Reporting and Analytics Aggregates Deployment Guide

How Do I Control Aggregation at Runtime?

2/20/2025

How Do I Control Aggregation at Runtime?

In addition to the configuration options that you can set within the Genesys Info Mart Application object, you can also tailor how aggregation works by specifying certain *runtime parameters* when invoking the aggregation engine manually or from a batch or shell script.

Invoking Aggregation From the Command Line

Invoke the aggregation engine from the command line using this format:

- on UNIX platforms: `java -jar agg/GIMAgg.jar -user=<dbo> -pass=<PASSWORD> -jdbcurl=<URL> <OTHERPARAMS>`
- on Windows platforms: `java.exe -jar agg\GIMAgg.jar -user=<dbo> -pass=<PASSWORD> -jdbcurl=<URL> <OTHERPARAMS>`

where <OTHERPARAMS> are one or more of the following optional parameters:

```
-agg_level_<level>_delay=<SECONDS>
-conf paramFile
-insertPendingAggRaw <TENANT>:<SET>:<STARTKEY>:<ENDKEY>
-insertPendingAggRaw <TENANT>:<SET>:DATES_FROM:<FACT_TABLE>
-insertPendingAgg <AGR_SET>:<START>:<END>
-insertPendingAgg <AGR_SET>:DATES_FROM:<FACT_TABLE>
-deadlockThreshold 1000
-defaultTZoffsets offsetFromUTC
-delFeature=eServicesSM, excludeConsult, or materialize-subhour-in-db
-getFeatures
-levelOfLog level
-printQuery aggQuery
-printQuery ALL
-realtimeOffset numOfSec
-setFeature=eServicesSM, excludeConsult, or materialize-subhour-in-db
-subHourInterval 15minOr30min
-updateAliases AliasFile
-writerSchedule default=p(a:b)[,hour(HH-HH)=p(c:d)]
        [,hour(HH-HH)=p(e:f)]
-zoneOffset numOfSec
```

A hyphen (–) precedes each parameter, as shown in the preceding syntax. The options use camel case and are optional unless otherwise specified. Refer to the “Runtime Parameters” section below for a description of each parameter and its permissible values. Invoking aggregation is discussed further in the *Reporting and Analytics Aggregates User’s Guide*. Reporting and Analytics Aggregates (RAA) recognizes additional parameters that are reserved for use by the aggregation migration utility (which is described in the *Genesys Migration Guide*). These reserved parameters are for internal use only and are not described in the documentation.

> **Important**
> The values of the runtime parameters that are described on this page supersede the values of comparable configuration options that are set in the Genesys Info Mart Application object.

RAA Aggregation Runtime Parameters

Most of the parameters listed in the table below are options that apply to aggregation in autonomous mode. However, when you invoke the following parameters, aggregation does not run, and the function associated with the parameter is performed instead:

- **insertPendingAggRaw**
- **insertPendingAgg**
- **setFeature**
- **delFeature**
- **printQuery**
- **updateAliases**

You can invoke these parameters only one at a time; to invoke more than one, you must issue the aggregation command more than once.

> **Important**
> The values of the runtime parameters that are described on this page supersede the values of comparable configuration options that are set within the Genesys Info Mart Application object in Configuration Server.

Runtime Parameter	Description
-agg-jdbc-url	Specifies the connection parameters for the JDBC connection to the Info Mart database for Reporting & Analytics Aggregates (RAA). More information: agg-jdbc-url
agg_level_<level>_delay	This parameter limits the frequency with which RAA performs aggregation. This parameter applies to materialized levels only (DAY and higher).

