EX Engage Connector Deployment Guide

Recording Deployment

12/14/2025

Contents

Recording Deployment

Prerequisites

No-media Integration Deployment

EXEC deployment for No-media Integration should be up and running before the configuration required for the Recording Injection is added to the environment. See No-media Deployment for more information.

Mandatory Environment Elements

WebDAV

WebDAV is required only for the Recording Injection. EX Engage Connector supports any standard WebDAV server. Customers should plan to use 30 GB storage for WebDAV server for contact centers with 1000 concurrent recording calls.

HTTP Load Balancer

HTTP Load Balancer is required only for the Recording Injection. EX Engage Connector supports any standard HTTP load balancer.

> **Important**
>
> Genesys does not deploy and operate databases, WebDAV servers, or load balancers in on-premises deployments. It is the responsibility of the customer. In a production deployment, data store components (Redis) and WebDAV must be deployed outside of the Docker network and managed by the customer's DBA team. The customer's DBA team is also responsible for ensuring that the data store components are configured with the appropriate scalability, resiliency, and data protection (backups).

Media Control Platform (MCP)

The minimal version required for MCP is 9.0.052.00 for the EXEC Recording Injection feature.

Virtual Machines for the EXRP Component

- A minimum of 2 VMs (N+1) for EX Engage Connector Recording Provider (EXRP)

The EXEC VM must sync the system time at least once every 24 hours as the communication between the EXEC services and Genesys Cloud are very time-sensitive.

All VMs running EX Engage Connector components should belong to the same local network segment and be interconnected so that all components can communicate over the network. EXEC components can use either FQDNs or IP addresses to establish communication with each other. The connectivity table lists both incoming and outgoing connections required by the EXEC components.

Service	Direction	Protocol	Local Port	Remote Peer	Remote Peer Port	Purpose
MCP	Outgoing	HTTP	ANY	WebDAV	80 (default)	Post the recording files to WebDAV Server using HTTP requests.
MCP	Outgoing	HTTP	ANY	HTTP Load Balancer	HTTP_LB_EXCP_PORT	Deliver recording metadata using HTTP POST requests.
HTTP Load Balancer	Outgoing	HTTP	ANY	EXCP	3650 (default)	HTTP POST proxied by HTTP Load Balancer for delivering recording metadata.
EXCP	Outgoing	HTTP	Any	HTTP Load Balancer	HTTP_LB_EXRP_PORT	HTTP POST to HTTP Load Balancer for interaction-end with recording metadata.
HTTP Load Balancer	Outgoing	HTTP	ANY	EXRP	8080 (default)	HTTP POST proxied by HTTP Load Balancer to EXRP for interaction-end with recording metadata.
EXRP	Outgoing	TCP	Any	WebDAV	80 (default)	HTTP request by EXRP to fetch recording.
EXCP	Outgoing	TCP	Any	GIR-RWS	443 (default)	HTTPS GET Request by

Service	Direction	Protocol	Local Port	Remote Peer	Remote Peer Port	Purpose
						EXCP to GIR to retrieve recording metadata. (Applicable only for GIR deployments)

High-Level Deployment Plan

1. Refer the sample deployment procedure for WebDAV & Load Balancer.
2. Execute the procedure described in the Configuring hybrid_integration transaction in CME section.
3. Execute the procedure described in Configuring record IVR profile inside Voice Platform Profiles under CME section.
4. Download the EXRP image from Genesys FTP:
    - EXRP (exrp-100.0.100.XX.tgz) - EXRP image
5. Load the EXRP image to the local container registry.
    - Extract the tarball of the EXRP component image from the *.tgz archives.
    - Deploy the tarball as an image to the local container registry using docker import command.
    - Save the paths to the images in the registry (for example myregistry.mycompany.com/exec) to use them later in the deployment process.
6. Execute the procedure described in the Configuring .env file section.
7. Execute the procedure described in the Deploying EXRP section.

