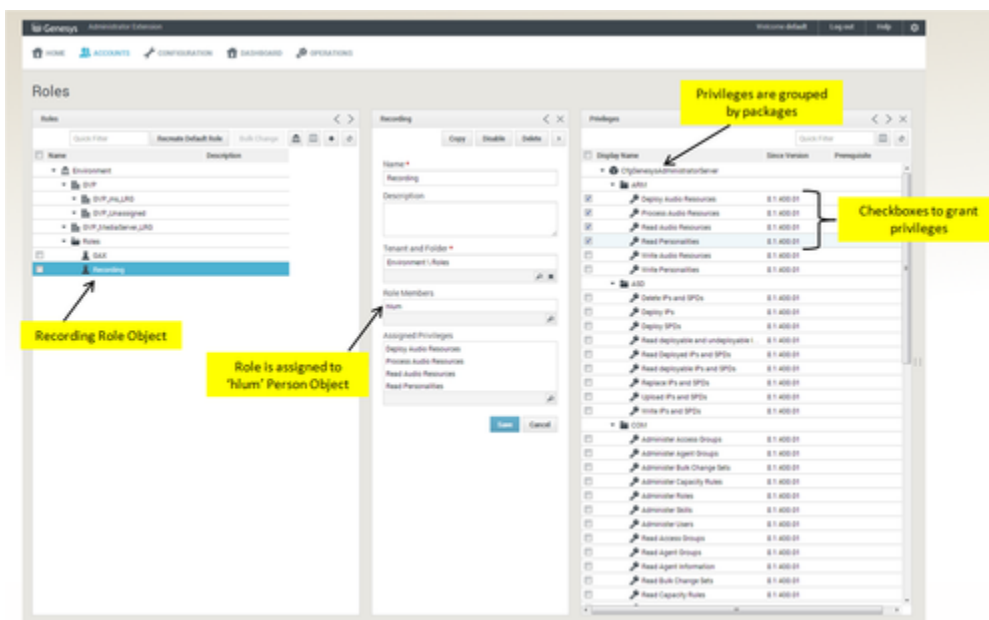
Access Control for Recording Users

Configuration Server stores user credentials and provides authorization to those applications requesting these credentials. The Genesys Administrator Extension logs into other dependent web services by forwarding the same credentials to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). These credentials are required only once.

Roles and privileges use the following basic concept:

- A user (Person object) can belong to one or more Access Groups (Access Group object).
- One or more roles (Role object) can be assigned to an Access Group or a user allowing the user to inherit one or more Roles.
- A Role is a collection of privileges.
- A privilege defines whether the Role is enabled for a specific action or function. The specific action or function can be granular based on the application needs.
- A user inherits all privileges from all Roles that the user belongs to.

The following screenshot provides an example of privileged assignment using Genesys Administrator Extension:



The user must have the following roles, privileges and permissions:

- Administration of Recording Certificates and Keys role privilege to enable uploading and deleting of recording certificates and assigning of recording certificates to IVR profiles

- Decrypt a Recording role privilege to view recording certificates and to access recording certificates to playback encrypted recordings
 - If the **show_cert_with_upload_privilege** option in the [rcs] section of the GAX application object is set to true, GAX will require the user to have the Administration of Recording Certificates and Keys and Decrypt a Recording role privileges to view recording certificates. The default value for the **show_cert_with_upload_privilege** option is false.
- Write access to the IVR profile object to change settings on the Recording tab of the IVR Profile
- Screen Recording Certificates role privilege to view assigned screen recording certificates
- Administration of Screen Recording Certificates role privilege to assign and remove assigned screen recording certificates
- Admin permission on the Interaction Recording Web Services (Web Services) server to view and manage screen recording certificates
- Recording Scheduler role privilege to enable access of recording scheduler settings, and to view recording schedule settings
- Administration of Recording Scheduler role privilege to enable administration of recording scheduler settings
- A role selected by your Administrator that will enable users to view their own recordings. Each administrator can create and choose the required role

The Recording Plug-in for GAX includes a **Solution Definition (SPD) file** that can be used to configure roles and access groups.

Recording Files

Each recording file is considered an object that is subject to access control at the user level. When a recording file is generated, the access control for the recording file is set based on the following criteria:

1. Access control is set based on the agent that was recorded. Agents are organized as an agent hierarchy; for example, the hierarchy can be a reporting structure in an organization.

When there are two agents (agent 1 and agent 2), that are both configured to be recorded and are on the same SIP server, and agent 1 calls agent 2, there will only be one recording and the recording will be associated with the agent that was called (agent 2).

In the case of an outbound call, the agent initiating the call should be configured to be recorded and not the trunk being used for the outbound call.

Note: with IVR recording, there is no associated agent for the specific segment of the call, since IVR is not a user.

2. Access control is set based on partitions. Partitions are set as a specific attached data in a call, and the attached data is typically set by a routing strategy.

To search and playback a recording file that is subject to access control, the user accessing SpeechMiner must be assigned to the appropriate Access Groups to access the recordings. If the user

accessing SpeechMiner is an agent, they are granted implicit playback access to their own recordings. However, if their agent hierarchy is changed, they will lose access to previous recordings. To enable agents to see their previous recordings, create a new recording access group in the format `/<old agent_hierarchy value>/<agent name>` and add the agent to this recording access group.

Agent Hierarchy

The agent hierarchy shows how the agents are organized in the hierarchy, and the hierarchy is represented as a field (`agent_hierarchy` option) in the recording section of the Annex tab in each Person (Agent Name) object.

The following example shows the agent hierarchy with four agents:

- /
 - Anthony
 - John
 - Agent1
 - Agent2
 - Paul
 - Agent3
 - Agent4

Agent1 and Agent2 are on John's team. John reports to Anthony.

To represent this structure, the following fields are stored in each of the Person object:

Person Object	agent_hierarchy Field
Agent1	/Anthony/John
Agent2	/Anthony/John
Agent3	/Anthony/Paul
Agent4	/Anthony/Paul

Important

When there are Person objects for items in the path, the path must contain the username for those persons. For example, for the hierarchy `/Anthony/John`, Anthony and John must match the usernames for Anthony and John.

If a user wants to listen to recordings handled by Agent1, the user needs to be granted access to either the Anthony, or the John Access Group. If a user is granted access to the Anthony Access Group, that user has access to recordings from all four agents, because all four agents are within Anthony's hierarchy.

Partitions

Partitions are arbitrary names that allows a contact center to partition recordings based on business rules. For example, partitions can be business groups such as sales, support, marketing, etc. To set one or more partitions to a recording, attach data to the call with the `GRECORD_PARTITIONS` key with a comma-separated list of partition names.

For example, if the `GRECORD_PARTITIONS` key is set to `/sales,/support`, the recording belongs to the `/sales` partition as well as the `/support` partition.

To access any recording belonging to a partition, the user must be assigned to an Access Group with the same name. For example, if user1 is assigned to the `/sales` Access Group, user1 can search and playback any recordings within the `/sales` partition

For examples about how to work with Agent Hierarchy and partitions, see [Agent Hierarchy Examples](#).

For more information about configuring the roles and permissions for Genesys Interaction Recording users, see [Access Control for Genesys Interaction Recording Users](#).

How the T-Library Works for GIR

The following sections describes the T-Lib interface for recording.

Enabling Call Recording

The T-Lib interface allows recording to be enabled in three ways:

1. Through configuration—Set the record option to true in the DN object to instruct SIP Server to enable full-time recording for this DN. This is an existing feature.
2. Set the extension in the TRouteCall request to enable recording selectively. When calling TRouteCall, add the record key in the extension attribute and set the value to destination for recording of inbound or outbound calls. This is an existing feature.
3. Add a new extension in the RequestPrivateService request to request call recording to be enabled on an existing connection as described in the following table:
RequestPrivateService—Request services that are supported only by certain T-Servers, and which are not covered by general feature requests.

Parameters	Description
AttrPrivateMsgID	This parameter is mandatory and must be equal to GSIP_RECORD_START.
AttrThisDN	This parameter is mandatory, and is the DN on behalf of which the operation is requested. It must be registered by the T-Client, but not necessary be a party on the call (for example, the supervisor may request recording of the agent's call).
AttrConnectionID	This parameter is mandatory and references the ID for the call to record.
AttrExtensions	<p>Additional request parameters:</p> <ul style="list-style-type: none">• record (string)—Set to source to record from this DN referenced in this connection. Set to destination to record from the other DN referenced in this connection. This parameter is optional, and defaults to source.• Id (string)—Adds a recording identifier to the recording session. This identifier must be globally unique and is passed back in the recording session. This parameter is optional and if not present, Media Server constructs a unique identifier.• Dest (string)—Overrides the default SIP

Parameters	Description
	<p>location of the recording server. This parameter is optional.</p> <ul style="list-style-type: none"> • Dest2 (string)—Overrides a second SIP location of the recording server for duplication of recording. This parameter is optional. • Params (string)—Additional parameters can be passed as generic name-value pairs. These parameters will show up in the recording session.
AttrReasons	The reasons. These are processed the same as for all other T-Library requests.

Important

SIP Server responds to the request with either EventACK to confirm the acceptance of the request, or EventError if the operation cannot be performed.

Runtime Control of Recording

When the recording session is established, the T-Lib interface allows run-time control of the recording for pause, resume, and stop. The following table describes a new extension for RequestPrivateService:

RequestPrivateService—Request services that are supported only by certain T-Servers, and which are not covered by general feature requests.

Parameters	Description
AttrPrivateMsgID	<p>Specifies the operation. Choose one of the following values:</p> <ul style="list-style-type: none"> • GSIP_RECORD_STOP—Stop the recording. • GSIP_RECORD_PAUSE—Pause the recording. • GSIP_RECORD_RESUME—Resume the recording.
AttrThisDN	The DN on behalf of which the operation is requested. It must be registered by the T-Client, but not necessarily be a party on the call.
AttrConnectionID	References the ID for the call being recorded.
AttrExtensions	<p>Additional request parameters:</p> <ul style="list-style-type: none"> • Params (string)—Additional parameters can be

Parameters	Description
	passed as generic name-value pairs that modify the recording session.
AttrReasons	The reasons. These are processed the same as for all other T-Library requests.

Important

SIP Server responds to the request with either EventACK to confirm the acceptance of the request, or EventError if the operation cannot be performed.

Recording Indication

Two mechanisms that enable SIP Server to provide recording indication:

- After SIP Server successfully starts recording on Media Server, SIP Server updates the UserData attribute of the call with GSIP_REC_FN using the file name of the recording. A T-Lib client monitoring the call receives an EventAttachedDataChanged with GSIP_REC_FN. This is existing functionality for legacy Stream Manager recording. For clients who only want to know if a recording has been enabled any time during the call, this userdata is sufficient.
- For clients such as, Interaction Workspace, who need to render the current recording state for the call, GSIP_REC_FN is not sufficient as a recording indicator. SIP Server provides a new value GSIP_RECORD for the UserData attribute to provide the current state of recording for this call. Whenever SIP Server knows there is a change in recording, SIP Server sends an EventAttachedDataChanged with GSIP_RECORD to update the value of the key. This key has three values:
 - On—Recording is currently in progress.
 - Off—No recording in progress.
 - Paused—Recording is currently in progress but no media is currently captured.

User Interfaces

Genesys Interaction Recording uses three User Interfaces to implement and manage a call and screen recording environment:

- **Playback Interface**—The SpeechMiner component.
- **Administrator Interface**—The Genesys Administrator Extension component.
- **Agent Interface**—The Workspace component.

Playback Interface

An agent, supervisor, or an administrator uses the playback interface to search, query, report, and retrieve call and screen recordings from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier). SpeechMiner uses the Recording Crypto Server to retrieve all recordings from Interaction Recording Web Services (Web Services), and automatically decrypts recordings that are encrypted.

Administrator Interface

The Genesys Administrator Extension interface provides three functional areas:

1. Administration of recording policies—Allows a tenant administrator to configure recording policies and recording rules. Configuration parameters include retention period, storage management, and file format. Some recording policies are part of existing configuration objects such as IVR Profile and DN and can be managed through the existing interfaces. This recording administration interface intends to consolidate the configuration of the policies to assist the administrator.
2. Call recording maintenance—Provides the maintenance function to configure the storage retention period.
3. Key management—Allows a security administrator to manage public and private keys for handling encryption and decryption of call recording. The **Recording Crypto Server** provides a REST API to manage the public and private keys, while the front-end UI is hosted through **Genesys Administrator Extension**. It is recommended that the Recording Crypto Server is hosted in a secure zone that is not accessible outside an enterprise firewall, while Genesys Administrator Extension can be configured to be accessible externally.

The recording administrator interface is designed as a **Genesys Administrator plug-in**, and the plug-in provides access to the different functional areas via different services. The plug-in is a proxy for the Call Recording API (**Web Services**) and the Key Management API (Recording Crypto Server).

Agent Interface

An agent handling calls will have additional interface elements to provide recording indication to the agent, as well as run time recording controls (start/stop/pause/resume). **Workspace** either directly sends recording controls directly to SIP Server, or indirectly via Interaction Recording Web Services (Web Services).

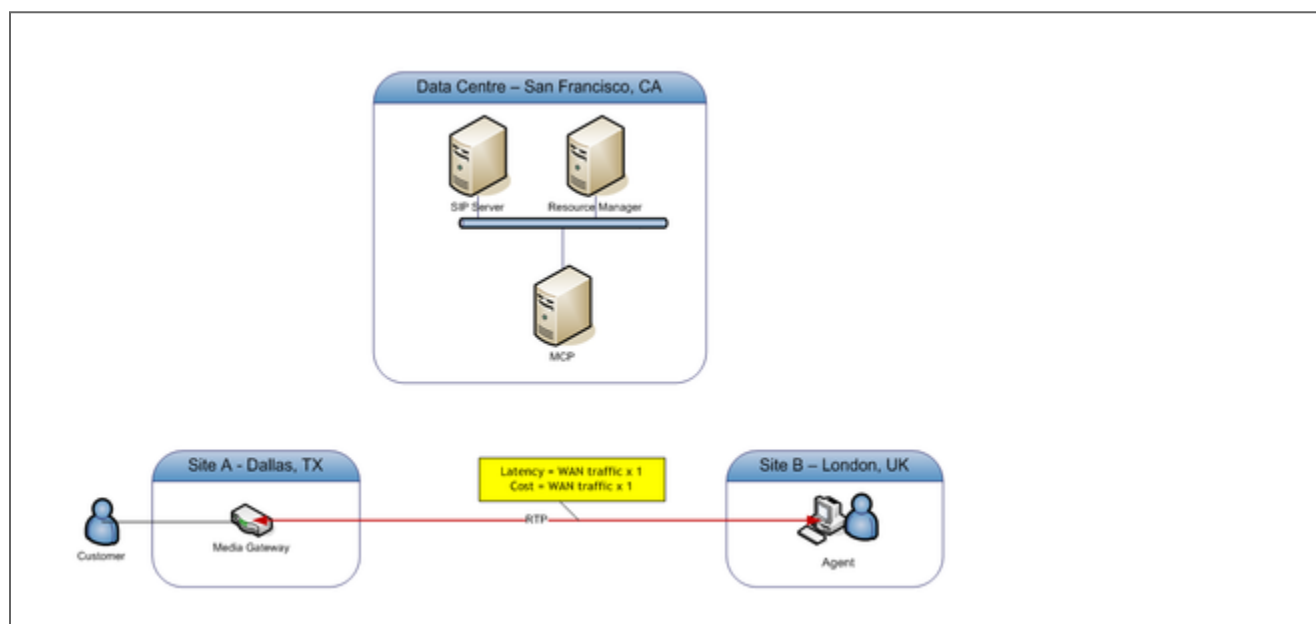
Geo-Location

The Genesys Interaction Recording solution uses geo-location to provide a deployment using multiple data centers with the capability to select specific pools of Media Control Platform instances that are located at specific sites with specific recording storage, specified through an IVR Profile.

Geo-location support provides a **multiple data center** deployment with the capability to select specific pools of MCPs that are located at specific sites, with appropriate storage. The main motivation for selecting specific MCPs is either to minimize WAN traffic or to minimize the latency introduced to a conversation when recording is enabled.

No Recording

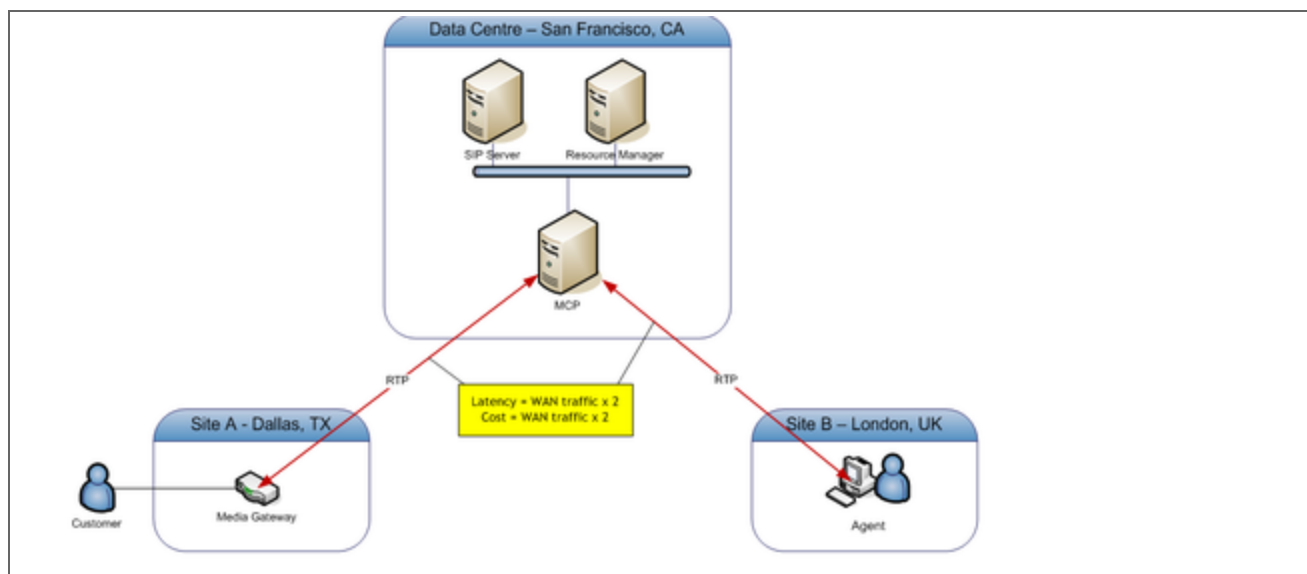
Customer and Agent Call Across the WAN with No Recording



In a typical scenario, the customer might be calling into a contact center site with a media gateway, and the agent is located in a different site from the media gateway.

