Reporting 7.6

ETL Runtime

User’s Guide

About Genesys

Genesys is the world's leading provider of customer service and contact center software with more than 4,000 customers in 80 countries. Drawing on its more than 20 years of customer service innovation and experience, Genesys is uniquely positioned to help companies bring their people, insights and customer channels together to effectively drive today's customer conversation. Genesys software directs more than 100 million interactions every day—maximizing the value of customer engagement and differentiating the experience by driving personalization and multi-channel customer service and extending customer service across the enterprise to optimize processes and the performance of customer-facing employees. Go to www.genesys.com for more information.

Each product has its own documentation for online viewing at the Genesys Documentation website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System’s Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Customer Care from Genesys

If you have purchased support directly from Genesys, please contact **Genesys Customer Care**. Before contacting Customer Care, please refer to the ***Genesys Care Program Guide*** for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesys.com

Document Version: 76rt_us_etlruntime_01-2014_v7.6.102.00

Table of Contents

Chapter 10

Supplements

Index

Preface

Welcome to the *Reporting 7.6 ETL Runtime User's Guide.* This guide introduces you to the concepts, terminology, and procedures relevant to Historical Reporting.

This document provides a high-level overview of ETL Runtime 7.6 features and functions, together with software-architecture information and deployment-planning materials.

This document is valid only for the 7.6 release of ETL Runtime.

Note: For releases of this document created for other releases of this product, please visit the Genesys Customer Care website, or request the Documentation Library CD, which you can order by e-mail from Genesys Order Management at `orderman@genesys.com`.

This preface provides an overview of this document, identifies the primary audience, introduces document conventions, and lists related reference information:

About ETL Runtime

ETL Runtime is a component of the Data Mart Services and serves both CC Analyzer and CCPulse+ by organizing historical data for contact centers using the Genesys T-Server. This document describes the role that ETL Runtime plays in your Reporting environment, including a discussion of:

- ETL Runtime modules.
- Runtime parameters.
- Options you can set to fine-tune configuration.

- Scheduling ETL Runtime processes.

Intended Audience

This guide, primarily intended for system and database administrators, assumes that you have a basic understanding of the following:

- Computer-telephony integration concepts, scheduling processes, terminology, and applications.
- Relational databases and administration of your specific database management system(s).
- Your own network configuration.
- Basic Microsoft Windows and/or Unix concepts.

Chapter Summaries

In addition to this preface, this document contains the following chapters:

- Chapter 1, "Overview of the Data Mart Services," on page 11, describes the role of ETL Runtime in the Data Mart Services. This chapter also lists the new features introduced in this release.
- Chapter 2, "Fine-Tuning Data Mart Configuration," on page 17, helps you fine-tune ETL Runtime's configuration using the Genesys Configuration Manager.
- Chapter 3, "Common Log Options," on page 25, describes log configuration options that are common to all Genesys server applications including ETL Runtime.
- Chapter 4, "Runtime Parameters," on page 41, describes each parameter and the properties files in which you can use these parameters.
- Chapter 5, "Transformation Module," on page 61, describes how this module functions and how to schedule or manually run a one-time transformation.
- Chapter 6, "Aggregation Module," on page 65, describes how this module functions and how to schedule or manually run an aggregation process.
- Chapter 7, "Configuration Tracking Module," on page 75, describes how ETL Runtime tracks the tenant and object changes that occur in the Configuration Server.
- Chapter 8, "Tenant Alias Tracking Module," on page 79, describes why you should run this module, the algorithm ETL Runtime uses for tenant account names, and how to schedule Tenant Alias Tracking (TAT) processes.

- Chapter 9, "Purging Module," on page 87, describes the ETL Runtime purge algorithm and how you can use ETL Assistant to purge the Data Mart of data that you no longer need.
- Chapter 10, "Application Files," on page 93, describes the files that are deployed during ETL Runtime installation.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to `Techpubs.webadmin@genesys.com`.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Customer Care if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Contacting Genesys Customer Care

If you have purchased support directly from Genesys, please contact Genesys Customer Care.

Before contacting Customer Care, please refer to the *Genesys Care Program Guide* for complete contact information and procedures.

Document Change History

New in Document Version 7.6.102.00

The following topics have been added or have changed significantly since the previous 7.6 release of this document:

- Dates have been corrected in Table 12, "The First Week of A Year," on page 69

New in Document Version 7.6.101.00

This section lists topics that are new or that have changed significantly since the initial 7.2 release of this document:

- Runtime parameters supported for 7.0.2 or prior releases have been removed from Table 8 on page 42.
- Support in the Aggregation Model for ODS data sources populated by high-availability (primary and backup) Data Sourcer applications is referenced on page 66.
- SAP Crystal Reports replaces Hyperion Intelligence Designer and Hyperion Interactive Reporting Studio in this release.
- RG Assistant is no longer supported in this release. References to this component have been removed from this document.

Chapter

1 Overview of the Data Mart Services

The *Reporting 7.6 ETL Assistant Help* file describes ETL Assistant functionality while the *Physical Data Models* describe the table and field structure of the Data Mart database. Both are components of the Reporting Data Mart Services. ETL Runtime is the third component and serves as the primary focus of this document.