Deployment Procedure

Reference deployment for WebDAV server and load balancer

Deploying Apache WebDAV Server for Recording Injection

1. Deploy WebServer that supports the WebDAV module and ensure that the WebDAV module is installed and enabled as well.
2. Create the directory to keep the recording files, and provide the required permission for the webserver user to access those recording files.

3. Create the user and password and configure it for authentication of the recording folder.
4. Open the firewall for the port used by the WebDAV server.

The sample installation and configuration for the Apache2 WebDAV server is described below:

1. Install Apache Web Server. The installation process for WebDAV Server on an Ubuntu machine can be found at How to install Apache2 page.
2. Check if the DAV module is installed and enabled in the machine.
3. Edit the **/etc/apache2/apache2.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>
   ```

4. Open the firewall. Ensure that the default incoming port 80 is open.
5. Create the directory to keep the recording files, and set the permission to Apache, using the following commands

   ```
   mkdir /mnt/recordings
   chown www-data:www-data /mnt/recordings
   ```

6. 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
   ```

7. Configure the Apache HTTP server start on boot up (and start it now) using the following commands:

   ```
   sudo systemctl enable apache2
   ```

8. Test the Apache HTTP Server installation:
   - Upload a sample **hello.world** file to the Apache HTTP Server using the following command:

     ```
     curl -T hello.world -u user:password http://<WEBDAV_URL>/hello.world  //
     WEBDAV_URL: <WEBDAV_VM_ADDRESS>:<WEBDAV_PORT>/recordings
     ```

   - Using a browser, open the http://<WEBDAV_URL>/hello.world URL. The browser will request user credentials.

Deploying HTTP Load Balancer for Recording Injection

1. Deploy HTTP Load Balancer in a different VM from the EXEC service VMs.
2. Configure the upstream instances for EXCP and EXRP that will be used for load balancing the request.

3. Configure the port that the HTTP Load Balancer will listen for both EXCP and EXRP requests.

The sample installation and configuration for the NGNIX HTTP Load Balancer is described below:

1. Install NGNIX Proxy in a different VM which is accessible to EXEC VM. See Installing nginx for information on installing nginx..
2. Edit the **nginx-upstreams.conf** under the **/etc/nginx/** directory in NGINX VM.

```
upstream excp {
    server <EXCP_1_VM_HOST_PAIR_IPADDR1>:<EXCP_1_PORT> max_fails=0;
    server <EXCP_1_VM_HOST_PAIR_IPADDR2>:<EXCP_1_PORT> max_fails=0;
}
upstream exrp {
    server <EXRP_VM_HOST_IPADDR1>:<EXRP_PORT> max_fails=0;
    server <EXRP_VM_HOST_IPADDR2>:<EXRP_PORT> max_fails=0;
    server <EXRP_VM_HOST_IPADDR3>:<EXRP_PORT> max_fails=0;
}
```

3. Edit the **nginx.conf** file under **/etc/nginx/** directory in NGINX VM.

```
events {
    worker_connections 1000;
}

http {
    include nginx-upstreams.conf;

    # conversation-provider
    server {
        listen <HTTP_LB_EXCP_PORT>; #eg: listen 3650

        location / {
            proxy_pass http://excp/;
            proxy_next_upstream_tries 3;
        }
    }

    # recording-provider
    server {
        listen <HTTP_LB_EXRP_PORT>; #eg: listen 8080

        location / {
            proxy_pass http://exrp/;
            proxy_next_upstream_tries 3;
        }
    }
}
```

Configuring hybrid_integration transaction in CME

Common Configuration

This section contains configuration required for both third-party recorders and GIR.

