Genesys Info Mart Deployment Guide

Exporting data from Genesys Info Mart

3/14/2025

Contents

Exporting data from Genesys Info Mart

> ## Important
>
> Starting with release 8.5.011.22, Genesys Info Mart provides on-premises support for the Data Export feature that uses Job_ExportGIM and, optionally, export views to export data into local .csv files, so that the data is available for further import into a data warehouse. If you want to use the Data Export feature, the information on this page is out of date. For full information about the Data Export feature, see the "About Data Export" page in the *Genesys Info Mart Physical Data Model* for your RDBMS.

You can export data from Genesys Info Mart to use in your own data warehouse. Prior to exporting data from Genesys Info Mart, use the information on this page to ensure that facts are up-to-date (if you are exporting non-aggregated data) or that aggregation is complete (if you are exporting aggregated data).

In order to export data, you must have direct access to the Info Mart database. For more information about accessing the Info Mart database, contact your administrator. For information about the Info Mart database schema, see the Genesys Info Mart 8.5 Reference Manual for your RDBMS type, which is available from docs.genesys.com. For information about the organization of aggregate tables, see *Reporting and Analytics Aggregates Reference Manual*.

> ## Warning
>
> This page describes actions that, if improperly performed, could negatively impact the operation of your database. Consult your database administrator before making any changes.

Tips for extracting Genesys Info Mart data

To incrementally export data (export only newly inserted or updated rows), you can check the audit key fields in Info Mart fact and dimension tables (CREATE_AUDIT_KEY, UPDATE_AUDIT_KEY). For example, on MSSQL deployments, select only newly inserted or updated rows by using logic similar to the following:

```
declare @MAX_AUDIT_KEY      numeric(19);
declare @PREV_MAX_AUDIT_KEY numeric(19);

SET @PREV_MAX_AUDIT_KEY = (SELECT max(AUDIT_KEY) FROM 'a table with high water mark of
extracted data'
WHERE TABLE_NAME='INTERACTION_FACT';

SET @MAX_AUDIT_KEY = (SELECT MAX(AUDIT_KEY) FROM CTL_AUDIT_LOG WITH (INDEX(PK_CTL_AUDIT_LOG))
WHERE JOB_ID NOT IN (
```

```
    SELECT JOB_ID FROM CTL_WORKFLOW_STATUS
    WHERE STATUS IN ('RUNNING'))
and AUDIT_KEY > @PREV_MAX_AUDIT_KEY);

select * into INTERACTION_FACT_NEW from INTERACTION_FACT
where  CREATE_AUDIT_KEY between @PREV_MAX_AUDIT_KEY+1 and @MAX_AUDIT_KEY
or     UPDATE_AUDIT_KEY between @PREV_MAX_AUDIT_KEY+1 and @MAX_AUDIT_KEY;

update 'a table with high water mark of extracted data'
set AUDIT_KEY=@MAX_AUDIT_KEY
where TABLE_NAME='INTERACTION_FACT';
```

Tips for extracting Reporting and Analytics Aggregates data

While facts are recorded promptly after calls terminate, aggregate tables are updated on a periodic basis. Before you export aggregated data, ensure that aggregation is complete before you select the data to export, using logic similar to the following:

To verify that aggregation is complete:

1. Execute queries with logic similar to one or more of the following examples:

   - For the AG2_AGENT_*, AG2_ID_*, and AG2_QUEUE_* aggregates:

     ```
     select @start=min(start_date_time_key), @end=max(start_date_time_key)
      from INTERACTION_FACT
      where CREATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY
      or UPDATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY;
     ```

   - For the AG2_I_AGENT_* aggregates:

     ```
     select @start=min(start_date_time_key), @end=max(end_date_time_key)
      from INTERACTION_FACT
      where CREATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY
      or  UPDATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY;
     ```

   - For the AG2_I_SESS_STATE_*, or AG2_I_STATE_RSN_* aggregates:

     ```
     select @start=min(start_date_time_key), @end=max(end_date_time_key)
      from SM_RES_STATE_FACT
      where CREATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY
      or  UPDATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY;
     ```

   - For the AG2_CAMPAIGN_* aggregates:

     ```
     select @start=min(start_date_time_key), @end=max(start_date_time_key)
      from CAMPAIGN_FACT
      where CREATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY
      or UPDATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY
     ```

   - For the AG2_CALLBACK_* aggregates:

     ```
     select @start=min(start_date_time_key), @end=max(start_date_time_key)
      from CALLBACK_FACT
      where CREATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY
      or  UPDATE_AUDIT_KEY BETWEEN @PREV_MAX_AUDIT_KEY+1 AND @MAX_AUDIT_KEY;
     ```

2. Execute:

```
select 1 from PENDING_AGR
 where START_DATE_TIME_KEY < @end and END_DATE_TIME_KEY > @start
```

If this step returns rows, then this AUDIT_KEY has not been processed by the aggregation engine, and you cannot pull data for it from aggregate tables.

To select aggregated data for export:

3. If step 2 returns no rows, then you can select data for all aggregates:

```
select * from AG2_<view>
 where DATE_TIME_KEY between @start and @end
```

Performance considerations

When creating queries to export data, Genesys recommends that you avoid full-table scans, as this can impact Genesys Info Mart performance.

Script	Description
make_export_indexes_mssql.sql	Example script to create indexes for Info Mart tables. Applies to MSSQL deployments.