Recording in Data Center

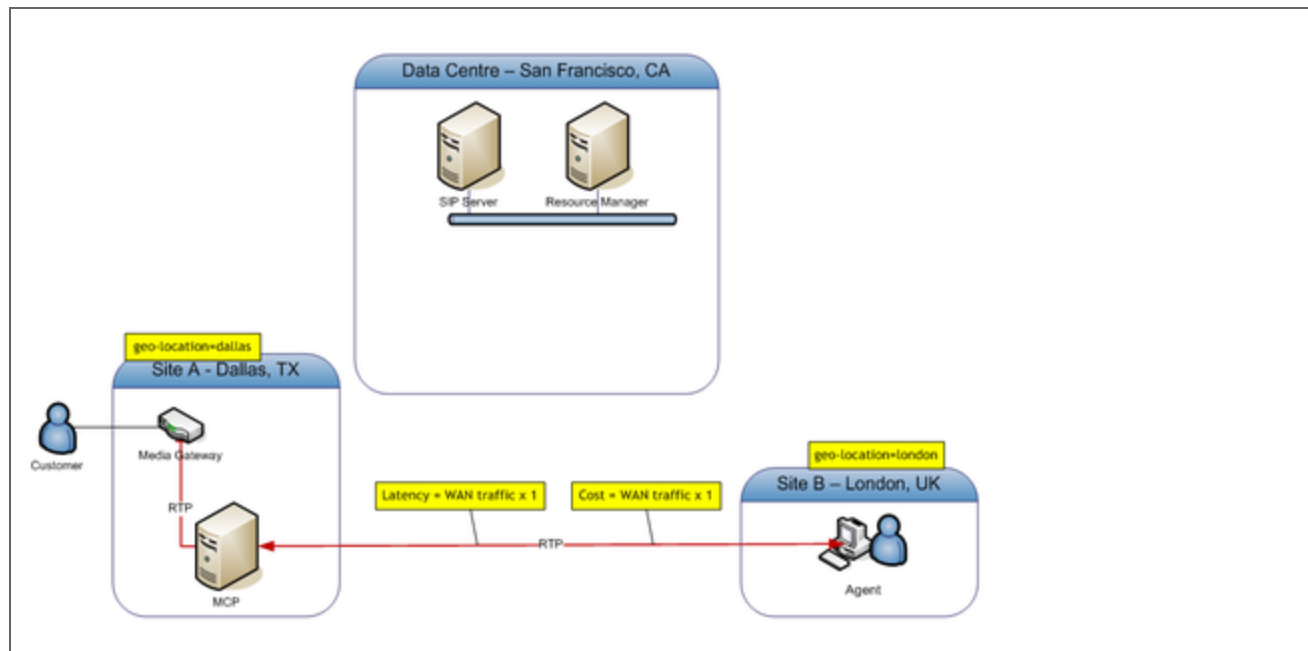
Customer and Agent Call Across the WAN with Recording in Data Center



When the Media Control Platform (MCP) is located at the data center site, the deployment needs to double the WAN traffic because the media path needs to be bridged through the data center. This increased WAN traffic leads to increased latency of the media path by doubling the WAN path.

Recording with Geo-location

Customer and Agent Call Across the WAN with Recording with Geo-location at the Customer Site



To minimize latency, the geo-location feature has been introduced in SIP Server and Resource Manager. This feature allows MCPs to be deployed in a remote site that is close to one of the parties in the call. This diagram is a deployment that places MCP in Dallas as set in the geo-location=dallas parameter.

Geo-location for MCP is considered separately by the Resource Manager.

For information about configuring geo-location, see the content describing Geo-Location within the [SIP Server Deployment Guide](#).

Audio Tones

To meet the regulatory requirements, some deployments require the system to periodically generate an audio tone to notify the participants in a call that the call is currently being recorded. The following sections describe how to work with audio tones:

Applying Audio Tones

Audio tones can be generated either as all-party consent or one-party consent:

- All-party consent requires that all parties in the call being recorded hear the audio tone periodically.
- One-party consent requires that only one of the parties on the call to hears the audio tone. The consent is configurable on Media Server.

There is a difference between all-consent/one-party-consent and applying the beep to certain calls:

- All-consent/one-party-consent setting is a global system setting on the MCP process.

These parameters are **configured** as Recording Parameters parameters in the IVR profile:

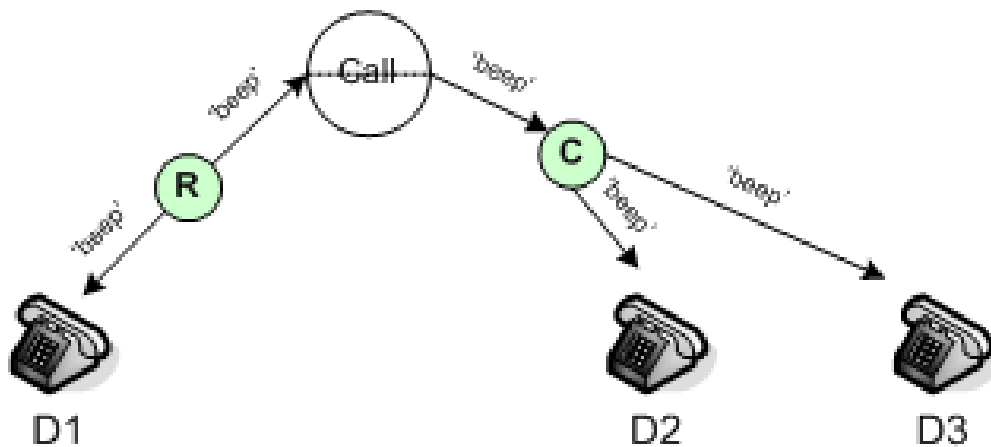
Parameter Name	Description
Recording Alert Tone	The URI of the audio tone. If the URI is set to an empty string, or not defined, or resolves to a bad URI, then no audio tone is applied to the call. No other notifications are generated by Media Server (for example, MSML events) when no audio tone is applied. Only .wav files are supported. QTMF tones and files stored in sub-directories with multiple codecs that are supported by Media Server are not supported. For example, "music/beep" cannot be specified for this option, even though it is valid for other Media Server treatments.
Frequency of Recording Alert Tone	The length of time, in milliseconds, between playing the audio tone. This is a mandatory parameter if the Recording Alert Tone parameter is defined, otherwise no audio tone is applied. The minimum accepted value is 1500 (if a smaller value is specified, 1500 will be used). In addition, if the Frequency of Recording Alert Tone parameter is not present, MCP applies the default value of 30000 instead of not applying tone.

Recording Conference Audio Tones

When recording a conference, there are two Media Servers involved in the call:

- One for recording the recording DN.
- One for mixing media for other parties.

The audio tone is generated from the recording Media Server and is propagated to the conferencing Media Server. In order to ensure that all parties get the consent, set `record_recorddnhearstone`, and the `record_otherdnhearstone` options in the conference section of the Media Server application to true.



When the recording is paused, no audio tone is generated. When the recording is resumed, the audio tone is applied.

For information about configuring audio tones, see [Audio Tones](#).

Reporting

Reports are summaries and analyses of interaction and external metadata. You can generate reports for analysis, view report details and status, and share the data with users throughout the enterprise. You can view reports in your browser, print them or send them via email.

To help you monitor your business, SpeechMiner UI offers a wide range of standard reports that can be customized to better suit your needs. Depending on the type of report, the results may be presented as lists or data and/or in graphic form. In some reports, you can drill down within a report to see additional details.

To learn more about Reports refer to the **Reports** section in the SpeechMiner UI User Manual and/or the SpeechMiner UI Online Help.

Appendixes

The follow sections provide examples and supplementary information that can help you configure your Genesys Interaction Recording solution.

- [Example Solution Definition SPD File](#)
- [Disk Storage Recommendations](#)
- [Sample Certificate and Key File Generation](#)
- [Interaction Recording Web Services \(or Web Services if you're using version 8.5.210.02 or earlier\) Group Settings](#)
- [Creating Folder Hierarchy for Recording Storage](#)
- [Agent Hierarchy Examples](#)
- [Secure Transport Configuration](#)
- [Recovering Metadata for SpeechMiner](#)
- [Automated Recovery of Recordings](#)
- [Troubleshooting](#)
- [Understanding Genesys Interaction Recording](#)
- [Minimum Recommended Versions](#)
- [GIR Alarms](#)

Example Solution Definition SPD File

The Recording Plug-in for GAX includes a Solution Definition (SPD) file. This file includes the options to create default Recording and SpeechMiner roles. For more information about deploying and configuring SPD files in GAX, see [Solution Package Definitions](#) and [Authoring Solution Definitions](#).

Important

- The SPD file only supports Configuration Servers that are provisioned in English.
- The SPD file does not support a configuration unit or a site that includes folders with the name Persons, Access Groups, or Roles.

The SPD file can optionally perform the following actions:

1. Creates T-Server and Media Control Platform (MCP) recording alarm conditions:
 - T-Server Record Failure
 - MCP Record Failure
 - MCP Record Post Failure
 - MCP Record Post Backlog Exceeded
 - MCP Too Many Recordings to Recover
2. Configures the T-Server recording alarm interval time:
 - Sets the recording-failure-alarm-timeout parameter in the TServer section on all T-Server applications.
3. Configures the MCP recording alarm interval time:
 - Sets the alarminterval parameter in the mcp section on all MCP applications.
4. Creates the recording certificate roles:
 - Record Certificate Administrator privileges—Administration of Recording Certificates and Keys and Decrypt a Recording
 - Record Certificate User privilege—Decrypt a Recording
5. Creates the default SpeechMiner roles:
 - SpeechMiner Administrator
 - SpeechMiner Power User
 - SpeechMiner Regular User
 - SpeechMiner Event Audit
 - SpeechMiner SMART Power User

- SpeechMiner SMART User

For more information about the SpeechMiner roles, see the SpeechMiner User Manual.

6. Creates the base SpeechMiner access groups:

- Recording access group folder
- Recording folder "/" access group

7. Configures record archiving:

- Creates a user (under Environment or a tenant) to be used for SpeechMiner API calls related to record archiving.
- Sets the user in the Environment or tenant recording archive configuration.

Disk Storage Recommendations

This section provides disk storage recommendations for premise Genesys Interaction Recording deployments.

MP3 is the archival and playback format; therefore, the storage sizing is a direct function of the MP3 bitrate, the duration of recording files, and the retention period. The estimates below are based on calculations using MP3 recordings at 16 kbit/s.

Storage Size Estimations

There are three sizing levels considered:

1. Small—0 to 150 seats
2. Medium—150 to 300 seats
3. Large—300 to 1000 seats

The following tables provide estimates per level.

	Small			Medium			Large		
Average Calls per Day	11400			22800			96000		
Average Call Duration	210			210			210		
Storage Period (Days)	100			100			100		

Quality	Small	Medium	Large		Quality	Small	Medium	Large
Average Size per Day in GB					Average Storage in TB			
16 kilobytes per second	4.6	9.2	38.4	16 kilobytes per second	0.44	0.9	3.76	

Sample Certificate and Key File Generation

Before generating the key file, create a Root Certificate Authority (CA). For more information about creating a CA using OpenSSL or Windows Certificate Services, see the Genesys Security Deployment Guide at [Certificate Generation and Installation](#).

The certificates generated using the procedures in the Genesys Security Deployment Guide can be used for recording encryption only if the certificate fields are set appropriately for the "HOST" certificate.

You can choose to generate to CA files themselves using non-Genesys procedures—for example, if you have a system with OpenSSL installed in your environment, a more general certificate can be created directly using OpenSSL. The following commands use an OpenSSL Root CA, and must be executed from the Root CA directory:

```
openssl req -nodes -newkey rsa:2048 -keyout cert.key -out cert.req  
openssl ca -out cert.pem -infiles cert.req
```

For the steps required to encrypt voice and screen recordings, see [Encrypting and Provisioning Certificates](#).

Interaction Recording Web Services (Web Services) Group Settings

You use the API to configure the settings groups that Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) uses for Genesys Interaction Recording, so that you can maintain these settings groups after deployment.

Important

For API calls mentioned below, the username and password must be of a user with **Contact Center Admin** role defined in the Genesys configuration environment.

Screen Recording Storage Settings

Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) provides location-based storage settings so that the Interaction Recording Web Services (Web Services) node can access external storage. The next sections describe how to use the API to create, update, delete and get the storage settings.

Create Storage Settings

The following example shows how to create the storage settings with the properties described in the [table](#) below:

POST `http://<Web Services-cluster-address>/api/v2/settings/screen-recording`

```
{
  "name": "storage",
  "location": "/US/CA",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "user",
        "password": "pass",
        "storagePath": "http://<IP Address>:<Port>/recordings"
      }
    }
  ]
}
```

Update Storage Settings

The following example shows how to create the storage settings with the properties described in the [table](#) below:

PUT <http://<Web Services-cluster-address>/api/v2/settings/screen-recording>

```
{
  "name": "storage",
  "location": "/US/CA",
  "value": [
    {
      "storageType": "webDAV",
      "active": true,
      "credential": {
        "userName": "user",
        "password": "pass",
        "storagePath": "http://<IP Address>:<Port>/recordings"
      }
    }
  ]
}
```

Delete Storage Settings

The following example shows how to delete the storage settings:

Important

The location parameter must match the location you want to delete. The wildcard (*) character cannot be used.

DELETE <http://<Web Services-cluster-address>/api/v2/settings/screen-recording?location=/US/CA>

```
{
  "name": "storage"
}
```

Retrieve the Storage Settings

The following examples shows how to retrieve the storage settings with the properties described in the [table](#) below:

For All Locations

GET http://<Web Services-cluster-address>/api/v2/settings/screen-recording?location=*&ignoreParentLocations=true

```
{
  "statusCode": 0,
  "settings":
```

```
[
  {
    "name": "storage",
    "values": [
      {
        "location": "/US/CA",
        "value": [
          {
            "storageType": "webDAV",
            "active": true,
            "credential": {
              "userName": "****",
              "password": "****",
              "storagePath": "http://<First IP Address>:<Port>/recordings"
            }
          }
        ],
        "location": "/US/OR",
        "value": [
          {
            "storageType": "webDAV",
            "active": true,
            "credential": {
              "userName": "****",
              "password": "****",
              "storagePath": "http://<Second IP Address>:<Port>/webdav"
            }
          }
        ]
      }
    ]
  },
  {
    "key": "name"
  }
]
```

For a Single Location

GET <http://<Web Services-cluster-address>/api/v2/settings/screen-recording?location=/US/CA&ignoreParentLocations=true>

```
{
  "statusCode": 0,
  "settings": [
    {
      "name": "storage",
      "values": [
        {
          "location": "/US/CA",
          "value": [
            {
              "storage": "webDAV",
              "active": true,
              "credential":

```

```

    {
      "userName": "****",
      "password": "****",
      "storagePath": "http://IP Address>:<Port>/recordings"
    },
  ],
}
],
"key": "name"
}

```

Property Descriptions

Property	Description
storageType	The storage location. Set it to webDAV, to store the recordings on the physical server.
active	Determines whether the location-based storage is enabled and available to store the recordings. If set to true, the corresponding Interaction Recording Web Services node(s) (or Web Services if you are using version 8.5.210.02 or earlier) will upload and save the screen recordings to this storage resource. If set to false, or the property is missing, recordings are not uploaded to this storage resource, but media located on this storage can be played back.
credential	<ul style="list-style-type: none"> • userName—The user that has access to the WebDAV storage location. • password—The password for the WebDAV storage location. • storagePath—The URL of the WebDAV storage location.
key	The name of the "key" attribute for this group's settings. Whenever an individual setting needs to be modified, this "key" attribute is used to identify the setting. The value of the "key" attribute must be unique for every setting and is read-only after the setting has been created. A setting cannot be created without this attribute.

The location matching for these settings are hierarchical matches. For more information and examples, see [Hierarchical Location Matching](#).

Screen Recording Decrypt URI Prefix

Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) defers the decryption process of screen recordings to the Recording Crypto Server. Interaction Recording Web Services (Web Services) provides a URI that directs the SpeechMiner application to the Recording Crypto Server for the decrypted screen recording, and for the mediaUri member in a Get Screen Recording Meta-Data response. Note that the value must be set to the URL of the Recording Crypto Server instance.

The following sections describe how to use the API to create, update, delete and retrieve the decrypt URI prefix.

Create Decryption URI Prefix Setting

The following example shows how to create the decryption URI prefix settings:

POST http://<Web Services-cluster-address>/api/v2/settings/screen-recording

```
{
  "name": "decrypt-uri-prefix",
  "location": "/US/CA",
  "value": "https://<IP Address>/rcs"
}
```

Update Decryption URI Prefix Setting

The following example shows how to update the decryption URI prefix settings:

PUT http://<Web Services-cluster-address>/api/v2/settings/screen-recording

```
{
  "name": "decrypt-uri-prefix",
  "location": "/US/CA",
  "value": "https://<IP Address>/rcs"
}
```

Delete Decryption URI Prefix Setting

The following example shows how to delete the decryption URI prefix settings:

Important

The location parameter must match the location you want to delete. The wildcard (*) character cannot be used.

DELETE http://<Web Services-cluster-address>/api/v2/settings/screen-recording?location=/US/CA

```
{
  "name": "decrypt-uri-prefix"
}
```

Retrieve the Decryption URI Prefix Setting

The following examples shows how to retrieve the decryption URI prefix settings:

For All Locations

```
GET http://<Web Services-cluster-address>/api/v2/settings/screen-
recording?location=*&ignoreParentLocations=true
{
  "name": "decrypt-uri-prefix",
  "values":
  [
    {
      "location": "/US/CA",
      "value": "https://<First IP Address>:<Port>/rcs"
    },
    {
      "location": "/US/OR",
      "value": "https://<First IP Address>:<Port>/rcs"
    }
  ]
}
```

For a Single Location

```
GET http://<Web Services-cluster-address>/api/v2/settings/interaction-recording?location=/US/
CA&ignoreParentLocations=true
{
  "name": "decrypt-uri-prefix",
  "values":
  [
    {
      "location": "/US/CA",
      "value": "https://<IP Address>:<Port>/rcs"
    }
  ]
}
```

The location matching for these setting are also hierarchical matches. See the example described above in the [Screen Recording Storage Settings](#).