Transaction Object

Transaction object hybrid_integration contains EXEC configuration. It should be configured as:

- Path to the transaction object:
  - for the single-tenant Engage deployment the transaction object should be created in the Transactions configuration unit in the Resources structure
  - for the multi-tenant Engage deployment there must be a separate transaction object under each of the tenant. Each transaction should point at a dedicated EX Org. EX Orgs cannot be shared by multiple Engage tenants.
- Name: *hybrid_integration*
- Type: *List*

Transaction Annex Folders

Folder	Parameter Name	Value	Description
conversation_provider	recording_enabled	True	Enable Recording Support. Default Value: False.
	sta_default_programid	<DEFAULT_STA_PROGRAM_ID>	Program UUID for topic detection
	sta_default_language	<DEFAULT_STA_LANGUAGE>	Represents the main Contact Center language. The ISO 639-1 two letter language code + '-' + ISO 3166-1 alpha-2 two letter country code string. The language should be chosen from the list of languages supported by Genesys Cloud (see *Native voice transcription* section).
	sta_userdata_language_key	<STA_USERDATA_LANGUAGE_KEY>	Key name for the `languageId` sent in user data from Routing Point, for example, "language" or "language_key".
recording-uploader.exec.transientStorage	type	WebDAV	The value must be WebDAV.
	username	WEBDAV_USERNAME	The username used for downloading recording files from WebDAV.
	password	WEBDAV_PASSWORD	Password used for downloading recording files from WebDAV.
	url	WEBDAV_URL	WebDAV URL from

Folder	Parameter Name	Value	Description
			where the recording files are downloaded.
recording-uploader.exec.genesysCloud	client_id	CFG_ALIAS_EX_ORG_OAUTH_CLIENT_ID	OAuth Client ID.
	password	CFG_ALIAS_EX_ORG_OAUTH_CLIENT_SECRET	OAuth Client Secret.

Third-party recorder integration

For third-party recorders, no additional configuration is required.

GIR Recorder integration

For integrating with GIR, add the following parameters to the transaction object.

Folder	Parameter Name	Value	Description
conversation_provider	gir_orgId	*<Engage_Contact_Center_ID>*	The Engage contact center ID to be used in request to GIR RWS. This value is same as the value of `rp.defaultccid` configured in RWS.
	gir_user	user1	GIR RWS user Important This is Ops user from RWS deployment specified under Ops Account section in Interaction Recording
	gir_password	*<RWS_PASSWORD>*	GIR RWS user password
	gir_metadata_fetch_delay	10	Determines the initial delay (in minutes) for trying out the first GIR request after the call completes.
	gir_metadata_fetch_interval	1380	Determines the time (in minutes) after which the retry of failed GIR request has to be made.
	gir_metadata_fetch_duration	1440	Determines how long (in minutes) the GIR request has to be retried.

Configuring record IVR profile inside Voice Platform Profiles under CME

Third-party recorder integration

Engage deployments with third-party recording solutions use recording client *recdest* parameters for connecting to third-party recorders. It is expected that parameter *callrec_dest* is not used for the third-party recorder integration and can be utilized to direct recording metadata to EXEC.

For recording injection functionality, MCP posts the recording file into the WebDAV server and also it posts the recording metadata to the EXCP service. As part of this process, the recording IVR profile inside the Voice Platform Profile must be configured.

Folder	Parameter	Value	Description
gvp.general	service-type	`record`	IVR profile type: always set to *record*.
gvp.service-parameters	recordingclient.callrec_dest	`fixed,http://<HTTP_LB_HOST_IPADDR>:<HTTP_LB_EXCP_PORT>/a v1/recording`	HTTP URL for posting recording metadata; pointed at the NGINX front-ending all EXCP instances.
gvp.service-parameters	recordingclient.httpauthorization2	`fixed,<WEBDAV_USERNAME>:<WEBDAV_PASSWORD>`	Credentials to post recordings to the intermediate WebDAV recording storage.
gvp.service-parameters	recordingclient.recdest2	`fixed,http://<WEBDAV_URL>/`	HTTP URL of the intermediate WebDAV recording storage.
gvp.service-parameters	recordingclient.type2	`fixed,audio/opus`	Recording file format. Currently only the opus format is supported.

GIR Recorder integration

Storing All Engage Recordings in WebDAV

See Storing All Engage Recordings in WebDAV for the description of this type of EXEC integration.