Runtime Parameter	Description

	Examples:
	• To perform MONTH-level aggregation no more than twice a day, enter either of the following strings:
	`-agg_level_month_delay=60*60*12`
	`-agg_level_month_delay=43200`
	• To perform DAY-level aggregation no more than every 3 hours, enter either of the following strings:
	`-agg_level_day_delay=10800`
	`-agg_level_day_delay=60*60*3`
	The value you specify represents a *minimum* period of time (in seconds) that must elapse between aggregation runs for data at a specified level, however, the actual time between aggregation runs is usually greater than the specified delay value. The exact time when aggregation occurs varies depending both on the time when the first aggregation run occurred, and on the Zone in which the data falls:
	• Data in Zone 1 is delayed up to an additional 15 minutes.
	• Data in Zone 2 is delayed up to an additional hour.
	The time when aggregation occurs may also vary if the aggregation job is inserted using `-insertPendingAgg`.
	For additional information refer to the description of the **agg-level-<LEVEL>-delay** option on the How Do I Configure Genesys Info Mart for Aggregation? page.

conf	Specifies the file name from which the aggregation engine reads parameter values that are not specified at the command line. Precede this file with a relative path or absolute path, if the file is not located in the same directory as the aggregation jar archive. If this parameter or file is missing, the aggregation engine uses only those parameters that are issued at the command line—for example: `java -jar agg/GIMAgg.jar -conf runagg`
defaultTZoffsets	Specifies the winter and summer UTC offset, in seconds, of the time zone of the DATE_TIME table for environments whose offsets are in increments other than one hour—that is, whose offset is not evenly divisible by 3600, and that configure more than one time zone in aggregation. For example: A time-zone offset of six and a half hours

Runtime Parameter	Description
	(UTC+06:30) with recognition of daylight saving time in the summer of one hour would be indicated as follows: `java -jar agg/GIMAgg.jar -conf runagg -defaultTZoffsets 23400,27000`
deadlockThreshold	Specifies the time, in seconds, in which each aggregation writer thread must return the results of its aggregation of a batch of data. If a writer does not respond within this time frame, RAA assumes either that the process is deadlocked or that the database is too slow and cannot process aggregation in a timely fashion. When the deadlock time period has elapsed, RAA cancels all database queries and closes all sessions. To resume processing, aggregation must be restarted. Refer to the description of the **deadlock-threshold** option (see What Options Can I Configure in the [agg] Section?) for additional information.
delFeature=eServicesSM	Stops aggregation of social-media data. To restart aggregation of social-media data, use the **setFeature** runtime parameter.
delFeature=materialize-subhour-in-db	Instructs RAA not to materialize RAA subhour views as tables. To materialize the subhour level, see the **setFeature=materialize-subhour-in-db** option.
delFeature=excludeConsult	Includes consult interactions in ACC_* and ABN_* queue aggregates. To exclude consult interactions, see the *setFeature=excludeConsult* option.
getFeatures	Submits a request to display a list of the features currently enabled in the database.

insertPendingAgg	Submits a request to run the aggregation engine over the specified time period at some later time. The following formats are supported:
	
	`-insertPendingAgg <AGR_SET>:<START>:<END>`
	
	OR
	
	`-insertPendingAgg <AGR_SET>:DATES_FROM:<FACT_TABLE>`
	
	Where:
	
	• <AGR_SET> indicates what set to aggregate (ALLSETS, or an aggregate set name). Aggregate set name is formatted as follows:
	
	`<HIERARCHY_NAME>-`

Runtime Parameter	Description

	`<AGG_LEVEL>[.Flavor].`
	
	Where:
	
	• `<HIERARCHY_NAME>` is the name of the hierarchy to be aggregated.
	• `<AGG_LEVEL>` is the aggregation level (SUBHOUR, HOUR, DAY, MONTH, QUARTER, YEAR).
	• `[.Flavor]` indicates what data to include (Online or Offline).
	
	• <START> is a value in the format YYYY-MM-DD
	• `<END>` is a value in the format YYYY-MM-DD
	• <FACT TABLE> is the name of the fact table from which to retrieve start and end time values. The start and end values are retrieved from the MIN and MAX values of the START_DATE_TIME_KEY field in the specified fact table.
	