Local Decrypt URI Prefix for Call Recording and Screen Recording

Interaction Recording Web Services defers the decryption process of call recordings and screen recordings to the Recording Crypto Server. Interaction Recording Web Services can optionally provide a URI that directs the SpeechMiner application to Interaction Recording Web Services instead of directing it to the Recording Crypto Server for the decrypted call recording or screen recording, and for the `mediaUri` member in a GET Call Recording Metadata response or Get Screen Recording Metadata response. To do this, the following are required:

- Set the `api-recordings-decryption-proxying` feature flag. If you're using Interaction Recording and Web Services component, [click here](#) for details. If you're using Web Services, [click here](#) for details.

- Add the local-decrypt-uri-prefix setting (described in this section)
- Remove the decrypt-uri-prefix setting (described [above](#))
- Remove External RCS URI from Speechminer configuration for encrypted screen recordings. For more information, refer to the table that describes the parameters under Step 10 in [Configuring SpeechMiner users](#).

The following sections describe how to use the API to create, update, delete and retrieve the local decrypt URI prefix. Note that the value property in these REST calls must be set to the URL of the Recording Crypto Server instance.

Important

This option only applies to Interaction Recording Web Services 8.5.204.02 or newer.

Create Local Decryption URI Prefix Setting

The following example shows how to create the local decryption URI prefix settings:

POST `http://<Web Services-cluster-address>/api/v2/settings/interaction-recording`

```
{
  "name": "local-decrypt-uri-prefix",
  "location": "/US/CA",
  "value": "https://<IP Address>/rcs"
}
```

Update Local Decryption URI Prefix Setting

The following example shows how to update the local decryption URI prefix settings:

PUT `http://<Web Services-cluster-address>/api/v2/settings/interaction-recording`

```
{
  "name": "local-decrypt-uri-prefix",
  "location": "/US/CA",
  "value": "https://<IP Address>/rcs"
}
```

Delete Local Decryption URI Prefix Setting

The following example shows how to delete the local decryption URI prefix settings:

Important

The location parameter must match the location you want to delete. Do not use the wildcard (*) character.

DELETE http://<Web Services-cluster-address>/api/v2/settings/interaction-recording?location=/US/CA

```
{
  "name": "local-decrypt-uri-prefix"
}
```

Retrieve the Local Decryption URI Prefix Setting

The following examples shows how to retrieve the local decryption URI prefix settings:

For All Locations

GET http://<Web Services-cluster-address>/api/v2/settings/interaction-recording?location=*&ignoreParentLocations=true

```
{
  "name": "local-decrypt-uri-prefix",
  "values": [
    {
      "location": "/US/CA",
      "value": "https://<First IP Address>:<Port>/rcs"
    },
    {
      "location": "/US/OR",
      "value": "https://<First IP Address>:<Port>/rcs"
    }
  ]
}
```

For a Single Location

GET http://<Web Services-cluster-address>/api/v2/settings/interaction-recording?location=/US/CA&ignoreParentLocations=true

```
{
  "name": "local-decrypt-uri-prefix",
  "values": [
    {
      "location": "/US/CA",
      "value": "https://<IP Address>:<Port>/rcs"
    }
  ]
}
```

The location matching for these setting are also hierarchical matches. See the example described above in the [Screen Recording Storage Settings](#).

SpeechMiner Settings

For MLM purge, label/tagging, and deletion protection operations, Interaction Recording Web Services issues requests to the SpeechMiner Interaction Receiver to update the corresponding records in the SpeechMiner database. The Interaction Recording Web Services node that executes the operation determines the SpeechMiner API URI prefix and the credentials based on the SpeechMiner group setting with the location-based interaction-receiver settings. The following sections describe how to

use the API to create, update, delete and retrieve the SpeechMiner settings.

Create SpeechMiner Settings

The following example shows how to create the Interaction Receiver settings with the properties described in the [table](#) below:

```
POST curl -u <ops-user>:<ops-pass> http://<Web Services-cluster-address>/api/v2/settings/speechminer
```

```
{
  "name": "interaction-receiver",
  "location": "/US/CA",
  "value": {
    "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
    "userName": "interaction receiver user name",
    "password": "interaction receiver password"
  }
}
```

Update SpeechMiner Settings

The following example shows how to update the Interaction Receiver settings with the properties described in the [table](#) below:

```
PUT curl -u <ops-user>:<ops-pass> http://<Web Services-cluster-address>/api/v2/settings/speechminer
```

```
{
  "name": "interaction-receiver",
  "location": "/US/CA",
  "value": {
    "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
    "userName": "interaction receiver user name",
    "password": "interaction receiver password"
  }
}
```

Delete SpeechMiner Settings

The following example shows how to update the Interaction Receiver settings.

Important

The location parameter must match the location you want to delete. The wildcard (*) character cannot be used.

```
DELETE http://<Web Services-cluster-address>/api/v2/settings/speechminer?location=/US/CA
```

```
{
```

```
    "name": "interaction-receiver"
  }
}
```

Retrieve SpeechMiner Settings

The following examples show how to update the Interaction Receiver settings with the properties described in the [table](#) below:

For All Locations

GET http://<Web Services-cluster-address>/api/v2/settings/speechminer?location=*&ignoreParentLocations=true

```
{
  "statusCode": 0,
  "settings":
  [
    {
      "name": "interaction-receiver",
      "values":
      [
        {
          "location": "/US/CA",
          "value":
          {
            "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
            "userName": "****",
            "password": "****"
          }
        },
        {
          "location": "/US/OR",
          "value":
          {
            "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
            "userName": "****",
            "password": "****"
          }
        }
      ]
    }
  ],
  "key": "name"
}
```

For a Single Location

GET <http://<Web Services-cluster-address>/api/v2/settings/speechminer?location=/US/CA&ignoreParentLocations=true>

```
{
  "statusCode": 0,
  "settings":
  [
    {
      "name": "interaction-receiver",
      "values":
      [
        {
          "location": "/US/CA",
          "value":
          {

```

```

        "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
        "userName": "****",
        "password": "****"
      }
    ]
  },
  "key": "name"
}

```

Property Descriptions for SpeechMiner

Property	Description
uri-prefix	The URL of the Interaction Receiver server.
userName	The userName that is used to access the Interaction Receiver server.
password	The password that is used to access the Interaction Receiver server.
key	The name of the "key" attribute for this group's settings. Whenever an individual setting needs to be modified, this "key" attribute is used to identify the setting. The value of the "key" attribute must be unique for every setting and is read-only after the setting has been created. A setting can not be created without this attribute.

Important

The Interaction Receiver username and password must be the same as the username and password configured in both of the following sections:

- **Configuring SpeechMiner Interaction Receiver Authorization Header** in the **Recording Destinations** section of [IVR profile configuration](#).
- [Step 5](#) of Configuring SpeechMiner users

The location matching for this setting is a hierarchical match. For more information and an example, see [Hierarchical Location Matching](#).

Corrections and Changes to Behavior

In GIR release 8.5.205.01 and prior, the location based matching of the **interaction-receiver** setting in the **speechminer** settings group was exact match, that is, a setting would have to have the location property specified exactly the same as the Web Services node's nodePath from the Web Services **application.yaml** file (for example, /US/10.2.100.100), if the node was to update recording information within SpeechMiner. This has been changed in GIR release 8.5.206.01 and onwards for this setting to hierarchical based location matching. This is also true for Interaction Recording Web Services. In other words, the **interaction-receiver** setting in the **speechminer**

settings group must have a location setting that is one of the folders of the node's nodePath from the Interaction Recording Web Services (Web Services) **application.yaml** file (if you are using Web Services and Applications version 8.5.201.09 or earlier use the value of the nodePath from an MLM node's server-settings.yaml file instead) in order for that setting to be used by the node (for example, / or /US for the MLM node nodePath of /US/10.2.100.100) to update recording information within SpeechMiner.

Impact to Customers Upgrading from GIR release 8.5.205.01 and prior

Customers upgrading their Web Services installation that have SpeechMiner, and have scheduled MLM purge tasks must update their **interaction-receiver** settings in the **speechminer** settings group by removing the last (node) portion of the location setting of each value. If, as a result, any settings is duplicated, these settings can be merged. For example, if the following is the settings for an existing installation:

```
GET http://<Web Services-cluster-address>/api/v2/settings/
speechminer?location=*&ignoreParentLocations=true
{
  "statusCode": 0,
  "settings":
  [
    {
      "name": "interaction-receiver",
      "values":
      [
        {
          "location": "/US/10.2.100.1",
          "value":
          {
            "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
            "userName": "*****",
            "password": "*****"
          }
        },
        {
          "location": "/US/10.2.100.101",
          "value":
          {
            "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
            "userName": "*****",
            "password": "*****"
          }
        }
      ]
    }
  ],
  "key": "name"
}
```

They are changed to:

```
GET http://<Web Services-cluster-address>/api/v2/settings/
speechminer?location=*&ignoreParentLocations=true
{
  "statusCode": 0,
  "settings":
  [
    {
      "name": "interaction-receiver",
      "values":
```

```

    [
      {
        "location": "/US",
        "value": {
          "uri-prefix": "http://<IP Address>:<Port>/interactionreceiver",
          "userName": "****",
          "password": "****"
        }
      }
    ],
    "key": "name"
  }
}

```

Voice Processor Tenant Level Settings

As the Voice Processor is designed to support Genesys cloud multi-tenancy model, settings that may vary from tenant to tenant are stored in an RWS group settings called **rps-provisioning**. For more information on these settings, see [Voice Processor](#).

Settings Groups Location Matching

Settings found within a Settings Group are shared by all Interaction Recording Web Services nodes (or Web Services nodes if you're using version 8.5.210.02 or earlier) nodes. For some of the Settings Groups, each setting has a *location* property that helps, along with the location matching behavior of the Setting Group, an individual Web Services node to determine which settings apply to them. Currently there are two location matching behaviors:

- **Exact location matching**—Allows a specific setting in a Settings Group to be applied to one Web Service node only, but does not allow a setting to be applied to a group of location-related Web Services nodes.
- **Hierarchical location matching**—Allows a setting to be applied to a group of location-related Web Services nodes, but allows another setting to override that setting for a single Web Services node in the group or a more closely location-related group of Web Services nodes within the original group.

Exact Location Matching

Web Services provides location-based settings for gir-scheduler operations. These settings are an exact match.

For example:

The Web Services Cluster is provisioned with gir-scheduler settings with locations "/US/CA/10.10.15.83", "/US/CA/10.10.15.84", "/CA/ON/10.10.15.85", and "/US/CA".

Nodes have the following nodePaths:

- Web Services node1 has nodePath = /US/CA/10.10.15.83

- Web Services node2 has nodePath = /US/CA/10.10.15.84
- Web Services node3 has nodePath = /CA/ON/10.10.15.85

The results are:

- node1 matches "/US/CA/10.10.15.83" and so will use gir-scheduler settings for that location.
- node2 matches "/US/CA/10.10.15.84" and so will use gir-scheduler settings for that location.
- node3 matches "/CA/ON/10.10.15.85" and so will use gir-scheduler settings for that location.

No node matches "/US/CA" and so those gir-scheduler settings will not be used by any node.

Hierarchical Location Matching

Web Services also provides the location-based hierarchical matches settings groups for storage location.

For example, if the Web Services node is provisioned with storage settings "/US/CA", "/US/OR", and "/US", the Nodes have the following nodePaths:

- Web Services node1 has nodePath = /US/CA/10.10.15.83
- Web Services node2 has nodePath = /US/NY/10.10.15.84
- Web Services node3 has nodePath = /CA/ON/10.10.15.85

The results are:

- node1 matches "/US/CA" and "/US". "/US/CA" will be selected as storage which match the location with "closest" (that is, closest in the hierarchy represented in the location property) match.
- node2 matches "/US" only, so "/US" storage will be selected.
- node3 matches none - it will have no access to external storage.

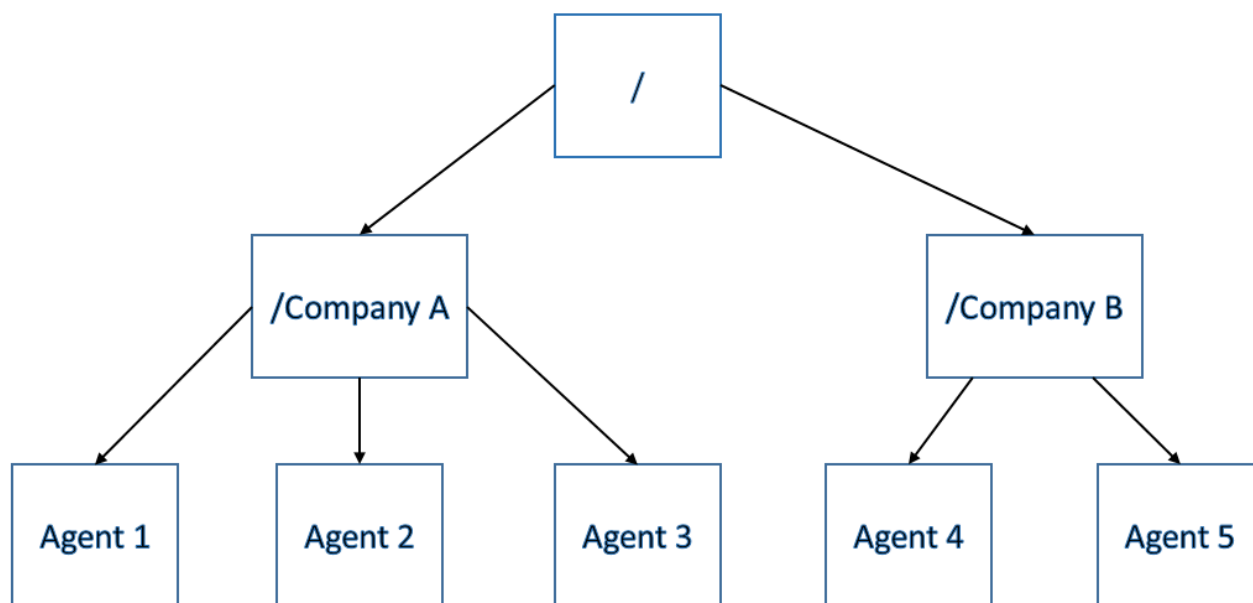
Agent Hierarchy and Partitioning Examples

Genesys Interaction Recording allows agents to be organized in a hierarchy to provide the ability to enforce access control of call recordings.

The basic rules of recording access control are defined in [Access Control for Voice Recording Users](#). The examples provided in that section are just one way of managing an agent hierarchy which is based on the organization reporting hierarchy. Genesys Interaction Recording allows for building agent hierarchies based on your particular business requirement.

The following sections provide examples of how you can apply agent hierarchy in your environment. In addition, the use of Partitioning is also discussed.

Multiple Third-Party Companies



Your contact center is comprised of agents from several third-party companies, and your business goal is to restrict users of each company from accessing the recordings of the other companies.

In this scenario, you would configure your agent hierarchy as follows:

On the **Annex** of each **Person** object, in the **[recording]** section, set the **agent_hierarchy** option:

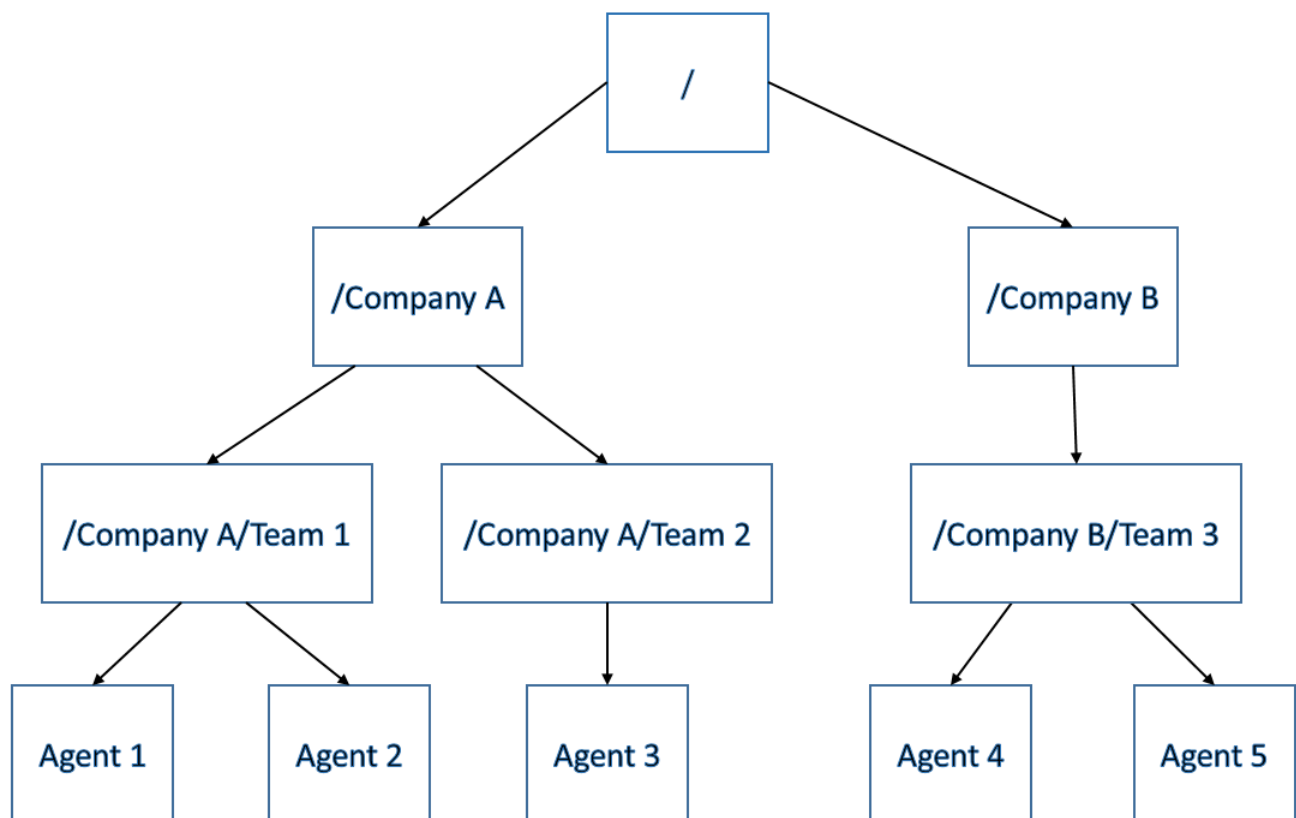
- Agent 1 to /Company A

- Agent 2 to /Company A
- Agent 3 to /Company A
- Agent 4 to /Company B
- Agent 5 to /Company B

This also means that the agents who belong to Company A should be assigned to the "/Company A" access group only. The following table describes the Access Group objects and available permissions:

Access Group Member	View Recordings of	Roles Assigned (Examples)
/	All agents	Super Administrators
/Company A	Agents 1, 2, 3	Supervisors for Company A
/Company B	Agents 4, 5	Supervisors for Company B

Multiple Third-Party Companies with Teams



Extending the above example, each third-party company has multiple teams managed by a

supervisor. Your business requirement is to restrict the recording access of the supervisor to their team members only. For each company, there are quality agents who can review all recordings within the third-party organization.

