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 Security Deployment Guide

Genesys Voice Platform Support for GDPR

5/6/2025

Genesys Voice Platform Support for GDPR

This page describes product-specific aspects of Genesys Voice Platform support for the European Union's General Data Protection Regulation (GDPR) in premise deployments. For general information about Genesys support for GDPR compliance, see General Data Protection Regulation.

> **Warning**
>
> Disclaimer: The information contained here is not considered final. This document will be updated with additional technical information.

Data Retention Policies

GVP has configurable retention policies that allow expiration of data. GVP allows aggregating data for items like peak and call volume reporting. The aggregated data is anonymous. Detailed call detail records include DNIS and ANI data. The Voice Application Reporter (VAR) data could potentially have personal data, and would have to be deleted when requested. The log data files would have sensitive information (possibly masked), but requires the data to be rotated/expired frequently to meet the needs of GDPR.

Configuration Settings

Media Server

Media Server is capable of storing data and sending alarms which can potentially contain sensitive information, but by default, the data will typically be automatically cleansed (by the log rollover process) within 30 days.

The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- recordutterance.path = $InstallationRoot$/utterance/
- recording.basepath = $InstallationRoot$/record/
- record.basepath = $InstallationRoot$/record
- cpd.record.basepath = $InstallationRoot$/record/
- record.basepath = $InstallationRoot$
- record.irrecoverablerecordpostdir = $InstallationRoot$/cache/record/failed
- recordcachedir = $InstallationRoot$/cache/record

- directory = $InstallationRoot$/callrec/

Log files and temporary files can be saved. The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- logdir = $InstallationRoot$/logs/
- tmpdir = $InstallationRoot$/tmp/
- directories.save_tempfiles = $InstallationRoot$/tmp/

Also, additional sinks are available where alarms and potentially sensitive information can be captured. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information. The location of these files can be configured in the GVP Media Control Platform Configuration [default paths are shown below]:

- ems.log_sinks = MFSINK|DATAC|TRAPSINK
- metricsconfig.DATAC = *
- dc.default.metricsfilter = 0-16,18,25,35,36,41,52-55,74,128,136-141
- ems.metricsconfig.MFSINK = 0-16,18-41,43,52-56,72-74,76-81,127-129,130,132-141,146-152

GVP Resource Manager

Resource Manager is capable of storing data and sending alarms and potentially sensitive information, but by default, the data will typically be automatically cleansed (by the log rollover process) within 30 days.

Customers are advised to understand the GVP logging (for all components) and understand the sinks (destinations) for information which the platform can potentially capture. See **Table 6** and **Appendix H** of the Genesys Voice Platform User's Guide for more information.

GVP Reporting Server

The Reporting Server is capable of storing/sending alarms and potentially sensitive information, but by default, these components process but do not store consumer PII. Customers are advised to understand the GVP logging (for all components) and understand the sinks (destinations) for information which the platform can potentially capture. See Table 6 and Appendix H of the Genesys Voice Platform User's Guide for more information.

By default, Reporting Server is designed to collect statistics and other user information. Retention period of this information is configurable, with most data stored for less than 30 days. Customers should work with their application designers to understand what information is captured as part of the application, and, whether or not the data could be considered sensitive.

Data Retention Specific Settings

- rs.db.retention.operations.30min.default = 7
- rs.db.retention.operations.5min.default = 1

- rs.db.retention.operations.counts.default = 7
- rs.db.retention.operations.daily.default = 90
- rs.db.retention.operations.hourly.default = 7
- rs.db.retention.operations.monthly.default = 1095
- rs.db.retention.operations.weekly.default = 364
- rs.db.retention.cdr.default = 30
- rs.db.retention.events.default = 7
- rs.db.retention.var.30min.default = 7
- rs.db.retention.var.5min.default = 1
- rs.db.retention.var.daily.default = 90
- rs.db.retention.var.hourly.default = 7
- rs.db.retention.var.monthly.default = 1095
- rs.db.retention.var.weekly.default = 364

Identifying Sensitive Information for Processing

The following example demonstrates how to find this information in the Reporting Server database – for the example where ‘Session_ID’ is considered sensitive:

- `select * from dbo.CUSTOM_VARS where session_ID = '018401A9-100052D6';`
- `select * from dbo.VAR_CDRS where session_ID = '018401A9-100052D6';`
- `select * from dbo.EVENT_LOGS where session_ID = '018401A9-100052D6';`
- `select * from dbo.MCP_CDR where session_ID = '018401A9-100052D6';`
- `select * from dbo.MCP_CDR_EXT where session_ID = '018401A9-100052D6';`

An example of a SQL query which might be used to understand if specific information is sensitive:

[+] View Example Query

```
USE [ems-rs]
DECLARE @SearchStr nvarchar(100) = '018401A9-100052D6'
DECLARE @Results TABLE (ColumnName nvarchar(370), ColumnValue nvarchar(3630))

SET NOCOUNT ON

DECLARE @TableName nvarchar(256), @ColumnName nvarchar(128), @SearchStr2 nvarchar(110)
SET  @TableName = ''
SET @SearchStr2 = QUOTENAME('%' + @SearchStr + '%','''')

WHILE @TableName IS NOT NULL

BEGIN
    SET @ColumnName = ''
    SET @TableName =
    (
        SELECT MIN(QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME))
        FROM     INFORMATION_SCHEMA.TABLES
        WHERE         TABLE_TYPE = 'BASE TABLE'
            AND    QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME) > @TableName
            AND    OBJECTPROPERTY(
                    OBJECT_ID(
                        QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME)
                         ), 'IsMSShipped'
                           ) = 0
    )

    WHILE (@TableName IS NOT NULL) AND (@ColumnName IS NOT NULL)

    BEGIN
        SET @ColumnName =
        (
            SELECT MIN(QUOTENAME(COLUMN_NAME))
            FROM     INFORMATION_SCHEMA.COLUMNS
            WHERE         TABLE_SCHEMA    = PARSENAME(@TableName, 2)
                AND    TABLE_NAME    = PARSENAME(@TableName, 1)
                AND    DATA_TYPE IN ('char', 'varchar', 'nchar', 'nvarchar', 'int', 'decimal')
                AND    QUOTENAME(COLUMN_NAME) > @ColumnName
        )

        IF @ColumnName IS NOT NULL

        BEGIN
            INSERT INTO @Results
            EXEC
            (
                'SELECT ''' + @TableName + '.' + @ColumnName + ''', LEFT(' + @ColumnName + ',
3630)
                FROM ' + @TableName + ' (NOLOCK) ' +
                ' WHERE ' + @ColumnName + ' LIKE ' + @SearchStr2
            )
        END
    END
END

SELECT ColumnName, ColumnValue FROM @Results
```