Genesys Interaction Recording Solution Guide

Deploying Web Services and Applications for GIR

4/2/2025

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 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	Y	Specifies whether to allow Web Services to schedule purge and backup events.	Boolean	`True`
createCallRecordingCF	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.
defaultBackupExportURI	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		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		Specifies the maximum TCP	Integer	10 * **recordCryptoServerDecryptMa**

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	No	Specifies the socket timeout, in milliseconds, for TCP connections to 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.	Integer	30000
webDAVMaxConnection	No	Specifies the maximum TCP connections for each WebDAV Storage.	Integer	50
webDAVMaxTotalConnection	No	Specifies the maximum TCP connections the Web Services node allows to all WebDAV Storages.	Integer	10 * **webDAVMaxConnection**
undocumentedExternalApiUrl	No	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.