In this scenario, you would configure your agent hierarchy as follows:

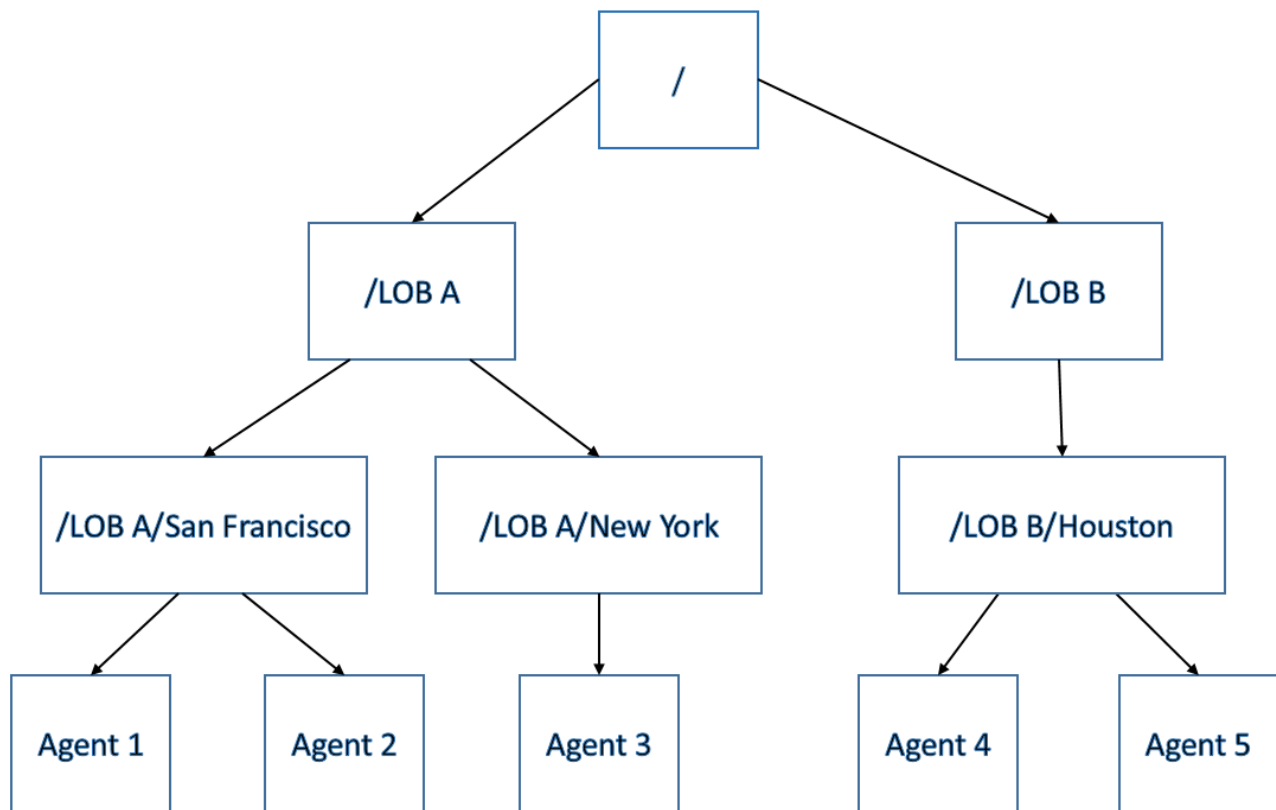
On the **Annex** of each **Person** object, in the **[recording]** section, set the **agent_hierarchy** option:

- Agent 1 to /Company A/Team 1
- Agent 2 to /Company A/Team 1
- Agent 3 to /Company A/Team 2
- Agent 4 to /Company B/Team 3
- Agent 5 to /Company B/Team 3

The following table describes the Access Group objects and available permissions:

Access Group Member	View Recordings of	Roles Assigned (Examples)
/	All agents	Super Administrators
/Company A	Agents 1, 2, 3	Quality agents for Company A
/Company B	Agents 4, 5	Quality agents for Company B
/Company A/Team 1	Agents 1, 2	Supervisor for Team 1
/Company A/Team 2	Agent 3	Supervisor for Team 2
/Company B/Team 3	Agents 4, 5	Supervisor for Team 3

Lines of Business



Another way to organize agent hierarchy can be based on different lines of business (LOB) in the organization and different geographical regions within the LOB.

In this scenario, you would configure your agent hierarchy as follows:

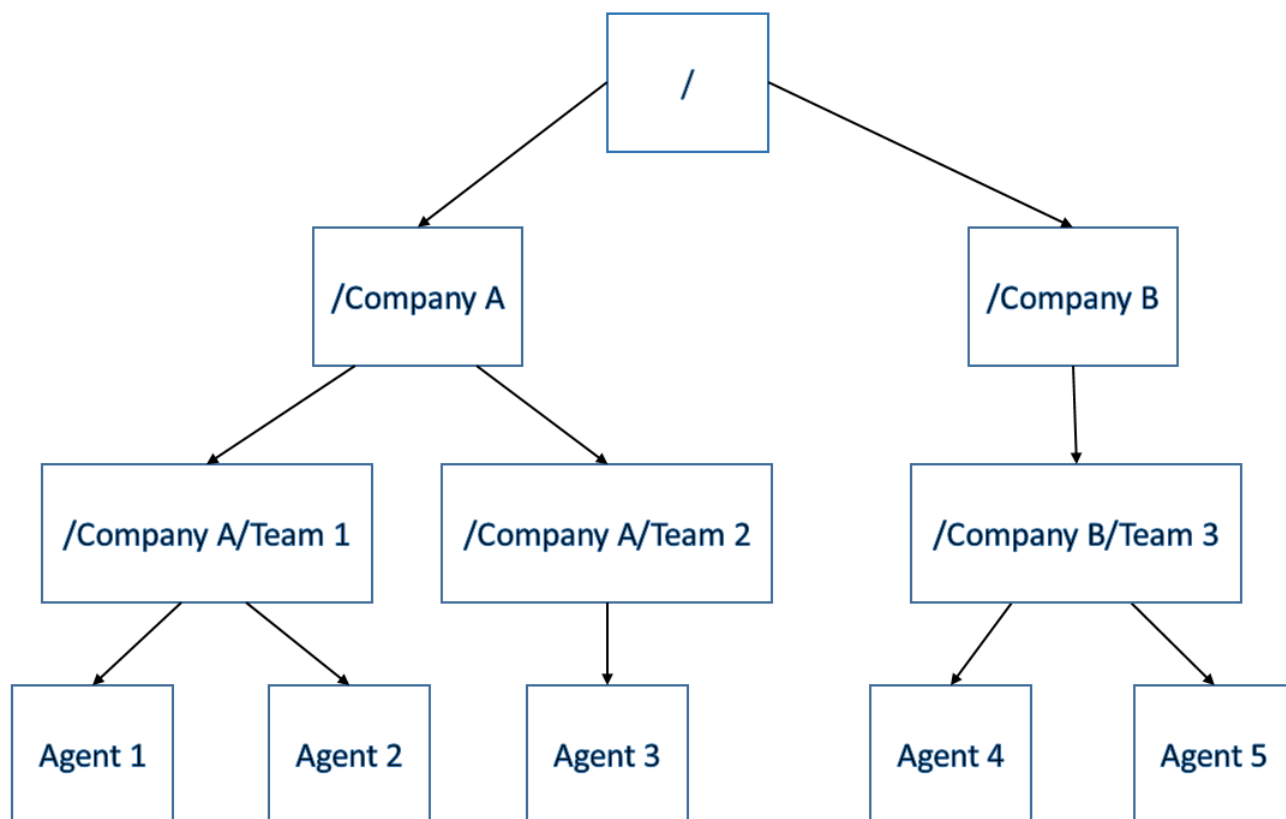
On the **Annex** of each **Person** object, in the **[recording]** section, set the **agent_hierarchy** option:

- Agent 1 to /LOB A/San Francisco
- Agent 2 to /LOB A/San Francisco
- Agent 3 to /LOB A/New York
- Agent 4 to /LOB B/Houston
- Agent 5 to /LOB B/Houston

The following table describes the Access Group objects and available permissions:

Access Group Member	View Recordings of	Roles Assigned (Examples)
/	All agents	Super Administrators
/LOB A	Agents 1, 2, 3	Quality agents for LOB A
/LOB B	Agents 4, 5	Quality agents for LOB B
/LOB A/San Francisco	Agents 1, 2	Manager in San Francisco
/LOB A/New York	Agent 3	Manager in New York
/LOB B/Houston	Agents 4, 5	Manager in Houston

Lines of Business with Multiple Companies Enforced Separately



This example describes a combination of both LOB and multiple third-party companies. Each company handles calls from different LOBs, and each call is assigned to a single line of business. Since there are agents handling calls for different LOBs, you need use **partitioning** to assign the LOB for each call from the routing strategy. Use the attached data key GRECORD_PARTITIONS to set the LOBs in the routing strategy.

In this scenario, you would configure your agent hierarchy as follows:

On the **Annex** of each **Person** object, in the **[recording]** section, set the **agent_hierarchy** option:

- Agent 1 to /Company A/Team 1
- Agent 2 to /Company A/Team 1
- Agent 3 to /Company A/Team 2
- Agent 4 to /Company B/Team 3
- Agent 5 to /Company B/Team 3

The following table describes the Access Group objects and available permissions:

Access Group Member	View Recordings of	Roles Assigned (Examples)
/	All agents	Super Administrators
/Company A	Agents 1, 2, 3	Quality agents for Company A
/Company B	Agents 4, 5	Quality agents for Company B
/Company A/Team 1	Agents 1, 2	Supervisor for Team 1
/Company A/Team 2	Agent 3	Supervisor for Team 2
/Company B/Team 3	Agents 4, 5	Supervisor for Team 3
/Line_of_BusinessA	All calls with attached data GRECORD_PARTITIONS = /Line_of_BusinessA	Quality agents for LOB A
/Line_of_BusinessB	All calls with attached data GRECORD_PARTITIONS = /Line_of_BusinessB	Quality agents for LOB B

Partitioning

As mentioned in the above examples, partitions for a call are set with the attached data key GRECORD_PARTITIONS. The value can be set as a comma-delimited list of partitions when you want to set more than one partition for the call. For example, GRECORD_PARTITIONS=/Line_of_BusinessA,/Line_of_BusinessB means that the call will be assigned both lines of businesses.

The attached data key can be set at anytime during the call, and the last effective value in the call is taken as the value for determining the partitions for the call.

Combining agent hierarchy with partitions means that a recording can be viewed by multiple groups of users based on business needs. A user belonging to any one of the Access Groups for the recording segment may access the recording segment.

For the **Lines of Business with Multiple Companies Enforced Separately** example, if a call is handled by Agent 3 and the attached data GRECORD_PARTITIONS=/Line_of_BusinessA,/Line_of_BusinessB, the following groups of users can access this call:

Access Group Member	View Recordings of	Roles Assigned (Examples)
/	All agents	Super Administrators
/Company A	Agents 1, 2, 3	Quality agents for Company A
/Company A/Team 2	Agent 3	Supervisor for Team 2
/Line_of_BusinessA	All calls with attached data GRECORD_PARTITIONS = /Line_of_BusinessA	Quality agents for LOB A
/Line_of_BusinessB	All calls with attached data GRECORD_PARTITIONS = /Line_of_BusinessB	Quality agents for LOB B

Recording Segments and Transfers

When a call is transferred from one agent to another and recording is enabled, the recorded interaction will contain two recording segments. Genesys Interaction Recording enforces access control for each recording segment separately. For each recording segment, agent hierarchy is assigned separately based on the agent for that segment. Partitions are assigned to all segments equally.

Extending the [Lines of Business with Multiple Companies Enforced Separately](#) example above, an inbound call is handled by Agent 1 and later transferred to Agent 3, with attached data GRECORD_PARTITIONS=/Line_of_BusinessA. The following access control applies:

Recording Segment	Agent Handling the Call	Access		
		Access Group Member	View Recording of	Roles Assigned (Examples)
1	Agent 1	/	All agents	Super Administrators
		/Company A	Agents 1, 2, 3	Quality agents for Company A
		/Company A/Team 1	Agent 1, 2	Supervisor for Team 1
		/Line_of_BusinessA	GRECORD_PARTITIONS = /Line_of_BusinessA	Quality agents for LOB A

Recording Segment	Agent Handling the Call	Access		
2	Agent 2	Access Group Member	View Recordings of	Roles Assigned (Examples)
		/	All agents	Super Administrators
		/Company A	Agents 1, 2, 3	Quality agents for Company A
		/Company A/Team 2	Agent 3	Supervisor for Team 2
		/Line_of_BusinessA	GRECORD_PARTITIONS = /Line_of_BusinessA	Quality agents for LOB A

Not Using Agent Hierarchy

If you do not use agent hierarchy, or if the agent hierarchy is not set for the person object, the recording can be accessed by a SpeechMiner user with Access Group "/" only. You can use partitions set independently for the recordings. For example, if agent hierarchy is not set, but the call contains attached data GRECORD_PARTITIONS = /Line_of_BusinessA, the following access control applies:

Access Group Member	View Recordings of	Roles Assigned (Examples)
/	All agents	Super Administrators
/Line_of_BusinessA	All calls with attached data GRECORD_PARTITIONS = /Line_of_BusinessA	Quality agents for LOB A

Secure Transport Configuration

This section describes how to configure Transport Layer Security (TLS) for the Genesys Interaction Recording solution.

Server-Side Configuration

The following components must configure secure transports for HTTP.

Interaction Recording Web Services

Configuring TLS for Interaction Recording Web Services

See [Configuring TLS on the Server Side for Interaction Recording Web Services](#).

Configuring TLS for the Recording Processor Script

1. Configure HTTPS on the primary recording server. For more information, see the "Configure SSL" section of [Configuring Recording Processor Script](#).
 - a. For Windows, make sure the pyOpenSSL is installed. pyOpenSSL is already be installed on RHEL6.
 - b. Create a self-signed certificate and private key for the Recording Processor host. For example, on Ubuntu run:

```
openssl req -new -x509 -days 1024 -nodes -out cert228.pem -keyout cert228key.pem
```
 - c. In the `rp_server` section of the Recording Processor's configuration file, set the following parameters:
 - `ssl_certificate`—Point to the certificate PEM file. For example, `ssl_certificate=cert228.pem`.
 - `ssl_private_key`—To point to the private key file. For example, `cert228.pem`.
 - d. Send the self-signed certificate PEM file to any MCP client that needs to validate the certificate during the SSL handshake. See the "Enable Secure Communication" section of the [GVP 8.5 User's Guide](#).
 - e. Restart Recording Processor.
2. Configure HTTPS on the backup recording server by following the same instructions as above using a new certificate and private key.

Configuring TLS for the Voice Processor

See [Voice Processor Service Level Configuration](#).

Configuring TLS for the Recording Crypto Server

See [Configure HTTP Port](#) tab in the [Configuring Recording Crypto Server](#) section.

Configuring TLS for the WebDAV Server

See [Configuring TLS for the WebDAV Server](#).

Configuring TLS for the Interaction Receiver and SpeechMiner UI Server

See [Enabling HTTPS for SpeechMiner](#).

Configuring TLS for the HTTP Load Balancer

See [Configuring TLS for the HTTP Load Balancer](#) in a single-tenant environment.
See [Configuring TLS for the HTTP Load Balancer](#) in a multi-tenant environment.

Client-Side Configuration

Configuring TLS for the Media Control Platform (MCP)

To add a Certificate Authority (CA):

1. Place the CA file on the MCP.
2. Using Genesys Administrator or Genesys Administrator Extension, in the **[fm]** section set the **ssl_ca_info** option to the location of the CA file.
3. Restart MCP.

To add client-side authentication:

1. Place the certificate file (PEM format) on the MCP.
2. Using Genesys Administrator or Genesys Administrator Extension, in the **[fm]** section set the **ssl_cert** option to the location of the certification file.
3. Restart MCP.

For more information about the MCP options, see the [Voice Platform Media Control Platform Configuration Options](#).

Configuring TLS for the IVR Profile

Using Genesys Administrator Extension, navigate to the Recording tab of the IVR Profile. Update the following addresses with the HTTPS locations:

- Storage Destination
- Recording Processor URI

- SpeechMiner Interaction Receiver
- SpeechMiner Destination for Analytics only

Configuring TLS for the Recording Processor Script

The Recording Processor Script creates three client connections, to:

- Interaction Recording Web Services (Web Services)
- SpeechMiner Interaction Receiver
- Backup Recording Processor Script

For details on configuring each connection, refer to the appropriate section at the [Configure SSL](#) link on the page [Deploying Recording Processor Script](#).

Configuring TLS for the Voice Processor

The Voice Processor creates three client connections, to:

- Interaction Recording Web Services (Web Services)
- SpeechMiner Interaction Receiver
- Genesys Info Mart

For details on configuring these connections, see [Configuring Voice Processor](#).

Configuring TLS for Interaction Recording Web Services

Interaction Recording Web Services (RWS) may be configured to use secure connections to the following components:

- Configuration Server
- SIP Server
- Interaction Server
- WebDAV
- Recording Crypto Server
- SpeechMiner Interaction Receiver
- Cassandra

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Security](#).

Configuring TLS for the Recording Muxer Script

The Recording Muxer Script creates client connections to the following:

- Interaction Recording Web Services
- Recording Crypto Server (if the recordings are encrypted)
- WebDAV

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Transport Layer Security \(TLS\) Connections](#).

Configuring TLS for the Recording Crypto Server

The Recording Crypto Server creates client connections to the following:

- Interaction Recording Web Services
- SpeechMiner Interaction Receiver
- Message Server
- Configuration Server

For details on configuring each connection using TLS, refer to the appropriate section in [Configuring Transport Layer Security \(TLS\) Connections](#).

Configuring TLS for the Recording Plug-in for GAX

See [Configuring Transport Layer Security](#).

Automated Recovery of Recordings

Important

The Lost Voice Recording (LVR) Recovery Script is not required when using the Voice Processor instead of the Recording Processor Script (RPS).

Starting with GIR release 8.5.216.01, the GIR solution provides automated recovery of recordings that have not been successfully posted for various reasons. The Lost Voice Recording (LVR) Recovery Script automatically goes through these recordings and recovers them. This means completing the posting process of the recordings: posting metadata to Recording Processor Script (RPS) and recordings to WebDAV (in the case of a Media Control Platform machine), and/or posting metadata to Interaction Recording Web Services (RWS) and SpeechMiner Interaction Receiver (in the case of a Recording Processor Script machine).

Some of the reasons that recordings processed by Media Control Platform (MCP) and RPS fail to be posted include:

- The recording storage is not available and MCP cancels the upload process.
- The RPS is not available and MCP does not send metadata to RPS.
- The RPS URL and credentials were incorrectly entered in the IVR Profile. As a result, MCP cannot send metadata to RPS.
- Storage credentials were incorrectly entered in the IVR Profile, causing MCP to fail to upload to WebDAV.
- SpeechMiner Interaction Receiver credentials were incorrectly configured in the IVR Profile.

Recoverable and Unrecoverable Recordings

The LVR Recovery Script categorizes each recording as recoverable or unrecoverable. A recording is considered unrecoverable if any of the following is true:

- The metadata cannot be parsed.
- The LVR Recovery Script cannot find the information it needs to recover the recording within the metadata.
- If recovering for MCP, if a recording is missing metadata or a media file.
- If recovering for MCP and the media file is encrypted, the encryption key is missing.

The LVR Recovery Script attempts to upload unrecoverable recordings to the unrecoverable storage specified in the properties file. Genesys recommends that you monitor the unrecoverable storage location and investigate these recordings to determine what caused the problem.

The LVR Recovery Script attempts to recover recoverable recordings using their IVR Profile and metadata.

Important

Automated recovery of Genesys Interaction Analytics (GIA) recordings is currently not supported.

Installing LVR Recovery Script

The LVR Recovery Script must be installed on each machine where MCP and/or RPS run. If installing the LVR Recovery Script for both MCP and RPS components (where both components are installed on the same machine), select both components during installation. In the LVR Recovery Script installation directory, there are two configuration files—for recovering recordings for MCP and for recovering recordings for RPS.

Installing on Linux

To install the LVR Recovery Script on Linux, complete the following steps:

1. Perform one of the following:
 - For LVR version 8.5.222.58 (or higher), install Java 17 . You can use the OpenJDK version of the software.
 - For LVR version 8.5.222.55 (or lower), install Java 8. You can use the Oracle or OpenJDK version of the software.

Note: Genesys recommends using the latest supported version of Java and deprecating any previous versions. See the [Genesys Interaction Recording](#) page in the *Genesys Supported Operating Environment Reference* for more details about supported versions.
2. In the directory to which the LVR Recovery Script installation package was copied, locate a shell script called **install.sh**.
3. Run this script from the command prompt by typing **sh** and the file name. For example: **sh install.sh**.
4. When prompted, schedule when you would prefer the LVR Recovery Script to run throughout the day, by entering a comma-separated line of text containing all the times of the day in 24-hour format. For example: 00:30,12:30,01:25. This will schedule the LVR Recovery Script to run every day at 12:30 AM, 12:30 PM, and 1:25 AM. Use a leading zero for a single-digit hour (01:25 instead of 1:25).
5. [Configure the LVR Recovery Script](#).

Installing on Windows

To install the LVR Recovery Script on Windows, complete the following steps:

1. Perform one of the following:

- For LVR version 8.5.222.58 (or higher), install Java 17 . You can use the OpenJDK version of the software.
- For LVR version 8.5.222.55 (or lower), install Java 8. You can use the Oracle or OpenJDK version of the software.

Note: Genesys recommends using the latest supported version of Java and deprecating any previous versions. See the [Genesys Interaction Recording](#) page in the [Genesys Supported Operating Environment Reference](#) for more details about supported versions.

2. In the directory to which the LVR Recovery Script installation package was copied, locate and double-click **Setup.exe** to start the installation.
3. When prompted, schedule when you would prefer the LVR Recovery Script to run throughout the day, by entering a comma-separated line of text containing all the times of the day in 24-hour format. For example: 00:30,12:30,01:25. This will schedule the LVR Recovery Script to run every day at 12:30 AM, 12:30 PM, and 1:25 AM. Use a leading zero for a single-digit hour (01:25 instead of 1:25).
4. [Configure the LVR Recovery Script.](#)

Upgrading LVR Recovery Script

To upgrade the LVR Recovery Script, complete the following steps:

1. Back up the **MCP_premise.properties** and **RPS_premise.properties** files.
2. Uninstall the existing version of the LVR Recovery Script.
3. Install the new version of the LVR Recovery Script.
4. Replace the generated **MCP_premise.properties** and **RPS_premise.properties** files with the backup files.

Rescheduling When LVR Recovery Script Runs

To reschedule when the LVR Recovery Script runs each day, reinstall the LVR Recovery Script or manually edit the system's scheduled tasks.

Reinstalling LVR Recovery Script

To reinstall the LVR Recovery Script:

1. Back up the **MCP_premise.properties** and/or **RPS_premise.properties** files by moving them to another directory.
 2. Uninstall the LVR Recovery Script.
 3. Reinstall the LVR Recovery Script and enter the new scheduling of the LVR Recovery Script.
 4. Replace the generated **MCP_premise.properties** and/or **RPS_premise.properties** files with the backup files.
-

Editing scheduled tasks on Linux

To edit scheduled tasks on Linux:

1. Edit the crontab by running the command: `crontab -e`. Each line in the crontab has the following format:

```
minute    hour    day    month    dayofweek    command
```

The first five numbers are when the command will be run. An asterisk (*) means the script is to be run at every instance (every hour, every weekday, and so on) within the time period. The scheduled runs of the LVR Recovery Script are the lines in the crontab that have the following command:

```
java -jar <LVR_install_directory>/recover_LVRs.jar --mode recover --component <component> --properties <LVR_install_directory>/<component>_premise.properties
```

2. Change the schedule of those lines by editing one or any of the following: minute, hour, day, month, or dayofweek.
3. Delete a scheduled run by deleting its corresponding line.

For more information about editing the crontab, refer to Linux documentation.

Editing scheduled tasks on Windows

To edit scheduled tasks on Windows:

1. Open the Task Scheduler.
2. Click the Task Scheduler Library on the left. You should see all scheduled tasks.
3. The LVR Recovery Script schedules its runs with the name **IntRcLVRRSPrem64_MCP** for MCP recoveries and **IntRcLVRRSPrem64_RPS** for RPS recoveries.
4. To edit the scheduled task, right-click the task and click **properties**. Go to the **triggers** tab to edit when the task should run.
5. To delete the scheduled task, right-click the task and click **delete**.

Configuring LVR Recovery Script

Configure the LVR Recovery Script in its installation directory. If the LVR Recovery Script is to be used for both MCP and RPS on the same machine, the two different properties files must both be configured.

For MCP, you must configure the following:

- [WebDAV storage connection](#)
- [Configuration Server connection](#)

For RPS, you must configure the following:

- [RWS connection](#)

The following connections can be configured to use TLS:

- [Configuration Server](#)
- [Recoverable WebDAV for a Tenant](#)
- [Unrecoverable WebDAV for a Tenant](#)
- [Recording Processor Script](#)
- [Interaction Recording Web Services](#)
- [SpeechMiner Interaction Receiver](#)

Configuring WebDAV Storage Connection for MCP

You must configure WebDAV if you are using MCP.

Recoverable recordings are uploaded to a WebDAV URL (specified in the recording metadata) using the IVR Profile specified in the configuration environment. No configuration is needed except for ensuring the LVR Recovery Script can communicate with Configuration Server to retrieve IVR Profiles and that the WebDAV server is running.

For unrecoverable recordings, the WebDAV URLs and credentials for each tenant must be specified in the **MCP_premise.properties** file. To specify a tenant's WebDAV information for unrecoverable recordings:

1. Open **MCP_premise.properties**.
2. Add the tenant's name to **lvrrcovery.webDAV.tenants**. Each additional tenant is separated by a comma. The tenant names specified in this file must match the tenant names in the metadata of the corresponding unrecoverable recording.
Example: `lvrrcovery.webDAV.tenants=tenant1,tenant2,tenant3`
3. Specify the tenant's WebDAV URL under `lvrrcovery.webDAV.<tenantname>.unrecoverable.url`. The directory must already exist or be manually created before running the LVR Recovery Script.

Important

If the tenant's WebDAV URL does not point to a directory in WebDAV, the LVR Recovery Script does not create the directory and, as a result, the unrecoverable recordings will not be uploaded.

4. Specify the tenant's WebDAV username under
-

`lvrrecovery.webDAV.<tenantname>.unrecoverable.username.`

5. Specify the tenant's WebDAV password under
`lvrrecovery.webDAV.<tenantname>.unrecoverable.password.`
6. Create a tenant with the name UNKNOWN. It will be used for uploading unrecoverable recordings when the tenant information cannot be recovered.

If a tenant's unrecoverable information is not specified, the LVR Recovery Script does not attempt to upload the unrecoverable recording and the unrecoverable recording will remain in the folder for failed recordings. The recoverable recordings for that tenant will be recovered normally.

Use the following optional properties to configure the maximum number of tolerated upload failures during a recovery for a tenant to WebDAV, as follows:

- **`lvrrecovery.webDAV.<tenantname>.unrecoverable.maxToleratedwebDAVFailures`**—specifies the maximum tolerated upload failures for the WebDAV server specified in **`lvrrecovery.webDAV.<tenantname>.unrecoverable`**. If the number of failures exceeds the threshold for a tenant, the LVR Recovery Script stops uploading unrecoverable recordings for this tenant during this run and will try again in the next run.
- **`lvrrecovery.webDAV.<tenantname>.recoverable.maxToleratedwebDAVFailures`**—specifies the maximum number of tolerated upload failures for the recoverable WebDAV server. The recoverable WebDAV server is the WebDAV server specified in a tenant's IVR profile and a recording's metadata. If the number of failures for a particular WebDAV server exceeds the threshold for a tenant, the LVR Recovery Script stops uploading recoverable recordings for this tenant to the specific WebDAV server during this run and will try again in the next run.

A tenant must be specified in **`lvrrecovery.webDAV.tenants`** for any related tenant properties for that tenant to work.

Secure connections to WebDAV storage are supported over HTTPS.

Configuring Configuration Server Connection for MCP

These mandatory properties must be configured to run the LVR Recovery Script for MCP:

- **`mcp.configserver.host`**—The hostname for Configuration Server
- **`mcp.configserver.port`**—The port for Configuration Server
- **`mcp.configserver.appname`**— The application name for the Configuration Manager Object (by default, the value is **default**).
- **`mcp.configserver.username`**—The username to be used for Configuration Server
- **`mcp.configserver.password`**—The password to be used for Configuration Server

Configuring RWS Connection for RPS

If using the LVR Recovery Script for RPS, complete the following steps:

1. In the **RPS_premise.properties** file, set the mandatory **rp.htccuri** property to the RWS base URI. For example: `http://vagrant.genesys.com:8081`
2. Specify ops credentials for RWS by setting the **rp.opsUser** and **rp.opsPassword** properties in the **RPS_premise.properties** file.
3. (Optional) Configure the default Contact Center ID to be used when it cannot be determined from the metadata by setting the **rp.defaultccid** property in the **RPS_premise.properties** file.

Secure connections to RWS and SpeechMiner Interaction Receiver are supported over HTTPS.

Configuring Transport Layer Security (TLS) Connections (Optional)

The following sections explain how to configure TLS connections.

Configuring a TLS Connection to Configuration Server

1. Ensure that the Configuration Server port is properly configured as an auto-detect port.
2. In the **MCP_premise.properties** file, set the **mcp.configserver.port** property to the Configuration Server auto-detect port.
3. If required, modify the TLS version when connecting to Configuration Server using the **mcp.configserver.defaultTlsVersion** property in the properties file. Supported TLS versions are:
 - TLSv1.1—TLS version 1.1 (the default)
 - TLSv1.2—TLS version 1.2

Configuring a TLS Connection to Recoverable WebDAV for a Tenant

In the **MCP_premise.properties** file, configure the **ivrrecovery.webDAV.<tenantname>.recoverable.trustedCA** property for that tenant's recoverable WebDAV connection as follows:

- If the TLS certificate was issued by a well-known certificate authority such as VeriSign, set the **ivrrecovery.webDAV.<tenantname>.recoverable.trustedCA** parameter to `true`.
- If the TLS certificate was issued by your own certificate authority, set the **ivrrecovery.webDAV.<tenantname>.recoverable.trustedCA** parameter to the path to a file containing the certificate of the CA that generated the certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then use the certificate chain containing the intermediate CAs and root CA in PEM format.

- If the TLS certificate is a self-signed certificate, set the **ivrrecovery.webDAV.<tenantname>.recoverable.trustedCA** parameter to the path to the self-

signed certificate. The file containing the certificate must be in PEM format.

- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set the **ivrrecovery.webDAV.<tenantname>.recoverable.trustedCA** parameter to false. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 7 versions and Java Runtime Environment 8 disallow MD5 signatures for certificates.

Important

The statement about JRE 7/8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring a TLS Connection to Unrecoverable WebDAV for a Tenant

1. Edit the following files as appropriate:
 - **MCP_premise.properties**—if LVR Recovery Script is recovering recordings for MCP.
 - **RPS_premise.properties**—if LVR Recovery Script is recovering recordings for RPS.
2. Set the **ivrrecovery.webDAV.<tenantname>.unrecoverable.url** property in properties file to use the unrecoverable WebDAV https URL.
3. In the properties file, configure the **ivrrecovery.webDAV.<tenantname>.unrecoverable.trustedCA** property for that tenant's unrecoverable WebDAV connection as follows:
 - If the TLS certificate was issued by a well-known certificate authority such as Verisign, set the **ivrrecovery.webDAV.<tenantname>.unrecoverable.trustedCA** parameter to true.
 - If the TLS certificate was issued by your own certificate authority, set the **ivrrecovery.webDAV.<tenantname>.unrecoverable.trustedCA** parameter to the path to a file containing the certificate of the CA that generated the certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then use the certificate chain containing the intermediate CAs and root CA in PEM format.

- If the TLS certificate is a self-signed certificate, set the **ivrrecovery.webDAV.<tenantname>.unrecoverable.trustedCA** parameter to the path to the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set the **ivrrecovery.webDAV.<tenantname>.unrecoverable.trustedCA** parameter to false. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime

Environment 7 versions and Java Runtime Environment 8 disallow MD5 signatures for certificates.

Important

The statement about JRE 7/8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring a TLS Connection to Recording Processor Script

In the **MCP_premise.properties** file, configure the **mcp rpTrustedCA** property as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set the **mcp rpTrustedCA** parameter to `true`.
- If the TLS certificate was issued by your own certificate authority, set the **mcp rpTrustedCA** parameter to the path to a file containing the certificate of the CA that generated the certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then use the certificate chain containing the intermediate CAs and root CA in PEM format.

- If the TLS certificate is a self-signed certificate, set the **mcp rpTrustedCA** parameter to the path to the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **mcp rpTrustedCA** to `false`. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 7 versions and Java Runtime Environment 8 disallow MD5 signatures for certificates.

Important

The statement about JRE 7/8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring a TLS Connection to Interaction Recording Web Services

1. In the **RPS_premise.properties** file, set the **rp.htccuri** property to use the Interaction Recording Web Services https URL.

2. In the **RPS_premise.properties** file, configure the **rp.rwsTrustedCA** property as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set the **rp.rwsTrustedCA** parameter to `true`.
- If the TLS certificate was issued by your own certificate authority, set the **rp.rwsTrustedCA** parameter to the path to a file containing the certificate of the CA that generated the certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then use the certificate chain containing the intermediate CAs and root CA in PEM format.

- If the TLS certificate is a self-signed certificate, set the **rp.rwsTrustedCA** parameter to the path of the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **rp.rwsTrustedCA** to `false`. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 7 versions and Java Runtime Environment 8 disallow MD5 signatures for certificates.

Important

The statement about JRE 7/8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring a TLS Connection to SpeechMiner Interaction Receiver

In the **RPS_premise.properties** file, configure the **rp.speechminerTrustedCA** parameter as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set the **rp.speechminerTrustedCA** parameter to `true`.
- If the TLS certificate was issued by your own certificate authority, set the **rp.speechminerTrustedCA** parameter to the path to the CA that generated the certificate. The file containing the certificate must be in PEM format.

Important

If there are intermediate certificate authorities forming a chain of trust, then use the certificate chain containing the intermediate CAs and root CA in PEM format.

- If the TLS certificate is a self-signed certificate, set the **rp.speechminerTrustedCA** parameter to the path to the self-signed certificate. The file containing the certificate must be in PEM format.
- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set the **rp.speechminerTrustedCA** parameter to false. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 7 versions and Java Runtime Environment 8 disallow MD5 signatures for certificates.