	For example:
	
	`... -insertPendingAgg ALLSETS:2014-01-01:2014-12-31`
	
	Refer to the description in the *Reporting and Analytics Aggregates User's Guide* for more information, and for information about performing re-aggregation. Notes:
	
	• A request to re-aggregate data for a specific time range first deletes aggregated data from that time range (to prevent duplicate data from being written to Info Mart). Before you issue a re-aggregation command, make sure that facts for your selected time range exist in the Info Mart database and have not been purged. Otherwise, you could be left with no aggregates at all for that time range.
	• When RAA is running in integrated mode, re-aggregation is possible only if Job_AggregateGIM is already running. If you attempt to pass re-aggregation job parameters when Job_AggregateGIM is not running, Genesys Info Mart simply starts aggregation, ignoring the job parameters. In this case, you can re-aggregate by issuing the command, with the re-aggregation parameters, a second time.

insertPendingAggRaw	Submits a request to run the aggregation engine over the specified time period at some later time. The following formats are supported:

Runtime Parameter	Description

	`-insertPendingAggRaw <TENANT>:<SET>:DATES_FROM:<FACT TABLE>`
	
	OR
	
	`-insertPendingAggRaw <TENANT>:<SET>:<STARTKEY>:<ENDKEY>`
	
	where:
	
	• <TENANT> is a TENANT_KEY value from TENANT table that indicates a tenant (or ALLTENANTS value).
	• <SET> is a AGR_SET_KEY value from AGR_SET table or (ALLSETS value).
	• <STARTKEY> is a DATE_TIME_KEY value from the DATE_TIME table that indicates the beginning of the reporting interval.
	• <ENDKEY> is a DATE_TIME_KEY value from the DATE_TIME table that indicates the end of the reporting interval.
	• <FACT TABLE> is the name of the fact table from which to retrieve start and end time values. The start and end values are retrieved from the MIN and MAX values of the START_DATE_TIME_KEY field in the specified fact table.
	
	For example:
	
	`-insertPendingAggRaw ALLTENANTS:ALLSETS:1256084100:1259748000`
	
	Refer to the *Reporting and Analytics Aggregates User's Guide* for examples of how to determine start-time and end-time keys.

jdbcurl	Specifies the string that is sent to the JDBC driver to indicate the database that the Genesys Info Mart server is to use. Format this required parameter as follows:
	
	For PostgreSQL:
	
	`-jdbcurl=jdbc:postgresql://<dbhost>:<dbport>/<dbname>`
	
	For Oracle:
	
	`-jdbcurl=oracle:thin:@<dbhost>:<dbport>:<SID>`
	
	For Microsoft SQL Server:
	
	`-jdbcurl=jdbc:jtds:sqlserver://<dbhost>:<dbport>;DatabaseName`

Runtime Parameter	Description
levelOfLog	Specifies the detailed level of log messages that the GIM Server generates for aggregation-related activity. For example: `... -levelOfLog=.:INFO` Refer to the description of the **level-of-log** option (see What Options Can I Configure in the [agg] Section?) for additional information.
pass	The unencrypted password of the database owner. Format this required parameter as follows: `java -jar agg/GIMAgg.jar -user=Administrator -pass=$y5t3m`

printQuery	Displays or prints the SELECT statement of the specified aggregation query, based solely on its definition within internal RAA files.
	For example:
	• To display the SQL statement for the QUEUE_ACC_ AGENT query:
	`java -jar ./agg/GIMAgg.jar -printQuery QUEUE_ACC_AGENT`
	• To print the statement of all existing queries to a file:
	`java -jar ./agg/GIMAgg.jar -printQuery ALL> logfile.sql`
	Because this parameter grabs the query's definition entirely from internal files, it does not require that you specify database-connectivity parameters. Refer to the *Reporting and Analytics Aggregates User's Guide* for the names of other RAA queries.

realtimeOffset	Specifies the number of seconds that the upper boundary of Zone 1 is offset from aggregation. For example: `... -realtimeOffset 7200` Use this runtime parameter in conjunction with the `realtimeOffset` and `writerSchedule` parameters to fine-tune aggregation dispatching. Refer to the description of the **realtime-offset** configuration option (see What Options Can I Configure in the [agg] Section?) for the default and valid values of this parameter as well as an explanation as to why you should define a real-time offset.
setFeature=eServicesSM	Maps IRF_USER_DATA_KEYS.GEN_ES_KEY to USER_DATA_KEY1 in the H_ID, H_AGENT, H_AGENT_GRP, and H_AGENT_QUEUE hierarchies, and turns on aggregation of the following social-media measures, based on user data stored in the USER_DATA_GEN_ES and USER_DATA_GEN_ESF tables:

Runtime Parameter	Description
	• INFLUENCE • INFLUENCE_ENTERED • INFLUENCE_OFFERED • ACTIONABILITY • ACTIONABILITY_ENTERED • ACTIONABILITY_OFFERED • SENTIMENT • SENTIMENT_ENTERED • SENTIMENT_OFFERED Not all of these measures exist in all of the aforementioned hierarchies. Refer to the *Reporting and Analytics Aggregates Reference Manual* for more information. To instruct RAA to cease aggregating social-media data, issue the `delFeature` runtime parameter.
setFeature=excludeConsult	Excludes consult interactions in ACC_* and ABN_* queue aggregates, causing RAA to count only customer calls (thus mimicking release 8.1.1 behavior).
setFeature=materialize-subhour-in-db	Instructs RAA to materialize RAA subhour views as tables.
subHourInterval	Specifies the lowest time level of aggregation, in minutes, for the AG2_*_SUBHR tables. You must choose a value for this option at deployment and avoid changing it afterwards. The value for this parameter is either `15min` or `30min`—for example: `... -subHourInterval 15min` Refer to the description of the **sub-hour-interval** option (see What Options Can I Configure in the [agg] Section?) for additional information.
updateAliases	Specifies the name of the file that defines which tenants map to which tenant accounts. Using tenant aliases enables automated maintenance of aliases in all configured tenant accounts. Precede this file by a relative path or, if the file is not located in the same directory as the aggregation jar archive, an absolute path. The aggregation module only maintains those tables and views that are related to aggregation. You must run an alias update each time a tenant is added or removed from configuration or the definition of any of the aggregates changes—for example:

Runtime Parameter	Description
	`java -jar agg/GIMAgg.jar -conf runagg -updateAliases AliasFile` The *Reporting and Analytics Aggregates User's Guide* describes how to format the alias mapping file.
user	The account name of the database owner. Format this required parameter as follows: `java -jar agg/GIMAgg.jar -user=Administrator -pass=$y5t3m`

writerSchedule	Sets the schedule for the number of writers that RAA dedicates to the aggregation of notifications received in Zone 1 and Zone 2. The value for this parameter is a schedule of hours, without spaces, that defines writer assignments.
	
	For example:
	
	`... -writerSchedule default=strict(3:5),hour(8-19)=flex(3:5), hour(20-7)=strict(1:7)`
	
	Refer to the description of the **writer-schedule** configuration option (see What Options Can I Configure in the [agg] Section?) for detailed information about the keywords available to you in order to define this parameter and for the definitions of Zone 1 and Zone 2. This parameter also accepts the argument `hour(#-#)=purge`, which enables and schedules purging of aggregate data.
	For example, to schedule all-day aggregation with 3 writers allocated to zone1, and 1 writer allocated to zone2 (with ability to borrow writers from idle zones) and schedules purging to occur between 1am and 2am (in the time zone of the java process (GMT, if RAA is embedded with Genesys Info Mart):
	
	`-writerSchedule default=flex(3:1),hour(1-2)=purge`
	
	For information about configuring purging rules, see the *Reporting and Analytics Aggregates User's Guide*. Prior to release 8.1.102, the number of writer threads dedicated to aggregation was controlled by the **numberOfWriters** configuration option.

zoneOffset	Specifies the length of Zone 1 in seconds. This option also indirectly defines the boundary between Zone 2 and Zone 1. Use this runtime parameter in conjunction with the **realtimeOffset** and **writerSchedule** parameters to fine-tune aggregation dispatching. Refer to the description of the **zone-**

Runtime Parameter	Description
	offset configuration option (see What Options Can I Configure in the [agg] Section?) for the default and valid values of this parameter as well as an explanation as to why you should define a zone offset.