To configure, follow the following:

1. Identify the recording profile where GIR and GIA destinations are configured.
2. Comment out the following parameters in the **gvp.service-parameters** folder, which are currently pointed at GIA:
    - `service-parameters.httpauthorization2`
    - `service-parameters.recdest2`
    - `service-parameters.type2`

3. Create instances of the three parameters in the recording IVR profile and configure them to point at EXEC services as described in the third-party recorder configuration.

Storing Only Engage Sync-Scope Recordings in WebDAV

See Storing Only Engage Sync-Scope Recordings in WebDAV for the description of this type of EXEC integration.

> Important
>
> For more information on configuration related to sync-scope recordings, please reach out to Genesys Customer Care.

Configuring .env file

The **.env** file contains the description of the EXEC docker environment. Configure all mandatory parameters to describe the environment where EXEC components are planned to be deployed. Bootstrap script converts data from this file into the docker compose yml configuration files.

You can specify parameters for the deployment by overriding the default values in the .env file. See the Parameters table for a full list of overridable values.

Common Configuration

EXCP Parameters

Parameter Name	Description
EXCP_RECORDING_PROCESSOR_PROXY_HOST*	HTTP Load Balancer Proxy's IP address or FQDN which proxies the request to EXRP instances.
EXCP_RECORDING_PROCESSOR_PROXY_PORT*	HTTP Load Balancer Proxy's Port which proxies the request to EXRP instances.
A ***** indicates mandatory fields.	

EXRP Parameters

Parameter Name	Description
EXRP_VM_HOST_LIST *	Comma Separated VM IP Address list where

Parameter Name	Description
	Recording Provider service will be running.
EXRP_PORT *	Recording Provider Service Port
EXRP_TAG *	Recording Provider container version
EXRP_LOG_LEVEL	Defines the EXRP log level. Values: trace, debug, info, warn, error, fatal
EXRP_CONFIG_SERVER_USER *	Config Server Username
EXRP_CONFIG_SERVER_PASSWORD *	Config Server Password
EXRP_CONFIG_SERVER_APPLICATION *	Config Server Application
EXRP_CONFIG_SERVER_PRIMARY_HOST *	IP Address of Config Server
EXRP_CONFIG_SERVER_PRIMARY_PORT*	IP Port of Config Port
A * indicates mandatory fields.	

For third-party recorders, no additional configuration is required.

For GIR Integration, configure the GIR RWS URL in EXCP using the environment variable **EXCP_GIR_HOST**.

Parameter Name	Description
EXCP_GIR_HOST*	Interaction Recording Web service *FQDN/host:port*
A ***** indicates mandatory fields.	

Deploying EXRP service

To deploy the EXRP service, run the procedure described in the Deploying EXEC section.

Starting EXEC services

1. Start the EXRP Service by running `./bootstrap.py --action start --service exrp`.
2. Check the health of EXRP Service and once it shows as healthy then it signifies that EXRP is running fine. Repeat the same for all the EXRP service instances.

```
curl -v http://<EXRP_VMHOST_IPADDRESS>:8080/health
 200 OK with dependency status
```

3. Re-start the EXCP Service by running `./bootstrap.py --action restart --service excp`.
4. Check if the EXCP Service is healthy and ready to support the recordings by running below command:

```
curl -X POST http://<EXCP_VMHOST_IPADDRESS>:3650/api/v1/recording
 {"status":{"code":40402,"message":"CallUUID not found"}}
```

5. Start the NGINX with the Configurations for EXCP and EXRP.
6. Check whether NGINX properly forwards requests to both EXCP and EXRP properly.

```
curl -X POST http://<NGINX_IPADDRESS>:<EXCP_PORT>/api/v1/recording
 {"status":{"code":40402,"message":"CallUUID not
 found","corrId":"2fdd55fe-8b16-41ab-84c9-2c5cc42d4eee"}}
```

```
    curl http://<NGNIX_IPADDRESS>:<EXRP_PORT>/health
    HTTP Status Code = 200
```