Important

The statement about JRE 7/8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuration File Parameters

Parameter Name	Mandatory	Description	Example
http.connection.timeout	No	The HTTP connection timeout value. This property must be set in seconds. The default HTTP connection timeout value is 60 seconds.	10
http.socket.timeout	No	The HTTP socket timeout value. This property must be set in seconds. The default HTTP socket timeout value is 30 seconds.	10
lvrrecovery.failedfolder	Yes	The folder where failed recordings are stored. This property must be set to match the value in the failed_folder_path option in the [processing] section of the rpconfig.cfg file. If failed_folder_path is not set in the rpconfig.cfg file, the default value of <Installation Directory>/RP/failed is used.	<Installation Directory>/RP/failed
lvrrecovery.maxDirectoryRecurse	No	The maximum directory depth the LVR Recovery Script goes when recovering recordings. Default is a depth of 20.	10
lvrrecovery.timeout	No	The time period, in minutes, during which the LVR Recovery Script attempts to recover recordings. If the LVR Recovery Script does not	60

		finish the recovery within the specified timeout, it exits. If this property is not specified, the timeout is not used.	
lvrrecovery.startdate	No	The date for the LVR Recovery Script to ignore recovery of recordings older than this date. If the time period specified by this property and the mcp.minimumRecordingAge property overlaps, this property is ignored. The format of the parameter: YYYY/MM/DD.	2017/01/01
lvrrecovery.maxRetry	No	The number of times the LVR Recovery Script will retry recovering a failed recording if the initial recovery attempt fails. Default is 5 attempts.	10
lvrrecovery.log_dir	No	The directory in which the LVR Recovery Script writes its log files. By default, the log files are written in the installation directory.	<Installation Directory>/logs
mcp.configserver.host	Yes if recovering for MCP	The hostname for Configuration Server.	host.example.com
mcp.configserver.port	Yes if recovering for MCP	The port for Configuration Server.	8888
mcp.configserver.appname	Yes if recovering for MCP	The application name for Configuration Server.	default
mcp.configserver.username	Yes if recovering for MCP	The username for Configuration Server.	username
mcp.configserver.password	Yes if recovering for MCP	The password for Configuration Server.	password
mcp.configserver.defaultTLSVersion	No	The initial TLS version used to connect to Configuration Server over a secure port. Valid versions are TLSv1.1 (the default), TLSv1.2.	TLSv1.1
mcp.minimumRecordingAge	No	LVR Recovery Script ignores MCP recordings that are newer than this property. The time used is when the recording	30

		was written to a disk on this machine, not the start time on the metadata. This is to prevent files still being processed by MCP, to be attempted to be recovered. This property must be set in hours. Default is 24 hours.	
mcp.rpTrustedCA	No	Configures TLS certificate validation when making a secure outbound connection to Recording Processor Script (RPS). Valid values are true, false, or a path to a trusted certificate authority (CA) bundle. If set to true, the certificate is validated. If set to false, the certificate is not validated. The CA file must be in PEM format. LVR Recovery Script exits during initialization under the following conditions: the CA path does not exist, the CA file is not a valid PEM file, or the CA file is corrupted. This parameter is optional, and defaults to false.	true
rp.htccuri	Yes if recovering for RPS	The URI for the Interaction Recording Web Services node.	http://vagrant.genesys.com:8081
rp.opsUser	No	The admin username for Interaction Recording Web Services.	username
rp.opsPassword	No	The admin password for Interaction Recording Web Services.	password
rp.defaultccid	No	The default Contact Center ID (CCID) that the LVR Recovery Script uses when posting metadata if the CCID is unknown.	
rp.rwsTrustedCA	No	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are true, false, or a path to a trusted certificate authority (CA) bundle. If set to true, the certificate is validated. If set to false, the certificate is not validated. The CA file must be in PEM format. LVR Recovery Script exits during initialization under the	true

		following conditions: the CA path does not exist, the CA file is not a valid PEM file, or the CA file is corrupted. This parameter is optional, and defaults to false.	
rp.speechminerTrustedCA	No	Configures TLS certificate validation when making a secure outbound connection to SpeechMiner Interaction Receiver. Valid values are true, false, or a path to a trusted certificate authority (CA) bundle. If set to true, the certificate is validated. If set to false, the certificate is not validated. The CA file must be in PEM format. LVR Recovery Script exits during initialization under the following conditions: the CA path does not exist, the CA file is not a valid PEM file, or the CA file is corrupted. This parameter is optional, and defaults to false.	true
lvrrecovery.webDAV.tenants	No	A comma-separated list of tenants that the LVR Recovery Script processes.	UNKNOWN,tenant1,tenant2,tenant3
lvrrecovery.webDAV.<tenantname>.unrecoverable.url	No	The WebDAV URL to be used for unrecoverable recordings for <tenantname>. The tenant must be specified in lvrrecovery.webDAV.tenants .	
lvrrecovery.webDAV.<tenantname>.unrecoverable.username	No	The WebDAV username to be used for unrecoverable recordings for <tenantname>. The tenant must be specified in lvrrecovery.webDAV.tenants .	
lvrrecovery.webDAV.<tenantname>.unrecoverable.password	No	The WebDAV password to be used for unrecoverable recordings for <tenantname>. The tenant must be specified in lvrrecovery.webDAV.tenants .	
lvrrecovery.webDAV.<tenantname>.unrecoverable.maxToleratedWebDAVFailures	No	The maximum number of tolerated upload failures during a recovery for a tenant WebDAV. Default is 50. The tenant must be specified in lvrrecovery.webDAV.tenants .	25
lvrrecovery.webDAV.<tenantname>.recoverable.maxToleratedWebDAVFailures	No	The maximum number of tolerated download failures during a recovery for a tenant WebDAV. Default is 50. The tenant must be specified in lvrrecovery.webDAV.tenants .	25

		of tolerated upload failures during a recovery for a tenant to their recoverable WebDAV. The recoverable WebDAV is the WebDAV server specified in the tenant's IVR Profile and a recording's metadata. Default is 50. The tenant must be specified in lvrrecovery.webDAV.tenants .	
lvrrecovery.webDAV.<tenantname>.recoverable.trustedCA		Configures TLS certificate validation when making a secure outbound connection to a tenant's recoverable WebDAV. Valid values are true, false, or a path to a trusted certificate authority (CA) bundle. If set to true, the certificate is validated. If set to false, the certificate is not validated. The CA file must be in PEM format. LVR Recovery Script exits during initialization under the following conditions: the CA path does not exist, the CA file is not a valid PEM file, or the CA file is corrupted. This parameter is optional, and defaults to false. The tenant must be specified in lvrrecovery.webDAV.tenants .	true
lvrrecovery.webDAV.<tenantname>.unrecoverable.trustedCA		Configures TLS certificate validation when making a secure outbound connection to a tenant's unrecoverable WebDAV. Valid values are true, false, or a path to a trusted certificate authority (CA) bundle. If set to true, the certificate is validated. If set to false, the certificate is not validated. The CA file must be in PEM format. LVR Recovery Script exits during initialization under the following conditions: the CA path does not exist, the CA file is not a valid PEM file, or the CA file is corrupted. This parameter is optional, and defaults to false. The tenant must be specified in lvrrecovery.webDAV.tenants .	true

Running LVR Recovery Script Manually

The LVR Recovery Script automatically runs at the times scheduled during installation.

To manually run the LVR Recovery Script to recover recordings, run the following command from the <LVR Installation Directory>:

For MCP:

```
java -jar recover_LVRs.jar --mode recover --component MCP --properties MCP_premise.properties
```

For RPS:

```
java -jar recover_LVRs.jar --mode recover --component RP --properties RPS_premise.properties
```

OR

```
java -jar recover_LVRs.jar --mode recover --component RPS --properties RPS_premise.properties
```

Recovering Metadata for SpeechMiner

The Recording Processor Script (RPS) installation package provides an additional script to manually post recordings to SpeechMiner by querying metadata from Interaction Recording Web Services (RWS).

Starting with GIR release 8.5.216.01, GIR implements automated recovery of failed recordings by using the Lost Voice Recording (LVR) Recovery Script. See [Automated Recovery of Recordings](#) for details.

Important

If you are using the Voice Processor instead of the Recording Processor Script and there is an outage that prevents posting of recordings to SpeechMiner Interaction Receiver, the Voice Processor automatically retries these posts for up to 40 days. As a result, you do not have to manually recover recordings if you resolve the downstream outage within that period.

Prerequisites

Before you start, you must have the following prerequisites:

- Recording Processor Script installed
- Python 2.7 or higher installed
- Admin username and password with read and write permissions on the folder where the script resides

Re-posting the Recordings to SpeechMiner

After you have installed the RPS, you can find the **repost_to_sm_from_htcc.py** reposting script in the **<Installation Directory>/scripts/** directory.

From the command line on the machine where the RPS is installed and running, execute the following command:

```
C:\<Installation Directory>\scripts>python repost_to_sm_from_htcc.py -startTime 1437057696000  
-endTime 1437058696000 -htccUri http://<Web Services URI> -ccid  
57c0b771-b57c-4ea8-8655-7ef6d3c58ccc  
-region us-west-1
```

To return the details of the parameters used, run the following command: `C:\GCTI\gir_rps\scripts>python repost_to_sm_from_htcc.py -h`

This command will return the following results:

```
usage: repost_to_sm_from_htcc.py [-h] -startTime STARTTIME -endTime ENDTIME
                                -region REGION -ccid CCID -htccUri HTCCURI
                                [-sleepMs SLEEPMS] [-pageLimit PAGELIMIT]
                                [-rp.speechminer.uri SPEECHMINERURI]
                                [-disableFailedFolder]

Recover RP failed folder
optional arguments:
  -h, --help                show this help message and exit
  -startTime STARTTIME      lower bound of range to recover
  -endTime ENDTIME          upper bound of range to recover
  -region REGION            range to recover, * for all
  -ccid CCID                contact center ID to recover
  -htccUri HTCCURI          htcc prefix. e.g. http://gws-elb.genesyscloud.com
  -sleepMs SLEEPMS          milliseconds sleep after each recording is processed. default to 0
  -pageLimit PAGELIMIT      page limit of script query from HTCC on each attempt. default to 10
  -rp.speechminer.uri SPEECHMINERURI
                            speechMiner IR Uri, if provided, it will override
                            values in metadata
  -disableFailedFolder
```

You can find the parameter descriptions of the parameters [below](#).

Examples

If only one region is recovered, the following command will re-post the recordings to SpeechMiner:

```
C:\<Installation Directory>\scripts>python repost_to_sm_from_htcc.py -startTime 1437057696000
-endTime 1437058696000
-htccUri http://<Web Services URI> -ccid 57c0b771-b57c-4ea8-8655-7ef6d3c58ccc -region us-
west-1
Please enter htcc ops username: ops
Please enter htcc ops password: ops
...
```

If there is more than one region recovered, the following command will re-post the recordings to SpeechMiner: Recover range of recording from all regions

```
C:\<Installation Directory>\scripts>python repost_to_sm_from_htcc.py -startTime 1437057696000
-endTime 1437058696000
-htccUri http://<Web Services URI> -ccid 57c0b771-b57c-4ea8-8655-7ef6d3c58ccc -region *
Please enter htcc ops username: ops
Please enter htcc ops password: ops
...
```

If the failed recording is caused by an invalid SpeechMiner Interaction Receiver, the metadata will contain the wrong **rp.speechminer.uri** or **rp.speechminer.auth**. The following command will re-post the recording to SpeechMiner with the correct SpeechMiner URI:

```
C:\<Installation Directory>\scripts>python repost_to_sm_from_htcc.py -startTime 1437057696000
-endTime 1437058696000
-htccUri http://<Web Services URI> -ccid 57c0b771-b57c-4ea8-8655-7ef6d3c58ccc -region us-
west-1 -rp.speechminer.uri http://<SpeechMiner IP Address>/interactionreceiver
Please enter htcc ops username: ops
Please enter htcc ops password: ops
Please enter rp.speechminer.auth (e.g. rp_user:123455): rpUser:123456
...
```

Command Line Parameters

The following table describes the command line parameters:

Parameter Name	Mandatory	Accept Prompt Input	Description	Example
startTime	Y	No	The time, in UTC milliseconds format, of the earliest recording to be processed.	1437057696000
endTime	Y	No	The time, in UTC milliseconds format, of the latest recording to be processed.	1437057696000
htccUri	N	No	The host and port of the Interaction Recording Web Services node (or Web Services node if you're using version 8.5.210.02 or earlier). If this parameter is not defined, the recordings are not rePOSTed to Interaction Recording Web Services (Web Services).	http://<Interaction Recording Web Services Server>:<Interaction Recording Web Services Port>
region	N	No	The location of the recording.	
ccid	N	No	The Contact Center ID.	
rp.speechminer.uri	N	Yes	The URI that points to the SpeechMiner Interaction Receiver responsible for accepting metadata from the Recording Processor script. If this parameter is not provided, or if it is an invalid URI, the script will prompt to override the value.	http://<SpeechMiner Host>/interactionreceiver

Parameter Name	Mandatory	Accept Prompt Input	Description	Example
rp.speechminer.auth	N	Yes	The credentials required to connect to the SpeechMiner Interaction Receiver used by the Recording Processor Script. The format is username:password, where the username and password are the Interaction Receiver credentials.	<SpeechMiner Username>:<SpeechMiner Password>
sleepsMs	N	No	The time, in milliseconds, to wait after processing each recording. If this parameter is not provided, the value defaults to 0.	30
pageLimit	N	No	The number of pages to query on each process attempt.	10
disableFailedFolder	N	No	Flag to control if the script should write the metadata of a recording file that cannot be rePOSTed to SpeechMiner. By default, the script caches them into the RPS failed folder. If this parameter is provided, the script only prints the error in the recovery logs.	

Logging

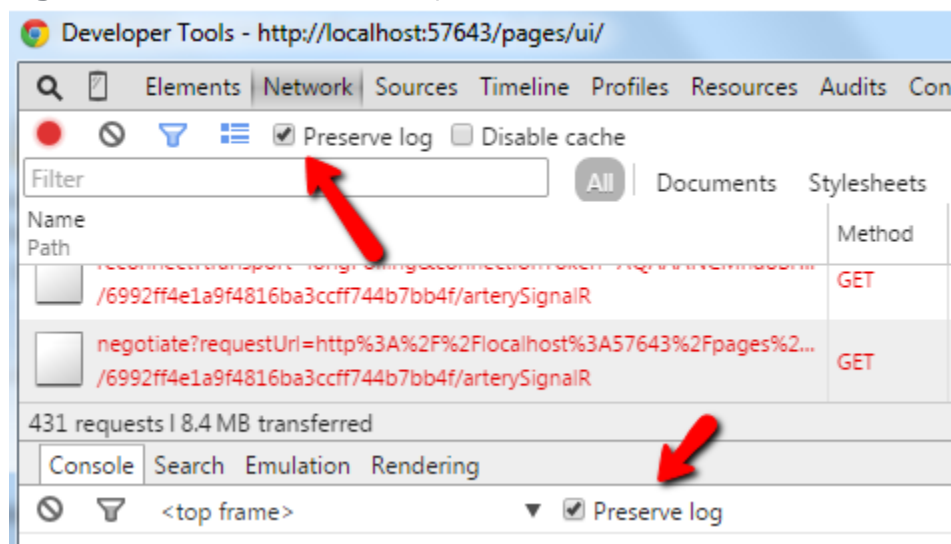
The **repost_to_sm_from_htcc.py** script writes audit logs in the **repost_to_sm_from_htcc_<execution-time>/results.log** file. If the **disableFailedFolder** parameter is not provided, and if there is an exception during recovery, the script will dump the

recording metadata into **repost_to_sm_from_htcc_<execution-time>/<ccid>** file.

Troubleshooting

How do I collect logs to debug issues relating to login to the SpeechMiner UI?

If you are having a problem logging into the SpeechMiner UI, you will need to collect the Network and Console log from the Chrome Developer Tools before and after the login attempts with the **Preserve log** checkbox enabled—for example:



Make sure that you have captured the requests for the following applications after you click the login button:

- Web Services
- Recording Crypto Server
- SpeechMiner

Important

Uncheck these options after you have collected the necessary information.

Why do only some of my calls have screen recordings associated

with them?

Screen Recording only occurs when an agent logs into a workstation that has Screen Recording Service installed.

To solve this issue try one of the following:

- If the Screen Recording Service is installed on your workstation and this problem persists, execute the following command to check the screen recording trigger condition settings for a given agent (for example, julie@genesys.com):

```
curl -u julie@genesys.com:123456 http://10.10.15.59/api/v2/me/settings/screen-recording-client
```

If the code below appears with Value = off, Screen Recording is disabled. If Value = random_voice(x) (where x is 0-100, representing the percentage of calls to record) then, the calls that do not fall into the selected percentage will not be recorded. The SRS is instructed to either record or not-record on a per-call basis. If you are not seeing screen recordings for your deployment, or, for a particular agent, you may wish to temporarily disable 'random_voice' or set to 100 to ensure that all interactions are instructed to have the screens recorded.

```
"statusCode": 0,
"settings": [
{
"name": "recordingWhen",
"value": <to-be-checked>
} ]
```

To reconfigure Screen Recording, refer to [Advanced Configuration for the Screen Recording Service](#).

- Go to C:\Genesys\SRC\Problem Videos. If the Problem Videos folder contains files, verify that you have the following information and contact your Genesys Professional:
 - GSR.log
 - config.json
 - The agent's name.
 - Directory listing

How do I restore the Find Recordings capability?

Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) uses Elasticsearch for metadata indexing, which is used by MLM, RCS archiving and the **SpeechMiner Screen Recording** grid.

If your **Elasticsearch** index is out of date, follow these steps to rebuild the index and restore the **Find Recordings** capability:

1. Select one **Interaction Recording Web Services (Web Services)** node, and do one of the following:
 - If you have a regular maintenance window, select an existing **Interaction Recording Web Services (Web Services)** node and proceed to step #2.

- If your business runs 24*7, prepare a new dedicated **Interaction Recording Web Services (Web Services)** node as follows:
 - a. Install Interaction Recording Web Services (Web Services) in the same way a regular **Interaction Recording Web Services (Web Services)** node is installed. Do not add this node to the Interaction Recording Web Services (Web Services) Load Balancer.
 - b. Edit the Interaction Recording Web Services application.yaml file (if you are using Web Services and Application version 8.5.201.09 or earlier modify the server-settings.yaml file instead), by adding the following configuration. Verify that you add lines under nodes for all the existing **Interaction Recording Web Services (Web Services)** nodes in your deployment:

```

elasticsearchSettings:
  useTransportClient: true
  transportClient:
    nodes:
      - {host: <elastic-search-node1>, port: 9300}
      - {host: <elastic-search-node2>, port: 9300}
      - {host: <elastic-search-node3>, port: 9300}
  useSniff: true
  ignoreClusterName: true
  pingTimeout: 10000
  nodesSamplerInterval: 10000
  enableIndexVerificationAtStartup: false
  indexPerContactCenter: true

```

Important

By default, the **Interaction Recording Web Services (Web Services)** node will also serve as the **Elasticsearch** node. This step is performed to avoid any change to the existing Elasticsearch Cluster.

2. Increase the Hystrix timeout for **RecordingOperationApiTaskV2** by adding the following line to the Hystrix configuration:

```

hystrix.command.RecordingOperationApiTaskV2.execution.isolation.thread.timeoutInMilliseconds=<max
time acceptable in milliseconds>

```

Important

<max time acceptable in milliseconds> should be 3 times the restored period or more.

3. Restart the selected Interaction Recording Web Services (Web Services) node, navigate to the following URL and wait until it returns the Interaction Recording Web Services (Web Services) version:

```

http://<selected-web-services-node>:<web-services-listening-port>/api/v2/diagnostics/

```

version

4. Perform Reindexing:

- a. Determine the contact center ID using the following command:

```
curl -u <ops-user>:<ops-pass> http://< selected-web-services-node>: <web-services-listening-port>/api/v2/ops/contact-centers; echo
```

The following output is returned:

```
{"statusCode":0,"uris":["http://< selected-web-services-node>: <web-services-listening-port>/api/v2/ops/contact-centers/<contact-center-id>"]}
```

- b. Run the following to regenerate indexing for Voice Recording if required. If you do not need to regenerate indexing for voice recording, skip this step and proceed to 4c.

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json" "http://<selected-web-services-node>:<web-services-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/recordings" -d '{
  "operationName":"forceIndex",
  "from": <start-range-in-milliseconds>,
  "to": <stop-range-in-milliseconds>,
  "purgeOld":false
}'
```

- c. Run the following command to regenerate indexing for Screen Recording:

```
curl -u <ops-user>:<ops-pass> -XPOST -H "Content-Type:application/json" "http://<selected-web-services-node>:<web-services-listening-port>/api/v2/ops/contact-centers/<contact-center-id>/screen-recordings" -d '{
  "operationName":"forceIndex",
  "from": <start-range-in-milliseconds>,
  "to": <stop-range-in-milliseconds>,
  "purgeOld":false
}'
```

Important

- If `purgeOld` is true, the entire index will be purged. For this reason, set `purgeOld` to false when only a period of your index is lost. The typical scenario is nightly indexing for a multi-tenancy deployment. When your existing ES map is wrong and you must rebuild the entire index, you should set `purgeOld` to true. If your restore range is too long and you want to restore each period at a time, verify that you set `purgeOld` to false after the first period is restored.
- Verify that the operation is not executed during the period at which the MLM or RCS archive job is scheduled.

4. Once all re-indexing operations are complete:

- Remove the change to the Hystrix timeout for the **RecordingOperationApiTaskV2** that was added in step 2.
- Restart the selected **Interaction Recording Web Services (Web Services)** node.

Understanding Genesys Interaction Recording

Thank you for calling Genesys - Your call may be recorded for quality and training purposes.

What does this really mean?

Genesys Interaction Recording (GIR) is a system that allows business managers to monitor the productivity and accuracy of the information that employees provide to customers.

GIR also provides business legal departments with the necessary evidence to ensure verbal commitments made by customers can be upheld in a court of law. For example, when verbally agreeing to a cell-phone contract extension. Consider the following examples:

- **Productivity**—An employee should know, or be able to find the correct procedure to advise a customer how to reset their password. Through listening to the information that the employee provides to the customer, or, watching the research steps that the employee takes to find this information - the employer should have a good indication of how productive that employee is.
- **Accuracy**— An employee should know, or be able to find (on-line) the correct information relating to product functionality. Through listening to the information that the employee provides to the customer, or, watching the content that the employee communicates to the customer in written formation (that is, watching what they type in email or chat) - the employer can be assured that the employee is properly representing what the product can do.
- **Legal**—If a customer is agreeing to a legal agreement (like extending their cell-phone contract), and subsequently claims that they have made no such agreement - the employer will be able to provide to legal officials a recording of the call which shows the truth.

GIR can:

- Record phone calls between a customer and an employee;
- Record employee screens - the actual screen that they are looking at while they're doing their work
 - while they're talking to a customer, or,
 - just to see what they're doing
- Allow for playback and sharing of the recordings "For Quality and Training purposes"
- Require no employee intervention to use since it is a background process

GIR enables businesses to perform these tasks.

Genesys will consult with business managers to advise them of any privacy concerns on a case-by-case basis - since, each country, state or business may have their own laws in this regard.

- Is it legal to record a customer conversation without their knowledge?

- Is it legal to record an employee screen without their knowledge?

The following table lists the acronyms and terminology used in the Genesys Interaction Recording solution.

Term/Acronym	Description
Audio recording	The traditional scenario where a call is recorded for quality purposes.
Active call recording	SIP based, VoIP telephony backend infrastructure.
Encryption	Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called plain text; encrypted data is referred to as cipher text.
GVP	Genesys Voice Platform
HTTP/HTTPS	Hypertext Transfer Protocol
IVR Profile	The description of the customer. An IVR Profile holds all the important information that is required to distinguish one customer from another, so interactions can be routed correctly.
Line of Business	Line of business (LOB) is a general term which often refers to a set of one or more highly related products which service a particular customer transaction or business need.
Location	<ul style="list-style-type: none"> • Node (location) and node path (location based hierarchy): A node represents a specific Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) instance. For example, if you have three Interaction Recording Web Services (Web Services) instances installed, you will have three nodes. A node can be identified using a node path. The node path is specified in the Interaction Recording Web Services (Web Services) application.yaml file (if you are using Web Services and Applications version 8.5.201.09 or earlier the node path is specified in the server-settings.yaml file instead). This node path must be unique for each Interaction Recording Web Services (Web Services) instance. Sometimes the node is called a 'location'. The preferred term is node. • Media Storage (location based storage): A storage location is a place where screen recordings or voice recordings are stored—for example, a WebDAV server. For voice recordings, the storage location is specified in the IVR Profile, and in the Interaction Recording Web Services (Web Services) recording settings. For screen recording, the storage location is specified in screen-recording settings. Also, for Screen Recording, a given node path can be associated with a specific storage. • Backup folder (location) for MLM: When Interaction Recording Web Services (Web Services)

Term/Acronym	Description
	performs a backup task, it will create an archive file that contains all the media and metadata for each recording, and store this somewhere on the local file system. The folder where the files are stored is sometimes called the <i>backup location/folder</i> or <i>archive location/folder</i> .
MCP	Media Control Platform—Part of the GVP suite of products.
MLM	Media Life Cycle Management—The tasks that include backing up and purging call and screen recording files.
MSML	Media Server Markup Language
Multiplexing	In telecommunications and computer networks, multiplexing (sometimes contracted to muxing) is a method by which multiple analog message signals or digital data streams are combined into one signal over a shared medium. The aim is to share an expensive resource.
PKCS7	See RFC 2315. Used to sign and/or encrypt messages under a PKI. Used also for certificate dissemination (for instance as a response to a PKCS#10 message). Formed the basis for S/MIME, which is as of 2010 based on RFC 5652, an updated Cryptographic Message Syntax Standard (CMS).
Policy based recording	Recording interactions based on a particular business requirement or strategy. For example, your customer may have many lines of business, requiring different compliance, bandwidth, encryption, and storage needs.
Real time monitoring	Observing the interaction activity and/or progression as it happens.
Remote agents	Agents working outside the contact center.
RTP	Real-time Transport Protocol
Screen recording for non-voice interactions	For chat, email or other non-real-time services where there is no strict continuity required in the communications with the customer.
Screen recording for voice interactions	The traditional scenario where a call is recorded for quality purposes; but, where the agent's screen is also captured to ensure they are taking the most efficient path towards finding a resolution to the problem.
WebDAV	Web Distributed Authoring and Versioning (WebDAV) is an extension of the Hypertext Transfer Protocol (HTTP) that allows clients to perform remote Web content authoring operations. A working group of the Internet Engineering Task Force (IETF) defined WebDAV in RFC 4918.
WFO	Workforce Optimization

Minimum Recommended Versions

The following tables contain a list of the minimum recommended versions for **Direct Dependencies** and **Indirect Dependencies**. You should check the contributing product release notes for important product information. Contact **Genesys Customer Care** if you have questions about your prerequisite versions.

Important

- You should check the contributing product release notes for important product information. Contact [Genesys Customer Care](#) if you have questions about your prerequisite versions.
- Versions appended with a * indicate that the dependent software has been changed to support either new functionality or it has corrected previously broken functionality. Review the release notes if you're considering to update only some of the dependent versions.

For current information about the latest features that were delivered, refer to the [New in this Release](#).

Direct Dependencies																			
Minimum Recommended Version TIP																			
TIP																			
GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	GIR	Release			
8.5.22	8.612	8.602	8.602	8.602	8.612	8.602	8.602	8.602	8.602	8.602	8.602	8.602	8.602	8.602	8.602	12.02			
Recording Processor Script 8.5.x	8.5.09	8.270	98.250	98.250	98.250	98.230	98.610	98.610	98.610	98.610	98.610	98.610	98.650	98.670	78.310	78.640	68.810	68.900	50.84
Voice Processor 9.0.x	9.0.00	9.025	9.025	9.025	9.025	9.025	9.030	0.003	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
Recording Crypto Server 8.5.x	8.5.09	8.820	98.380	98.330	98.330	98.330	98.360	98.360	98.360	98.360	98.360	98.360	98.360	78.310	78.310	78.940	68.320	68.320	50.73
Recording Plug-in for GAX 8.5.x (including	8.5.09	8.570	98.530	98.500	98.500	98.500	98.530	98.530	98.610	98.510	98.670	98.310	78.320	78.320	78.320	78.930	68.640	68.800	50.75

Genesys Interaction Recording Solution Guide 1099

Direct Dependencies Minimum Recommended Version TIP																	
TIP																	
VP Resource Manager	8.5.059.53	49.58	29.07	29.07	09.09	09.09	09.09	08.59	88.61	88.61	78.39	78.39	78.95	78.64	78.64	60.42	
SIP Server	8.1.108.34	08.96	08.96	08.76	08.06	08.06	08.06	08.06	08.73	08.73	08.54	08.54	08.54	08.54	08.42	02.28	
Genesys Info Mart	8.5.018.54	018.54	018.54	014.74	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Indirect Dependencies Minimum Recommended Version																	
TIP																	
GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	GIR Release	
	8.5.228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	228.6	22.02
Interaction Server	8.5.118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	118.50	05.04
Interaction Concentrator	8.1.518.55	18.17	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	18.05	09.07
Genesys Admin Portal Extension	9.0.109.01	109.06	109.06	108.56	98.69	98.69	98.69	98.69	58.52	58.52	48.59	28.54	28.54	28.54	28.54	28.54	20.20
Configuration Server	8.5.108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	108.50	00.16
Solution Control Server	8.5.108.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	08.34	00.19
Message Server	8.5.108.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	08.53	00.17
Local Control Agent	8.5.108.57	08.57	08.57	08.30	08.30	08.30	08.30	08.30	08.30	08.30	08.30	08.30	08.30	08.30	08.30	08.30	00.12
SNMP Master Agent	8.5.108.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	08.67	00.07
Universal Routing Server	8.1.408.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	08.36	00.27
Orchestration Server	8.1.408.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	08.47	00.00
Universal	8.5.208.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	08.57	

Indirect Dependencies Minimum Recommended Version															
TIP															
Contact Server															
Chat Server	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	8.5.20	N/A

GIR Alarms

The following is a list of the alarms. The list includes a summary of the possible cause and resolution for each alarm.

[+] Playback Error: Non-Encrypted

Playback Error: Non-Encrypted

Log ID: 40030

Problem

When trying to play back an interaction, the following common playback problems may occur:

- Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) Connection:
 - The connection cannot be started.
 - The connection is aborted accidentally.
- Interaction Recording Web Services (Web Services) Response:
 - Response has a non-200-OK HTTP status.
 - Response cookies cannot be parsed by RCS.
 - Response content does not contain the information RCS is looking for (for example, specific attributes in the JSON response).
- The requested media file cannot be found.
- The actual media file cannot be fetched using the information given by Interaction Recording Web Services (Web Services).
- An error occurs when trying to output the requested media.

Resolution

- Verify that a table connection can be established between RCS and Interaction Recording Web Services (Web Services).
- Verify that Interaction Recording Web Services (Web Services) responds with the content expected by RCS.
- Verify that Interaction Recording Web Services (Web Services) responds with the correct information.

[+] Playback Error - Encrypted

Playback Error - Encrypted**Log ID:** 40031**Problem**

- Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) Connection:
 - The connection cannot be started.
 - The connection is aborted accidentally.
- Interaction Recording Web Services (Web Services) Response:
 - Response has a non-200-OK HTTP status.
 - Response cookies cannot be parsed by RCS.
 - Response content does not contain the information RCS is looking for (for example, specific attributes in the JSON response).
- The requested media file cannot be found.
- The actual media file cannot be fetched using the information given by Interaction Recording Web Services (Web Services).
- An error occurs when trying to output the requested media.
- Pem problem:
 - The format is incorrect.
 - Pem is not of type PKCS7:
 - Unmatched OID.
 - PKCS7 pem cannot be parsed:
 - Invalid syntax.
 - Unexpected content.
- Password problem:
 - Password is incorrect for the Pem file.
- Decryption problem:
 - Encryption is not of type RSA.
 - Private key given by the Pem file is invalid.
 - Session key cannot be decrypted by the private key.
 - Encrypted media cannot be decrypted by the session key.
 - Encrypted media is corrupted.

Resolution

- Verify that a table connection can be established between RCS and Interaction Recording Web Services (Web Services).
- Verify that Interaction Recording Web Services (Web Services) responds with the content expected by RCS.
- Verify that Interaction Recording Web Services (Web Services) responds with the correct information.
- Verify that the pem returned by Interaction Recording Web Services (Web Services) is valid PKCS7.
- Verify that RCS is configured with the correct pass phrase.
- Verify that the encrypted media is not corrupted.

[+] Add Certificate

Add Certificate

Log ID: 40022

Problem

- Certificate problem:
 - The certificate cannot be parsed.
 - The certificate does not have an X509 format.
 - The information cannot be extracted from the certificate.
 - The Public key in the certificate is not of type RSA.
- Private key problem:
 - The Private key not in pem format.
 - The Private key is encrypted while a password is not provided.
 - The Encrypted private key cannot be parsed.
- Decryption problem:
 - The Encrypted Private key cannot be decrypted by the password provided.
- Certificate-key unmatched problem:
 - The text encrypted by the certificate's public key cannot be decrypted by the provided private key.
- The certificate to be added already exists in the key store.

Resolution

- Verify that the certificate provided is valid.
-

- Verify that the provided key is valid.
- Verify that the password provided can decrypt the private key if it is encrypted.
- Verify that the certificate and the key provided are a pair, so that what should be encrypted by the certificate's public key can be decrypted by the key.

[+] Delete Certificate

Delete Certificate

Log ID: 40023

Problem

- The certificate alias that should be deleted cannot be parsed.
- The certificate to be deleted cannot be found.

Resolution

- Verify that the alias has the format with the prefix: *prefix_tenantName:tenantDbid:issuerDN:serialNo.*

[+] Error accessing agent hierarchy (SWITCH NAME, AGENT ID) from Config Server cache

Error accessing agent hierarchy (<SWITCH NAME>, <AGENT ID>) from Config Server cache

Problem

When connecting to the Configuration Server an error occurs when the Configuration Server attempts to retrieve information about agent hierarchy.

Resolution

- Open the **rpconfig.cfg** file and review the **[config_server]** section. **Verify that the Configuration Server settings are correct for the following options:** hostname, port, username, password, backup_host, backup_port
- Verify that there is a stable connection between Recording Processor Script (RPS) and Configuration Server.
- Verify that the Configuration Server is up and running.

[+] RP failed to access local queue for message processing

RP failed to access local queue for message processing

Problem

The Recording Processor Script does not interact as expected with the local SQLite file.

Resolution

- Open the **rpconfig.cfg** file and review the **[persistence]** section. Verify that **db_filename** is filled out and the proper SQLite file is created.
- Verify that the **SQLite** file is not corrupted.
- Verify that the Recording Processor has read/write access to the **SQLite** file.
- Verify that other processes are not using the **SQLite** file. If the SQLite file is being used by other processes, consider using a new SQLite file, and make changes to the other processes.

[+] JSON loading error. Deleting record: UUID

JSON loading error. Deleting record: <UUID>

Problem

An problem occurs when loading the MCP record that was saved to a local SQLite file.

Resolution

- Verify that **SQLite** is not corrupted.
- Verify that the column **data** contains **JSON** strings. The column can be found in the table defined in the **[persistence]** section.

[+] Could not access metadata...

Could not access metadata...

Problem

A problem occurs when the Recording Processor Script (RPS) retrieves stored metadata saved to a local **SQLite** file.

Resolution

- Verify that the **[persistence]** section in **rpconfig.cfg** points to the correct table and **SQLite** file.
-

- Verify that the **SQLite** file is not corrupted.
- Verify that the Recording Processor has read/write access to the **SQLite** file.
- Verify that other processes are not using the **SQLite** file. If the SQLite file is being used by other processes, consider using a new SQLite file, and make changes to the other processes.

[+] Fail to retrieve update from config server, connection might be broken

Fail to retrieve update from config server, connection might be broken

Problem

Failed to retrieve updated information from the Configuration Server.

Resolution

- Verify that the Configuration Server is running.
- Verify that the **[config_server]** section in **rpconfig.cfg** is configured correctly.
- Verify that there is a stable connection between Recording Processor Script and the Configuration Server.

[+] Could not access contact center: URI

Could not access contact center: <URI>

Problem

Failed to retrieve information from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) about a specific contact center.

Resolution

- Verify that there is a stable connection between Recording Processor Script and Interaction Recording Web Services (Web Services).
- Verify that the contact center is still available and has not been deleted.
- Verify that Interaction Recording Web Services (Web Services) is up and running.

[+] Unable to query Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for list of contact centers: CC URI

Unable to query Interaction Recording Web Services (Web Services) for list of contact centers: <CC URI>**Problem**

Failed to retrieve information from Interaction Recording Web Services (Web Services) regarding a list of contact centers.

Resolution

- Verify that there is a stable connection between Recording Processor Script and Interaction Recording Web Services (Web Services).
- Verify that the **[htcc]** section in **rpconfig.cfg** has the correct credentials.
- In the **[htcc]** section in the **rpconfig.cfg** file, verify that the **contact_center_uri** points to the correct GET request, and **base_uri** points to the correct Interaction Recording Web Services (Web Services) node.
- Verify that Interaction Recording Web Services (Web Services) is up and running.

[+] Fail to parse party information of call UUID with Exception, EXCEPTION**Fail to parse party information of call <UUID> with Exception, <EXCEPTION>****Problem**

Failed to parse information retrieved from the ICON database.

Resolution

- Verify that the **[processing]** section in **rpconfig.cfg** is set with the correct **Encoding** option.
- Verify that the information returned from ICON is not missing information. Ensure validity.

[+] Could not read ICON DB configuration.**Could not read ICON DB configuration.****Problem**

Failed to parse **rpconfig.cfg** to obtain information regarding the ICON databases.

Resolution

- Verify that the switches in the **[icon_db_servers]** section in **rpconfig.cfg** point to the correct ICON databases.
- Every ICON database specified in the section **[icon_db_servers]** requires its own section in the form of **[{database_name}_db_info]**. Each section must contain the following information: **dbserver_host**, **dbserver_port**, **username**, **password**, **dbname**, **dbms**, **dbengine**.

[+] Fail to query agent state update from ICON

Fail to query agent state update from ICON

Problem

Failed to retrieve information from the ICON database regarding agent state.

Resolution

- Verify that the switches in the **[icon_db_servers]** section in **rpconfig.cfg** point to the correct ICON databases.
- Every ICON database specified in the section **[icon_db_servers]** requires its own section in the form of **[{database_name}_db_info]**. Each section must contain the following information: **dbserver_host**, **dbserver_port**, **username**, **password**, **dbname**, **dbms**, **dbengine**.

[+] Fail to parse ICON result for agent state update, EXCEPTION

Fail to parse ICON result for agent state update, <EXCEPTION>

Problem

Failed to parse information retrieved from ICON.

Resolution

- Verify that the ICON database is using the correct schema (specifically **G_AGENT_STATE_HISTORY**). The data should be valid and values in the following columns should be integers: **added_ts**, **prevstate**, **state**.

[+] No ICON DB configuration information found!

No ICON DB configuration information found!

Problem

Failed to parse **rpconfig.cfg** to obtain information regarding the ICON databases.

Resolution

- Verify that the switches in the **[icon_db_servers]** section in **rpconfig.cfg** point to the correct ICON databases.
- Every ICON database specified in the section **[icon_db_servers]** requires its own section in the form of **[{database_name}_db_info]**. Each section must contain the following information: **dbserver_host**, **dbserver_port**, **username**, **password**, **dbname**, **dbms**, **dbengine**.

[+] Error Message - Can not retrieve data from DB

<Error Message> Can not retrieve data from DB

Problem

Failed to get SQL results from the ICON database.

Resolution

- Verify that the **[icon_db_servers]** section in **rpconfig.cfg** and the ICON database are configured correctly. The ICON databases require information regarding: **dbserver_host**, **dbserver_port**, **username**, **password**, **dbname**, **dbms**, **dbengine**
- Verify that the ICON database exists and is reachable on the target host.
- Verify that there is a stable connection between Recording Processor Script and the ICON databases.

[+] Failed to POST to URI

Failed to POST to <URI>

Problem

POST requests to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) / SpeechMiner failed.

Resolution

- Verify that there is a stable connection between Recording Processor Script and SpeechMiner, as well as the Recording Processor and Interaction Recording Web Services (Web Services).
- Verify that the failed URI is a valid URI used for POST requests.

- Verify that the **post_uri** and **base_uri** options and the credentials in the **[speechminer]** section in **rpconfig.cfg** are configured properly.
- Verify that the **post_uri** and **base_uri** options and the credentials in the **[htcc]** section in **rpconfig.cfg** are configured properly.
- If the POST request is related to a contact-center, ensure the contact center still exists.

[+] Could not access event time: EVENT

Could not access event time: <EVENT>

Problem

When processing the events from ICON, the system is unable to find the event time.

Resolution

- Verify that the ICON data in the **G_PARTY_HISTORY** and **G_PARTY** tables is correct.
- Verify the data stored on the local **SQLite** file is correct. To verify the correct table, look at the **[persistence]** section in **rpconfig.cfg**
- SQLite file is corrupted

[+] Could not access event uuid: EVENT

Could not access event uuid: <EVENT>

Problem

When processing the events from ICON, the system is unable to find the event UUID.

Resolution

- Verify that the ICON data in the **G_PARTY_HISTORY** and **G_PARTY** tables is correct.
- Verify the data stored on the local **SQLite** file is correct. To verify the correct table, look at the **[persistence]** section in **rpconfig.cfg**
- SQLite file is corrupted

[+] Could not access last event: EVENT ID

Could not access last event: <EVENT ID>

Problem

When attempting to process a series of events to retrieve the latest known time a number of issues arise.

Resolution

- Verify that the ICON data in the **G_PARTY_HISTORY** and **G_PARTY** tables is correct.
- Verify the data stored on the local **SQLite** file is correct. To verify the correct table, look at the **[persistence]** section in **rpconfig.cfg**
- SQLite file is corrupted

[+] Could not parse stop time

Could not parse stop time

Problem

When a record is processed, the information appears to be invalid because Recording Processor Script has issues parsing the data.

Resolution

- Verify that the metadata received from MCP is in the proper JSON format.
- Verify the data stored in the local **SQLite** file is correct and not corrupted.
- Verify if additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.

[+] Could not mask credentials

Could not mask credentials

Problem

When a record is processed, the information appears to be invalid because Recording Processor Script has issues parsing the data.

Resolution

- Verify that the metadata received from MCP is in the proper JSON format.
 - Verify the data stored in the local **SQLite** file is correct and not corrupted.
-

- Verify if additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.

[+] Back up metadata record failed: cannot make directory [DIRECTORY]

Back up metadata record failed: can't make directory [<DIRECTORY>].

Problem

Recording Processor Script is having problems trying to create a directory on the local file system.

Resolution

- Verify that the Recording Processor instance has permission to read and write at the specified directory and its parent directory.
- Verify that there is enough disk space on the local hard drive to allow the write.

[+] Back up metadata record into [FILE NAME] failed.

Back up metadata record into [<FILE NAME>] failed.

Problem

Recording Processor Script is having problems trying to create a file on the local file system.

Resolution

- Verify that the Recording Processor instance has permission to read and write at the specified directory.
- Verify that there is enough disk space on the local hard drive to allow the write.

[+] Error getting record from GWS RECORD ID

Error getting record from GWS<RECORD ID>

Problem

It is not possible to retrieve record information from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) about a specific record.

Resolution

- Verify that the following properties are set properly in the **[htcc]** section in **rpconfig.cfg**: **base_uri**, **get_uri**, **contact_center_uri**, **username**, **password**.
- Verify that there is a stable connection between Recording Processor Script and Interaction Recording Web Services (Web Services).
- Verify that the contact center is available and has not been deleted.
- Verify that Interaction Recording Web Services (Web Services) is up and running.

[+] Processing error

Processing error

Problem

- Recording Processor Script is having problems interacting with the local **SQLite** file.
- It is not possible to read the metadata stored in the **SQLite** file.

Resolution

- Verify if the **[persistence]** section in **rpconfig.cfg** points to the correct table and **SQLite** file.
- Verify that the **SQLite** file exists, as specified in **rpconfig.cfg**.
- Verify that the **SQLite** file has not been corrupted.
- Verify that Recording Processor has read/write access to the **SQLite** file.
- Verify that additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.
- Verify if the **data** column contains JSON strings. The column can be found in the table defined in the **[persistence]** section.

[+] Could not access event uuids: UUID

Could not access event uuids: <UUID>

Problem

A problem occurred when sorting UUIDS and times when processing the events from ICON.

Resolution

- Verify that the **ICON** data in the **G_PARTY_HISTORY** and **G_PARTY** tables are correct.
- Verify if the metadata received from MCP is in the proper **JSON** format.
- Verify that the data stored in the local **SQLite** file is correct. To verify which table may be problematic, look at the **[persistence]** section in **rpconfig.cfg**.
- Verify that the **SQLite** file is not corrupt.

[+] Could not merge records. Skipping merge: UUID

Could not merge records. Skipping merge: <UUID>

Problem

While trying to consolidate numerous records into a single record:

- The system failed to parse information properly.
- A problem occurred when writing or reading to the local **SQLite** file.

Resolution

- Verify that the **ICON** data in the **G_PARTY_HISTORY** and **G_PARTY** tables are correct.
- Verify that the data stored in the local **SQLite** file is correct. To verify which table may be problematic, look at the **[persistence]** section in **rpconfig.cfg**.
- Verify that the **SQLite** file is not corrupt.
- Verify that Recording Processor has read/write access to the **SQLite** file.

[+] Could not access metadata to merge...

Could not access metadata to merge...

Problem

It is not possible to properly access information in the metadata.

Resolution

- Verify that the **ICON** data in the **G_PARTY_HISTORY** and **G_PARTY** tables are correct.
 - Verify that the data stored in the local **SQLite** file is correct. To verify which table may be problematic, look at the **[persistence]** section in **rpconfig.cfg**.
-

- Verify that the **SQLite** file is not corrupt.

[+] Could not save merged metadata: UUID

Could not save merged metadata: <UUID>

Problem

Deleting records from the local **SQLite** file is problematic.

Resolution

- Verify that **db_filename** in the **[persistence]** section in **rpconfig.cfg** is filled out and the proper **SQLite** file is created.
- Verify that **table_name** in the **[persistence]** section in **rpconfig.cfg** is correct.
- Verify that the **SQLite** file is not corrupt.
- Verify that Recording Processor has read/write access to the **SQLite** file.

[+] Exception thrown when applying interaction callType

Exception thrown when applying interaction callType

Problem

A problem occurred when applying a call type to the record.

Resolution

- Verify that the metadata received from MCP is in the proper **JSON** format.
- Verify that the **SQLite** file is not corrupt.

[+] Could not add partitions: UUID

Could not add partitions: <UUID>

Problem

Unable to parse the recordings properly to add the partition properly.

Resolution

- Verify that the metadata received from MCP is in the proper **JSON** format.
- Verify that the **SQLite** file is not corrupt.

[+] Invalid lock released: ID

Invalid lock released: <ID>

Problem

Internal error with Recording Processor Script locking mechanism.

Resolution

Nothing immediate can be done for this alarm. Contact your Genesys contact for help

[+] Fail to start web server URL

Fail to start web server @https://<URL>

Problem

You cannot start the Recording Processor Script because the specified port is in use.

Resolution

Verify that the **port** option in the **[rp_server]** section in **rpconfig.cfg** is configured properly.

[+] Could not parse auth. header

Could not parse auth. header

Problem

Recording Processor Script cannot parse an HTTP request.

Resolution

Ensure the the HTTP request sent to the Recording Processor is correct. If there is an HTTP header called **AUTHORIZATION**, verify that it has the **Basic <ENCODED CREDENTIALS>** form."

[+] Failed to process metadata: DATA

Failed to process metadata:

Problem

There is a problem with parsing the data received from MCP.

Resolution

Verify that the metadata received from MCP is in the proper **JSON** format.

[+] Contact center (ID1) does not match (ID2). ID: ID3

Contact center (<ID1>) does not match (<ID2>). ID: <ID3>

Problem

The **CCID** in the local **SQLite** file and the record from MCP do not match correctly for the same metadata ID.

Resolution

Ensure that **rpconfig.cfg** contains the correct configurations. If the configurations are correct, contact Genesys Support for additional help.

[+] Unable to initialize contact center cache

Unable to initialize contact center cache

Problem

Recording Processor Script is having problems starting creating a cache.

Resolution

- Verify that the Recording Processor Script is installed properly with the recommended version of Python. If it is installed properly, please contact Genesys Support for additional help.

[+] Unable to access configuration server data

Unable to access configuration server data

Problem

The Recording Processor Script connection to the Configuration Server is problematic.

Resolution

- Verify that the Configuration Server settings in **[config_server]** section in **rpconfig.cfg** are correct for the following options: hostname, port, username, password, backup_host, backup_port.
- Verify that there is a stable connection between Recording Processor Script and the Configuration Server.
- Verify that the Configuration Server is up and running.

[+] error in backup thread

error in backup thread

Problem

A problem occurred when running the Recording Processor in backup mode. May be attributed to either:

- **SQLite** file.
- Internal implementation of threads and locks.

Resolution

- Verify that the **[persistence]** section in **rpconfig.cfg** points to the correct table and **SQLite** file.
- Verify that the **SQLite** file exists, as specified in **rpconfig.cfg**.
- Verify that the **SQLite** file is not corrupt.
- Verify that the Recording Processor has read/write access to the **SQLite** file
- Verify that additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.

[+] SQLite3 error: SQL CMD - PARAMETERS:ERROR MSG

SQLite3 error: <SQL CMD> - <PARAMETERS>:<ERROR MSG>

Problem

An SQL call was made, and an issue occurred.

Resolution

- Verify that the **[persistence]** section in **rpconfig.cfg** points to the correct table and **SQLite** file.
- Verify that the **SQLite** file exists, as specified in **rpconfig.cfg**.
- Verify that the **SQLite** file is not corrupt.
- Verify that the Recording Processor has read/write access to the **SQLite** file
- Verify that additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.

[+] SQLite3 warning: SQL CMD - PARAMETERS

SQLite3 warning: <SQL CMD> - <PARAMETERS>

Problem

An SQL call was created and tried to perform multiple commands at the same time.

Resolution

Verify that the parameters are correct for the SQL command. Verify whether or not it is trying to invoke another command. If yes, contact Genesys Support for help.

[+] SQLite3 rollback error

SQLite3 rollback error

Problem

A problem occurred when attempting a rollback with the local **SQLite** file.

Resolution

- Verify that the **SQLite** file is not corrupt.
- Verify that the Recording Processor has read/write access to the **SQLite** file.
- Note: There may be nothing to roll back, since this error may just be a warning.

[+] error in processing thread

error in processing thread

Problem

Issues occurred for Recording Processor Script processes that run in "active" mode. This message is used as a catchall for many different errors in the Recording Processor Script. The issues may include:

- Processing metadata from MCP.
- Processing data from ICON.
- Encoding
- SQLite
- Internal handling

Resolution

- Verify that the metadata received from MCP has the proper **JSON** format.
- Verify that the data found in **ICON** are correct.
- Verify that the **[persistence]** ' section in *rpconfig.cfg* points to the correct table and **SQLite** file.
- Verify that the data stored in the local SQLite file is correct. To verify that correct table, look at the '[persistence]' section in *rpconfig.cfg*.
- Verify that the **SQLite** file is not corrupt.
- Verify that the Recording Processor has read/write access to the **SQLite** file.
- Verify that additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.
- If the above resolutions do not fix the issue, please contact Genesys Support for help.

[+] Fatal exception occurred during metadata processing

Fatal exception occurred during metadata processing

Problem

An error occurred when starting the Recording Processor Script in **Active** mode.

Resolution

- Verify that the Recording Processor Script is installed properly with the recommended version of Python. If it is installed properly contact Genesys Support for help.

[+] SHUTTING DOWN BECAUSE OF FATAL ERROR

**** SHUTTING DOWN BECAUSE OF FATAL ERROR ****

Problem

An error occurred when starting the Recording Processor Script in **Active** mode.

Resolution

- Verify that the Recording Processor Script is installed properly with the recommended version of Python. If it is installed properly contact Genesys Support for help.

[+] Failed to reconnect to config server

Failed to reconnect to config server

Problem

Recording Processor Script is having problems connecting to the Configuration Server.

Resolution

-
- Verify that the Configuration Server settings in **[config_server]** section in rpconfig.cfg are correct for the following options: hostname, port, username, password, backup_host, backup_port.
- Verify that there is a stable connection between the Recording Processor Script and the Configuration Server.
- Verify that the Configuration Server is up and running.

[+] Fail to parse ICON customized data: ERROR

Fail to parse ICON customized data: <ERROR>

Problem

Recording Processor Script is having problems parsing the information from ICON.

Resolution

- Verify that the data received from ICON is correct.

- Verify that the date is a proper integer that can be converted into a proper ISO 8601 format.

[+] Failure: DESCRIPTION Headers: HEADERS

Failure: <DESCRIPTION> Headers: <HEADERS>

Problem

The response from a POST or GET request was not successful.

Resolution

- Verify that all the options in the **[speechminer]** section in **rpconfig.cfg** are configured properly (if this is being used): `base_uri`, `post_uri`, `username`, `password`, `disable_ssl_certificate_validation`.
- Verify that all the options in the **[htcc]** section in **rpconfig.cfg** are configured properly (if this is being used): `base_uri`, `post_uri`, `get_uri`, `contact_center_uri`, `username`, `password`, `disable_ssl_certificate_validation`.
- Verify that Speechminer / Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) are in an operational state.

[+] Content: CONTENT

Content: <CONTENT>

Problem

The response from a POST or GET request was not successful.

Resolution

- Verify that all the options in the **[speechminer]** section in **rpconfig.cfg** are configured properly (if this is being used): `base_uri`, `post_uri`, `username`, `password`, `disable_ssl_certificate_validation`.
- Verify that all the options in the **[htcc]** section in **rpconfig.cfg** are configured properly (if this is being used): `base_uri`, `post_uri`, `get_uri`, `contact_center_uri`, `username`, `password`, `disable_ssl_certificate_validation`.
- Verify that Speechminer / Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) are in an operational state.

[+] Failed to GET record from URL ERROR MSG

Failed to GET record from <URL><ERROR MSG>**Problem**

Recording Processor Script has an issue with a GET request to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier).

Resolution

- Verify that all the options in the **[htcc]** section in **rpconfig.cfg** are configured properly (if this is being used): `base_uri`, `post_uri`, `get_uti`, `contact_center_uri`, `username`, `password`, `disable_ssl_certificate_validation`.
- Verify that there is a stable connection between the Recording Processor Script and Interaction Recording Web Services (Web Services).
- Verify that Interaction Recording Web Services (Web Services) is up and running.
- Verify that SM is sending back the correct response. The response must be decode-able for usage.

[+] Failed to parse record from URL ERROR MSG Content: PAY LOAD**Failed to parse record from <URL><ERROR MSG> Content: <PAY LOAD>****Problem**

Recording Processor Script has an issue parsing a GET request from Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier).

Resolution

Verify that SpeechMiner is sending back the proper response that is decode-able for usage.

[+] Failed to vacuum**Failed to vacuum****Problem**

Recording Processor Script failed to call VACUUM on the ICON database.

Resolution

Verify that the ICON database is operational and not corrupted.

[+] Failed to merge data

Failed to merge data

Problem

Recording Processor Script is having problems merging data from MCP to data in the local **SQLite** file.

Resolution

- Verify that the metadata received from MCP is in the proper **JSON** format.
- Verify that the '*[persistence]*' section in `rpconfig.cfg` points to the correct table and **SQLite** file.
- Verify that the **SQLite** file is not corrupt.
- Verify that the Recording Processor has read/write access to the **SQLite** file.
- Verify that additional processes are using the **SQLite** file. If additional processes are using a new **SQLite** file, change those processes to use another **SQLite** file, or create a new **SQLite** file for Recording Processor.

[+] Failed to compile regex pattern for ud_filter: ERROR

Failed to compile regex pattern for ud_filter: <ERROR>

Problem

Recording Processor Script has an issue compiling a regex pattern.

Resolution

Verify that the **attached_data_filter** option is in the **[filter]** section.

[+] Failed to compile regex pattern for acw_filter: ERROR

Failed to compile regex pattern for acw_filter: <ERROR>

Problem

Recording Processor Script has an issue compiling a regex pattern.

Resolution

Verify that the **acw_custom_data_filter** option is in the **[filter]** section.

[+] Failed to compile regex pattern for ud_filter_exception: ERROR**Failed to compile regex pattern for ud_filter_exception: <ERROR>****Problem**

Recording Processor Script has an issue compiling a regex pattern.

Resolution

Verify that the **attached_data_filter_exception** option is in the **[filter]** section.

[+] Failed to compile regex pattern for acw_filter_exception: ERROR**Failed to compile regex pattern for acw_filter_exception: <ERROR>****Problem**

Recording Processor Script has an issue compiling a regex pattern.

Resolution

Verify that the **acw_custom_data_filter_exception** option is in the **[filter]** section.