This chapter describes the Data Mart Services as it relates to Reporting as a whole:

Where Do the Data Mart Services Fit In?

The Data Mart Services service both CC Analyzer and CCPulse+ to grab historical data from the Data Collection Services and to store and organize the data that the Information Delivery Services requests for each product. Figure 1 shows where the Data Mart Services fit into the overall Reporting picture.

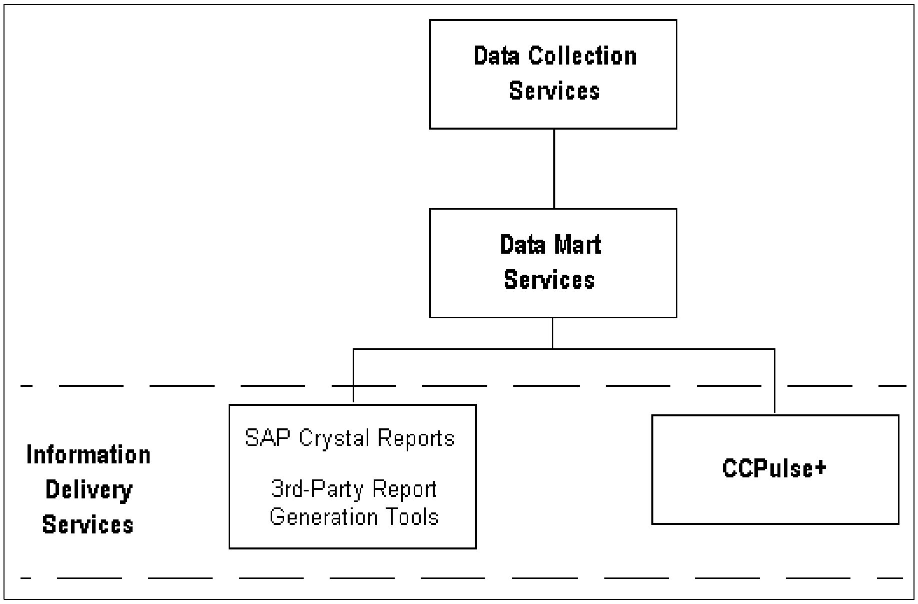


Figure 1: Reporting Services

Component Interrelationships

ETL Runtime is the workhorse of the Data Mart Services as it interacts with the following components to read, write, aggregate, and delete data:

- ODS
- Configuration Server
- ETL Assistant
- Data Mart

Figure 2 illustrates this interconnectivity.

ODS

The Operational Data Storage (ODS) is the final component of the Data Collection Services. ODS holds historical data until ETL Runtime transfers it to the Data Mart using an asynchronous connection. ETL Runtime periodically scans the ODS for any changes to the report layouts that collect data. If it finds a modification, it temporarily suspends all ETL activity on the layout. ETL Runtime then adds or removes the columns from the respective tables, rebuilds all aliases, and automatically resumes all ETL activity without loss of data. Also, if the `dropTransferredTables` ETL Runtime parameter is used (see page 48), ETL Runtime deletes the corresponding tables from ODS after it determines that data transfer successfully completed.

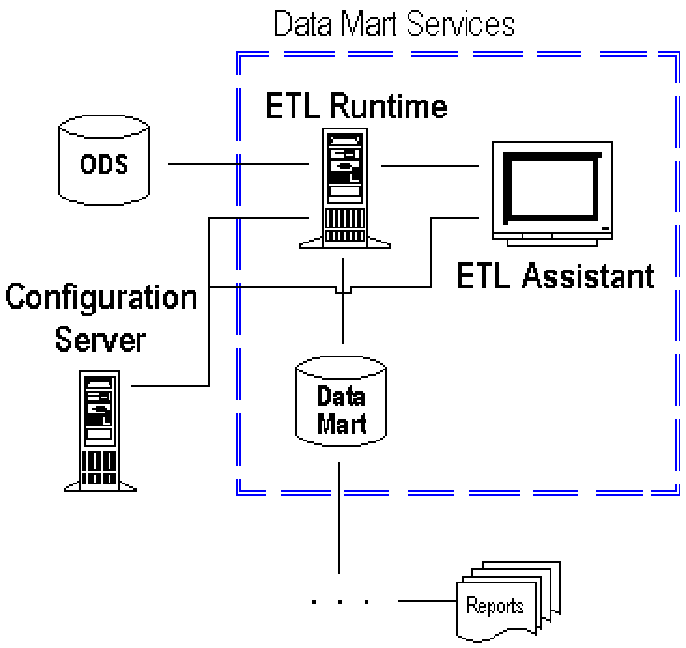


Figure 2: Components of the Data Mart Services

ETL Assistant

ETL Assistant is the front-end tool you use to communicate with the ETL Runtime server. Using ETL Assistant, you do the following:

- Specify which ODSs ETL Runtime is to read data from.
- Define which data ETL Runtime will purge from the Data Mart.
- Discover the names of the Data Mart tables where data has been transferred.
- View any new statistics that have been added to the data collection templates and layouts through Data Modeling Assistant.

Note: Refer to *Reporting 7.6 Data Modeling Assistant Help* for more information on statistics and data collection templates and layouts.

Data Mart

The Data Mart is the database where data is permanently stored for retrieval using report generation tools such as the SAP Crystal Reports, CCPulse+, or other third-party tools. For quick access, data is organized in report folders by aggregation level. Historical Reporting supports four database types for your Data Mart:

- Oracle
- Microsoft SQL

- Sybase
- DB2

For the specific version of each RDBMS supported, refer to the *Reporting Deployment Guide* or the *Genesys Supported Operating Environment Reference Manual.*

Configuration Server

Though Configuration Server is not a component of the Data Mart Services, ETL Runtime does connect to it to do the following:

- Be registered as a Genesys component (all Genesys software is registered in the Configuration Server).
- Track tenant changes made within Configuration Server, such as tenant additions or deletions.
- Track group changes, such as when new members are added to or deleted from a group.

ETL Runtime tracks only those objects, groups, and tenants that you specified for it to monitor (using ETL Assistant).

ETL Runtime

ETL Runtime is a Genesys server that transfers and organizes the data collected by Data Sourcer to the Data Mart using the following modules:

- Transformation Module—detects new data in ODS, transfers it to the Data Mart, and then deletes it from ODS (at your discretion).
- Aggregation Module—aggregates the transferred data into hourly, daily, weekly, monthly, quarterly, and yearly time buckets for each report view for quick retrieval by CCPulse+, SAP Crystal Reports, or other third-party report-generation tool.
- Configuration Tracking Module—tracks the relationship of configuration objects to their respective groups.
- Tenant Alias Tracking Module—tracks the configuration of tenants in the Configuration Server enabling you to use the same queries to generate reports for each tenant in a multi-tenant environment.
- Purging Module—permanently deletes data from the Data Mart.

Genesys ETL Service Starter

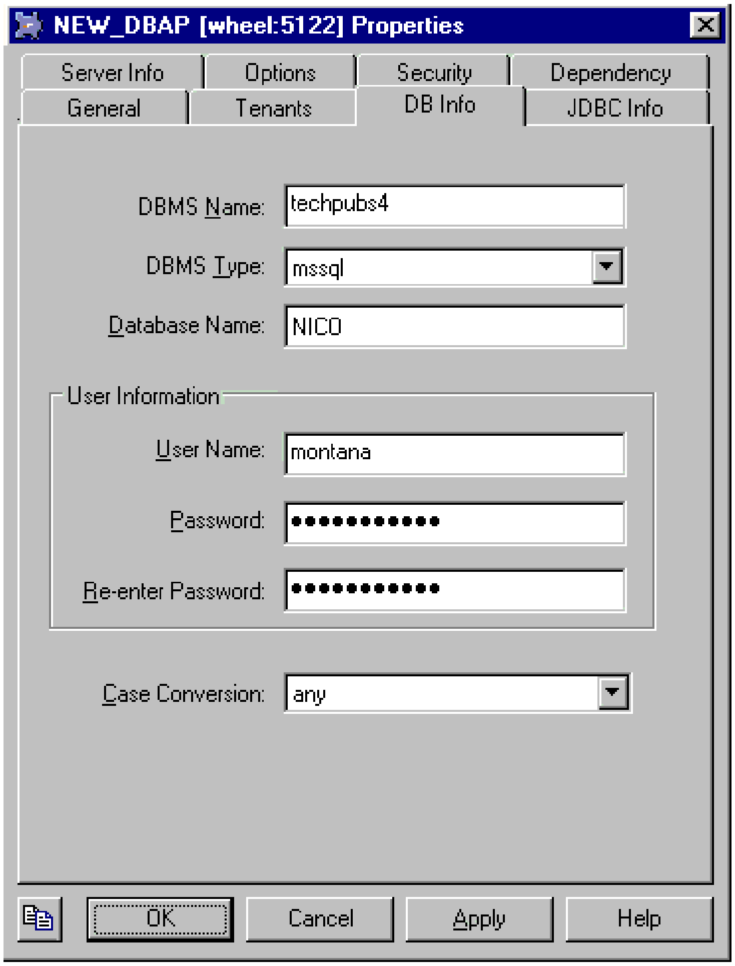
The Historical Reporting Wizard deploys an ETL Service Starter application (or Starter for short), among other applications, in Configuration Manager. When run, Starter schedules the following processes:

- Transformation process

- Aggregation process
- Configuration object tracking

The Transformation process runs continuously until stopped. The other two processes run and then exit when completed. The processes start again according to their next schedule. ETL Runtime obtains:

- Configuration Server connection information from the `etl.properties` file, which the Historical Reporting Wizard configures
- Data Mart connection information from the properties of the corresponding Database Access Point `Application` object in Configuration Server. Figure 3 shows the `DB Info` tab of a sample DB access point where Data Mart connection information is stored.



NEW_DBAP [wheel:5122] Properties

DBMS Name: techpubs4

DBMS Type: mssql

Database Name: NICO

User Information

User Name: montana

Password: ••••••••••••

Re-enter Password: ••••••••••••

Case Conversion: any

Figure 3: Specify Data Mart Connection Parameters at the DB Info Tab

In addition, Starter issues some parameters directly on the command line (instead of reading them from `etl.properties`) including the `setFirstDayOfWeek` parameter (described on page 53). You can find the complete listing of these parameters at the `Start Info` tab of your Starter `Application` object.

You can configure the ETL Service Starter to run other ETL Runtime processes on a daily basis. If, however, you want to schedule processes beyond a daily event (such as every Sunday at 2:00 AM or on the first of each month), you will have to use a third-party scheduler.

Note: The screen shots in this document may differ from those in your environment depending on your release of Configuration Server and whether ETL Runtime is operating in a single- or multi-tenant environment.

Chapter

2 Fine-Tuning Data Mart Configuration

ETL Runtime Application objects are of type Data Mart. Using Configuration Manager, you can modify the application object's configuration after initial configuration and installation to change ETL Runtime behavior. In the following sections, this chapter talks about which configuration options you can set for ETL Runtime, how to modify them, and how to create and schedule new processes:

Options for the Log section are described in a separate chapter, Chapter 3, "Common Log Options", beginning on page 25. You can also impact the behavior of ETL Runtime's modules by the issuance of command-line parameters. These are described in Chapter 4, "Runtime Parameters" beginning on page 41.

Configuration Sections

Three types of sections are valid for a Data Mart Application object:

- One Genesys ETL Service Starter section
- One log section
- One or more process sections

To add, modify, or delete configuration options:

1. Double-click the corresponding Data Mart Application object from the Application folder in Configuration Manager to open its properties.
2. Select the Options tab.

3. Choose the appropriate section and specify your configuration option changes.

Figure 4 shows the selection of the ETL_Tracking process for editing.

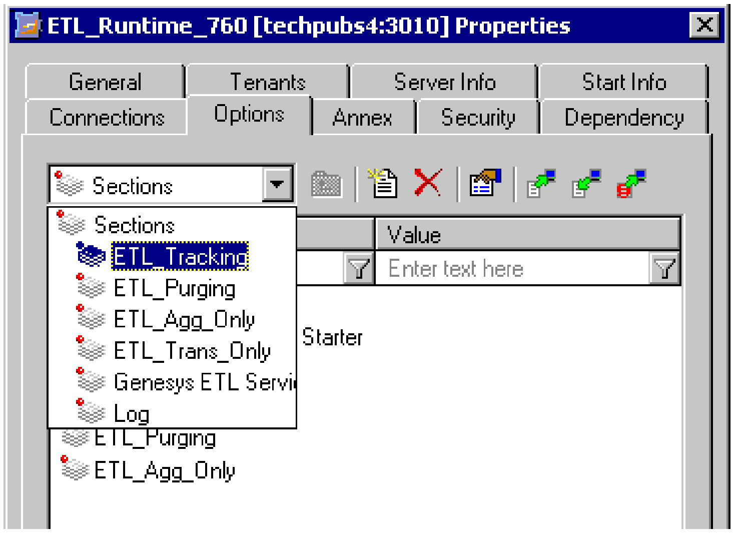


Figure 4: Selecting a Configuration Section

The following sections describe each configuration section and the configuration options it recognizes.

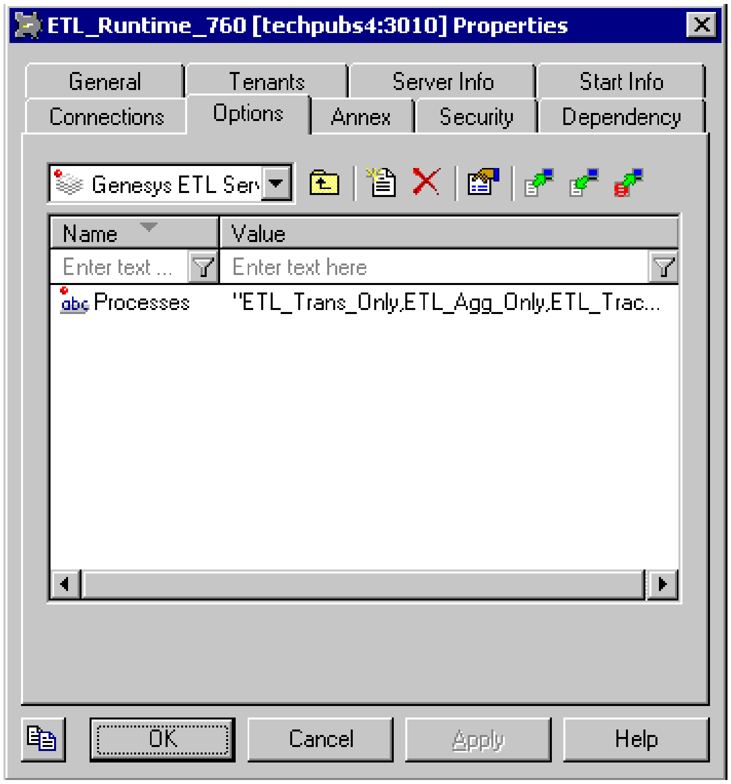
Genesys ETL Service Starter Section

You use the Starter section to schedule the various ETL Runtime modules. This section must be named Genesys ETL Service Starter. A 7.*x* Data Mart installation automatically configures this section for each Data Mart Application object. Table 1 describes the processes configuration option for the Service Starter section of a Data Mart Application object.

Table 1: The Processes Configuration Option

Option	Description
Processes	Specifies a list of the processes to be started. Default Values: ETL_Trans_Only, ETL_Agg_Only, ETL_Tracking Valid Values: a comma-separated list of process names no longer than 255 characters Describe each process in a separate process section of ETL Runtime configuration options.

Figure 5 shows the default setting for the ETL Service Starter section in Configuration Manager.



Processes "ETL_Trans_Only,ETL_Agg_Only,ETL_Trac...

Figure 5: ETL Service Starter Section

If you change the process definition of the ETL Service Starter section (or the definition of any of the processes it starts), you must stop and restart the Starter component for ETL Runtime to recognizes the changes.

[Process] Section

Configuration Server designates ETL Runtime applications as Data Mart Application objects.Within Configuration Manager, you can define one or more process sections to your Data Mart Application object. If you use the Historical Reporting Wizard to configure a Data Mart Application object, and during the configuration you select default options, your Data Mart Application object will contain five process sections:

- ETL_Trans_Only
- ETL_Agg_Only
- ETL_Tracking
- ETL_Purging
- Log

However, you can name (or rename) these sections as you see fit subject to the limitations described on page 21. (Refer to the *Reporting Deployment Guide* for information about the Historical Reporting Wizard.)

Table 2 describes the generic options that are available for any process.

Table 2: ETL [Process] Section Options

Option	Description
Priority	Specifies the priority with which the process will be initiated. Default Value: `medium` Valid Values: `critical`, `high`, `medium`, `low` (case insensitive) Change Takes Effect: Immediately **Note:** Do not enclose values in quotation marks.
Command and Command[0.n]	Specifies the arguments to be executed on the command line of the current operating system from `Starter`'s home directory. Arguments are limited to 255 characters. For cases in which you need more than 255 characters, use `Command0`, `Command1`, `Command2`, and so forth. ETL Runtime reads commands in chronological order: `Command`, then `Command0`, then `Command1`, and so forth. The Configuration Server ignores trailing blank spaces so a command line should end with the start of the next option. Default Value: n/a Valid Values: any string, 255 characters in length or less Change Takes Effect: Immediately **Note:** Do not enclose values in quotation marks.
StartTime	Specifies the schedule by which the process should be started. A new process is not started if one is already started. Default Value: Not applicable Valid Value: String specifying a time profile. No commas permitted. `00:11+1:00`, for example, specifies to start the corresponding ETL process at 11 minutes past midnight and restart it every hour thereafter. Change Takes Effect: Immediately Refer to the *Framework Stat Server User's Guide* for an explanation of how to specify time profiles.

You can view the option definitions for each `process` section by opening the `Properties` dialog box of your Data Mart `Application` object within Configuration Manager. Table 3, for instance, shows the default definition of the `ETL_Agg_Only` process.

Table 3: The ETL_Agg_Only Process

Option	Value
Priority	`medium`
Command	`java -jar transform.jar -conf etl.properties -aggOnly`
StartTime	`0:11+1:00` That is, the aggregation process starts 11 minutes after midnight and every hour thereafter. This parameter follows the same paradigm that is used for Stat Server time profiles. Refer to the "TimeProfiles Section" in the "Statistic Configuration Options" chapter of the *Framework Stat Server User's Guide* for more information and examples.

Refer to page 43 for a description of the `aggOnly` parameter used above as well as other ETL Runtime parameters.

Name Restrictions

Note that process names should not contain blanks or commas. Valid names consist of the characters [`a-z`], [`A-Z`], [`0-9`], _ (underscore) or `-` (hyphen). The maximum length for any one process name is 255 characters, but keep in mind that your `ETL Service Starter` may specify more than one process. `ETL Service Starter`'s full list of processes is confined to 255 characters.

Creating a New Process

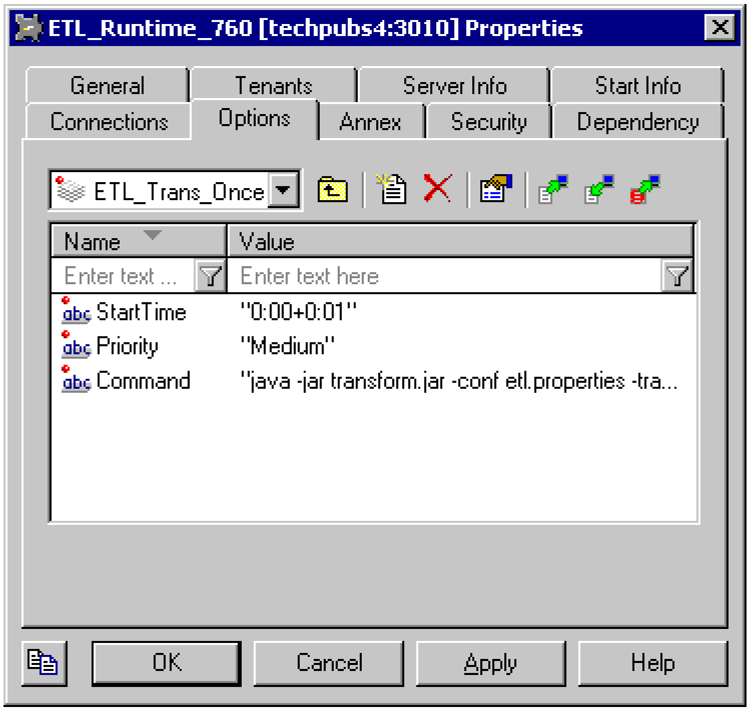
To create a new ETL Runtime `process` section:

1. Edit the properties of your Data Mart `Application` object.
2. Add and name a new `process` section.
3. Add `Priority`, `Command`, and `StartTime` options and specify their values.

Figure 6, for example, shows one way to create an `ETL_Trans_Once` process. (The name you choose for this process is arbitrary). The method is similar to the procedure used for creating the `ETL_Trans_Only` process, but instead places the `transOnce` runtime parameter at the command line, where it does not exist for the `ETL_Trans_Only` process.

Note: Figure 6 does not show the full command. It is:

```
java -jar transform.jar -conf etl.properties -transOnce
```



Name	Value
StartTime	"0:00+0:01"
Priority	"Medium"
Command	"java -jar transform.jar -conf etl.properties -tra...

Figure 6: Specifying Options for a New Process

Warning! Do not simultaneously run both a continuous transformation and a one-time transformation. If you schedule an `ETL_Trans_Once` process, make sure you unschedule the `ETL_Trans_Only` process.

Scheduling Examples

Here are three examples illustrating how you might schedule the ETL Runtime Transformation and Aggregation modules.

Scheduling a Daily, Onetime Transformation

To schedule a one-time ETL transformation occurring nightly at 2:00 AM:

1. Create an `ETL_Trans_Once` process as described above using `StartTime = 2:00`
2. Change the definition of `ETL Service Starter` to replace `ETL_Trans_Only` with `ETL_Trans_Once`.
3. Stop and restart ETL Service Starter for your changes to take affect.

To achieve the same results, you could add the `transOnce` runtime parameter to the `etl.properties` file or modify the `ETL_Trans_Only` process to include `-transOnce` on the command line.

Scheduling a Daily, Onetime Aggregation

To schedule a one-time ETL aggregation following the one-time transformation, change the `StartTime` definition of the `ETL_Agg_Only` process to a value at least one hour later: `StartTime = 4:00, for example.` Stop and restart ETL Service Starter for your changes to take effect.

Scheduling Four, Daily Transformations and Aggregations

To schedule transformation to occur at 5:00 AM and every 6 hours thereafter, with the aggregation process following, you might set `ETL_Trans_Only`'s `StartTime` to `5:00+6:00` and `ETL_Agg_Only`'s `StartTime` to `0:11+6:00`. Notice here that you do not specify a `StartTime` of `6:11+6:00` for aggregation, because then the last aggregation of the day (12:11 AM) would actually fall into the next day. The Scheduler resets at midnight, so the last aggregation would not be performed.

Using an aggregation `StartTime` of `0:11+6:00,` the transformation and aggregation processes run concordantly at (5:00 AM, 6:11 AM), (11:00 AM, 12:11 PM), (5:00 PM, 6:11 PM), and (11:00 PM, 12:11 AM). Optionally, you could define an `ETL_Trans_and_Agg` process and schedule it using ETL Service Starter to replace both the default `ETL_Trans_Only` and `ETL_Agg_Only` processes. This way, ETL Runtime automatically starts aggregation after transformation.

Note: You must assess the time that is required for ETL Runtime to complete data transfer if you run the Transformation module less frequently than the hourly default. In the example in which a one-time aggregation follows a one-time data transfer, the Transformation module should complete its task and exit before aggregation begins. A number of factors, including memory, CPU, the amount of contact center activity recorded by ODS, network activity, and whether other ETL Runtime modules are running in parallel, contribute to the time required. Typically, if you run the Transformation module hourly, transformation takes one hour.

You must use a third-party scheduler to specify a specific day or group of days (for example, every Sunday at 2:00 AM) to run ETL Service Starter or the individual ETL Runtime modules. The level of granularity for which you can schedule ETL Service Starter is one day.

Chapter

3 Common Log Options

This chapter describes log configuration options that are common to all Genesys server applications and applicable to any Framework server component. This chapter includes the following sections:

Mandatory Options

You are not required to configure any common log options to start ETL Runtime.

Log Section

You must call this section `log`. Table 4 lists the log configuration options available to you. Note that to use these options, you must actively set them, either using the Reporting Solution Wizard or manually within Configuration Manager. The default ETL Runtime application template includes only the `verbose` and `all` log options.

Table 4: Log Options

Option	Description
verbose	Determines if a log output is created. If it is, this option specifies the minimum level of log events generated. The log events levels, starting with the highest-priority level, are `Standard`, `Interaction`, `Trace`, and `Debug`. Refer to “Log Output Options” on page 32 for more information. Default Value: `all` (when you create a new application) Valid Values: `all` All log events (that is, log events of `Standard`, `Trace`, `Interaction`, and `Debug` levels) are generated if you set the `debug-level` option in the `ETLRuntime` section to `all`. `debug` The same as `all`. `trace` Log events of the `Trace` and higher levels (that is, log events of `Standard`, `Interaction`, and `Trace` levels) are generated, while log events of the `Debug` level are not generated. `interaction` Log events of the `Interaction` and higher levels (that is, log events of `Standard` and `Interaction` levels) are generated, while log events of the `Trace` and `Debug` levels are not generated. `standard` Log events of the `Standard` level are generated, while log events of the `Interaction`, `Trace`, and `Debug` levels are not generated. `none` Produces no output. Changes Take Effect: Immediately Refer to the *Framework Deployment Guide* or to *Framework Solution Control Interface Help* for more information on the `Standard`, `Trace`, `Interaction`, and `Debug` log levels.
buffering	Turns operating system file buffering on or off. This option applies only to `stderr` and `stdout` output (see page 32). Setting this option to `true` increases output performance. **Note:** When you enable buffering, log messages may appear in the log with a delay. Default Value: `true` Valid Values: `true` Enables buffering `false` Disables buffering Changes Take Effect: Immediately

Table 4: Log Options (Continued)

Option	Description
segment	Specifies if there is a segmentation limit for a log file. If there is, this option sets the unit of measurement along with the maximum size. If the current log segment exceeds the size set by this option, the current file is closed and a new file is created. Default Value: `false` Valid Values: `false` No segmentation allowed. `<number> KB` or `<number>` Sets the maximum segment size in kilobytes. The minimum segment size is 100 KB. `<number> MB` Sets the maximum segment size in megabytes. `<number> hr` Sets the number of hours for the segment to stay open. The minimum number is 1 hour. Changes Take Effect: Immediately
keep-startup-file	Specifies whether a startup segment of the log, containing the initial ETL Runtime configuration, is to be kept. If it is, you can set this option to `true` or to a specific file size. A `true` setting means that the size of the initial segment will be equal to the size of the regular log segment defined by the `segment` option (defined above). ETL Runtime ignores this option if you set the `segment` option to `false`. Default Value: `false` Valid Values: `false` No startup segment of the log is kept. `true` A startup segment of the log is kept. The size of the segment equals the value of the `segment` option. `<number> KB` Sets the maximum size, in kilobytes, for a startup segment of the log. `<number> MB` Sets the maximum size, in megabytes, for a startup segment of the log. Changes Take Effect: After restart

Table 4: Log Options (Continued)

Option	Description
expire	Determines if log files expire. If they do, this option sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. Default Value: false Valid Values: false — No expiration. All generated segments are stored. <number> file or <number> — Sets maximum number of log files to store. Specify a number from 1-100. <number> day — Sets the maximum number of days before log files are deleted. Specify a number from 1-100. Changes Take Effect: Immediately **Note:** If an option's value is set incorrectly—out of the range of valid values— it will be automatically reset to 10.
messagefile	Specifies the file name for application-specific log events. The name must be valid for the operating system on which the application is running. The option value can also contain the absolute path to the application-specific .lms file. Otherwise, ETL Runtime looks for the file in its working directory. Default Value: datamart.lms Valid Value: <string>.lms Changes Take Effect: After ETL Runtime restarts if ETL Runtime finds datamart.lms at startup or immediately if ETL Runtime cannot find this file at startup. **Warning!** If ETL Runtime cannot find its message file upon startup, it will not be able to generate ETL Runtime–specific log events or send them to Message Server.

Table 4: Log Options (Continued)

Option	Description

message_format	Specifies the format of log record headers that an application uses when writing logs in the log file. Using compressed log record headers improves application performance and reduces the log file's size.
	Default Value: `short`
	Valid Values:
	`short` An application uses compressed headers when writing log records in its log file.
	`full` An application uses complete headers when writing log records in its log file.
	Changes Take Effect: Immediately
	With the value set to `short`:
	• A header of the log file or the log file segment contains information about the application (such as the application name, application type, host type, and time zone) while single log records within the file or segment omit this information.
	• A log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg`, for `Standard`, `Interaction`, `Trace`, or `Debug` messages respectively.
	• `Message ID` does not contain the prefix GCTI or the application type ID.
	A log record in the `short` format looks like this:
	`2002-05-07T18:15:33.952 Std 05060 Application started`
	A log record in the `full` format looks like this:
	`2002-05-07T18:11:38.196 Standard localhost cfg_dbserver GCTI-00-05060 Application started`
	Note: Whether `full` or `short` format is used, time is printed as specified by the `time_format` option.

time_convert	Specifies the system in which an application calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970). Default Value: `Local` Valid Values: `local` Time of log-record generation expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of the application's host computer is used. `utc` Time of log-record generation expressed as Coordinated Universal Time (UTC). Changes Take Effect: Immediately

Table 4: Log Options (Continued)

Option	Description
time_format	Specifies how to represent the time in a log file when an application generates log records. Default Value: `time` Valid Values: `time` Time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format. `locale` Time string is formatted according to the system's locale. `ISO8601` Date in the time string is formatted according to ISO 8601 format. Fractional seconds are given in milliseconds. Changes Take Effect: Immediately A log record's time field in `ISO8601` format looks like this: `2001-07-24T04:58:10.123`
print-attributes	This log option has no effect on ETL Runtime. Default Value: `false` Valid Values: `true, false` Changes Take Effect: Immediately
check-point	Specifies how often, in hours, ETL Runtime generates a `check-point` log event to divide the log into sections equal in time. By default, ETL Runtime generates this log event every hour. Setting the option to `0` prevents generation of `check-point` events. Default Value: `1` Valid Values: `0–24` Changes Take Effect: Immediately

Table 4: Log Options (Continued)

Option	Description
memory	If configured, specifies the name of the file to which ETL Runtime regularly prints a snapshot of the memory output (see page 32). The new snapshot overwrites previously written data. If ETL Runtime terminates abnormally, this file contains the latest log messages. Memory output is not recommended for processors with a CPU frequency lower than 600 MHz. **Note:** If the file specified as the `memory` file is located on a network drive, ETL Runtime does not create a snapshot file (with the extension `*.memory.log`). Default Value: No default value Valid Value: `<string>` (memory file name) Changes Take Effect: Immediately

memory-storage-size	If configured, specifies the buffer size for log output to the memory. Refer also to “Log Output Options” on page 32 for more information.
	Default Value: `2 MB`
	Valid Values:
	`<number> KB` or `<number>` — Size of the memory output, in kilobytes. The minimum value is `128 KB.`
	`<number> MB` — Size of the memory output, in megabytes. The maximum value is `64 MB.`
	Changes Take Effect: When memory output is created

spool	Specifies the folder, including full path to it, in which ETL Runtime creates temporary log-related files. If you change this value while ETL Runtime is running, the change does not affect the currently open network output. Default Value: The application’s working directory Valid Value: `path` — The full path of the folder Changes Take Effect: Immediately

Table 4: Log Options (Continued)

Option	Description

compatible-output-priority	Specifies whether ETL Runtime uses 6.*x* output logic.
	Default Value: `false`
	Valid Values:
	`true` The log of the level specified by one of the log output options described on page 32 is sent to the specified output.
	`false` The log of the level specified by one of the log output options described on page 32 and higher levels is sent to the specified output.
	Changes Take Effect: Immediately
	For example, you configure the following options in the `log` section for a 6.*x* application and for a 7.*x* application:
	`[log]`
	`verbose = all`
	`debug = file1`
	`standard = file2`
	The 6.*x* application's log file content is as follows:
	• `file1` contains `Debug` messages only.
	• `file2` contains `Standard` messages only.
	The 7.*x* application's log file content is as follows:
	• `file1` contains `Debug`, `Trace`, `Interaction`, and `Standard` messages.
	• `file2` contains `Standard` messages only.
	If you set `compatible-output-priority` to `true` in the 7.*x* application, the content of its log file is the same as for the 6.x application.
	Warning! Genesys does not recommend changing the default value of the `compatible-output-priority` option unless you have specific reasons to use the 6.*x* log output logic—that is, to mimic the output priority as implemented in releases 6.*x*. Setting this option to `true` affects log consistency.

Log Output Options

To configure log outputs, set log level options (`all`, `standard`, `interaction`, `trace`, `memory`, and/or `debug`) to the desired types of log output (`stdout`, `stderr`, `network`, `memory`, and/or `[filename]` for log file output).

You can use:

- One log level option to specify different log outputs.
- One log output type for different log levels.

- Several log output types simultaneously for logging the events of the same or different log levels.

You must separate the log output types by a comma when you are configuring more than one output for the same log level. See “Examples” on page 36.

Note: The log output options are activated according to the setting of the `verbose` configuration option.

Warning!

- If you direct log output to a file on the network drive, an application does not create a snapshot log file (with the extension `*.snapshot.log`) in case it terminates abnormally.
- Directing log output to the console (by using the `stdout` or `stderr` settings) can affect application performance. Avoid using these log output settings in a production environment.

Table 5: Log Output Options

Option	Description
all	Specifies the outputs to which ETL Runtime sends all log events. You must separate log output types with commas when you configure more than one output type. Default Value: `stdout` Valid Values: `stdout` Log events are sent to the Standard output (`stdout`). `stderr` Log events are sent to the Standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores log events in the Log Database. Setting the `all` log level option to `network` enables ETL Runtime to send log events of `Standard`, `Interaction`, and `Trace` levels to Message Server. Log events of `Debug` level are neither sent to Message Server nor stored in the Log Database. `memory` Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the file is created in the application’s working directory. Changes Take Effect: Immediately For example, `all = stdout, logfile` **Note:** To ease the troubleshooting process, consider using unique names for log files that different applications generate.

Table 5: Log Output Options (Continued)

Option	Description
standard	Specifies the outputs to which an application sends log events of the `Standard` level. You must separate log output types with commas when you configure more than one output type. Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the Standard output (`stdout`). `stderr` Log events are sent to the Standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the file is created in the application's working directory. Changes Take Effect: Immediately For example, `standard = stderr, network`
interaction	Specifies the outputs to which an application sends log events of the `Interaction` and higher levels (that is, log events of `Standard` and `Interaction` levels). You must separate log outputs with commas when you configure more than one output type. Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the Standard output (`stdout`). `stderr` