



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.

Reporting and Analytics Aggregates User's Guide

What is Aggregation and How Do I Enable It?

What is Aggregation and How Do I Enable It?

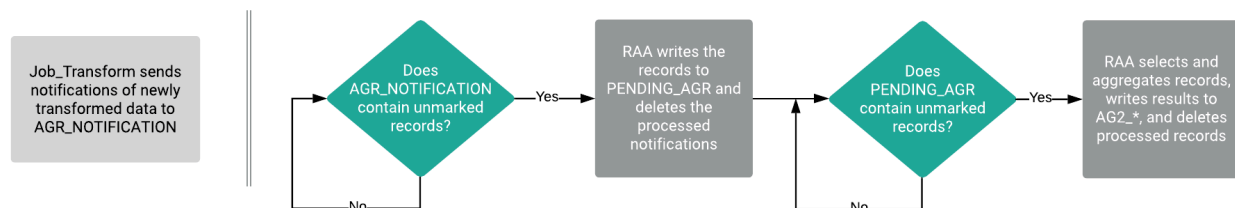
Contents

- **1 What is Aggregation and How Do I Enable It?**
 - 1.1 How Does The Aggregation Process Work?
 - 1.2 How Do I Enable Aggregation?
 - 1.3 In What Order is Data Aggregated?

This page describes how to enable Reporting and Analytics Aggregates (RAA) and how it interfaces with the Genesys Info Mart Server to aggregate contact center data for reporting purposes. The RAA option of Genesys Info Mart must be installed before aggregation can occur. Unlike most other Genesys products though, RAA is not configured within the Genesys Configuration Server. Refer to the [Reporting and Analytics Aggregates Deployment Guide](#) for installation instructions.

How Does The Aggregation Process Work?

A Genesys Info Mart job, **Job_TransformGIM**, is responsible for sending notifications about newly transformed factual data that is ready for aggregation. If this job is not running, the Genesys Info Mart Server cannot send notifications. As part of its extraction, transformation, and loading (ETL) cycle and before it commits updates, the Genesys Info Mart Server determines the start and end DATE_TIME keys for which the updates apply and sends notification of the updates along with the corresponding time range to an intermediate queue—a table named AGR_NOTIFICATION. After some asynchronous processing, the aggregation interface timestamps and grabs these notifications and writes them to the PENDING_AGR internal queue. The aggregation process is illustrated in the Figure, **The Aggregation Process**.



The Aggregation Process



After processing the notifications, the aggregation engine aggregates low-level details (sourced from Info Mart *_FACT tables) and writes the results to aggregate tables (with the prefix AGT_*) in the Info Mart database, and presented through views with the prefix AG2_*. When the aggregation engine is enabled, it constantly polls both AGR_NOTIFICATION and PENDING_AGR for newly added notifications. In this fashion, aggregated data becomes available for reporting in near-real time.

How Do I Enable Aggregation?

Because RAA is an optional component of Genesys Info Mart 8.5, aggregation is not automatically enabled. To start aggregation, you must enable the aggregation engine, and invoke the aggregation process.

To enable the aggregation engine and cause the Genesys Info Mart Server to start sending notifications to the internal queue, you must assign the aggregation class within the Genesys Info Mart application by setting the aggregation-engine-class-name configuration option (described in the "Disable Aggregation in the GIM Application" section of the [Reporting and Analytics Aggregates](#)

Deployment Guide).

Invoke the aggregation process in one of two modes, which launches the activity that routinely creates, modifies, and updates the aggregation tables:

- **Autonomous mode**—Requiring no direct involvement with the Genesys Info Mart Server. Invoking the aggregation process in Autonomous mode runs the aggregation engine stand-alone, from the command line, without referencing configuration settings of the Genesys Info Mart Application object in Configuration Server or invoking the **Job_AggregateGIM** job (invoking this job is equivalent to operating aggregation in Integrated mode). In fact, the Genesys Info Mart Server need not even be running or sending notifications to the AGR_NOTIFICATION queue—although this mode of operation is not particularly useful for capturing ongoing contact center activity. Operating RAA in Autonomous mode, however, is recommended for those circumstances where you want to reaggregate data (see [Reaggregating Data over a Certain Time Range](#)) or when you want to migrate 7.6 data to 8.5 (described in the [Genesys Migration Guide](#)). To learn how to invoke aggregation in Autonomous mode, see [How Do I Configure Continuous Aggregation?](#).
- **Integrated mode**—Where the Genesys Info Mart Server drives aggregation activity. Operating the aggregation process in Integrated mode relies on Genesys Info Mart internal processes to manage all aspects of aggregation. The aggregation engine is driven by **Job_AggregateGIM**, a job that is managed by the Genesys Info Mart Manager. With respect to aggregation, this console respects the values of aggregation-related configuration options that are defined in the Genesys Info Mart Application object. Other Genesys Info Mart jobs, including **Job_MaintainGIM**, are also managed using Genesys Info Mart Manager; however, this user interface is beyond the scope of this document. Refer to the [Genesys Info Mart Operations Guide](#) to learn how to start, stop, and manage jobs.

In What Order is Data Aggregated?

The 8.5 aggregation engine processes chunks of data according to priority, using a simple distribution algorithm that processes chunks in the order in which they appear in the PENDING_AGR queue (beginning with the oldest data), and without giving priority to any particular hierarchy or aggregation level. (Hierarchies and aggregation levels are described in [What is an Aggregation Hierarchy?](#))

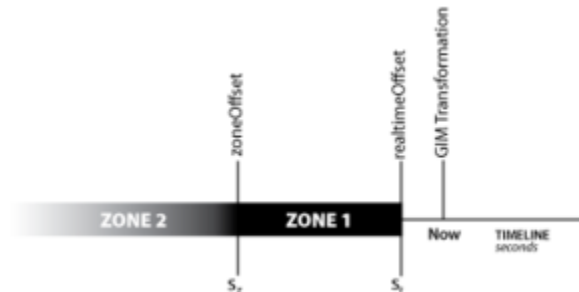
RAA 8.5 determines priority by the time at which RAA registers notifications. Older notifications receive higher priority than newer notifications and are generally processed before newer notifications are processed for a given aggregate set. RAA rounds notification timestamps to nearest 15 minutes. For purposes of priority comparison, these are grouped into larger units. For instance, data about which notifications were registered between 2–4 hours ago are considered to have higher priority than data notifications that were registered between 0–2 hours ago.

RAA 8.5 attributes all data to one of two sliding zones:

- **Zone 1** contains notifications about more recent pending aggregation requests;
- **Zone 2** contains notifications about all other older data.

You configure the boundary that divides the zones by setting the **zone-offset** option in the Genesys Info Mart Application object for aggregation run in Integrated mode, or by issuing the zoneOffset runtime parameter for aggregation run in Autonomous mode. (The [Reporting and Analytics Aggregates Deployment Guide](#) describes both the option and the runtime parameter. [How Do I](#)

Manage the Aggregation Process? describes both modes of running aggregation.) Use the **realtime-offset** option (**realtimeOffset** runtime parameter) to delay RAA processing of notifications that are received up to two hours after Genesys Info Mart transformation. The Figure **Two Zones Contain All Notifications** illustrates the two zones and their boundaries.



Two Zones Contain All Notifications

RAA 8.5 is able to process data concurrently in both zones. Data within each zone is processed in order of priority.

The number of writers allocated across the two zones establishes which zone is dominant and which is recessive. When you assign more writers to Zone 1, it becomes the dominant zone and Zone 2 becomes the recessive zone. When there is an idle writer (one that has just finished processing a chunk of data, for example), RAA first tries to allocate it to the dominant zone. If the allocation fails, RAA allocates the idle writer to the recessive zone. Depending on the degree of pliability (strict or flex), RAA will borrow writers, as necessary, between zones in order to handle the workload of the zone with the higher resource demands. The **writer-schedule** configuration option (**writerSchedule** runtime parameter) defines how many writers are initially allocated to each zone for a particular set of hours. Refer to **How Do I Control Aggregation at Runtime?** for more information.

For any chunk of data, the aggregation engine performs aggregations first for the lowest table node of the aggregation hierarchy—the *_SUBHR or *_HOUR level depending on the model (for a discussion of models, see **What is an Aggregation Hierarchy?**)—then moves up the line to the highest table node of the hierarchy—the *_MONTH level. As aggregation gets propagated up a particular aggregation hierarchy, all aggregation levels within that hierarchy inherit the priority of the original notification. The higher nodes (quarter and year levels) are views that are based on the *_MONTH tables, so no further aggregation processing is necessary beyond the month level. When the aggregation process completes aggregating data at certain stages, it deletes the corresponding row(s) from the PENDING_AGR table as part of the transaction and then commits the transaction.

Note that data for all aggregation levels might not be available at the same point in time. The aggregation engine might propagate data to higher aggregation nodes in different transactions. For a short period of time, it is possible for users to be able to retrieve data from one aggregation level that is not available at the higher nodes. This momentary discrepancy is most evident with the subhour views and hour tables of the Disposition-Based Measures model when newly transformed data has not yet been aggregated. The subhour views retrieve data directly from the source Genesys Info Mart *_FACT tables and immediately reflect this newly transformed data in detail-level reports. Minutes could pass, however, before this data would be reflected in the hour and higher level aggregate tables.

Also playing a role to a lesser extent, RAA might process more than one data chunk simultaneously depending on how many threads it is instructed to use. This instruction comes from the value of the **write-schedule** configuration option or the **writerSchedule** runtime parameter.

When are Notifications Sent?

For voice interactions, the Genesys Info Mart Server sends notifications about completed interactions only. Stuck calls are not eligible for aggregation until they are cleared from queue; Genesys Info Mart does not extract from ICON information about active, in-process calls. For multimedia interactions that originate from an Interaction Server, **Job_TransformGIM** transmits notifications about active interactions as well as completed interactions.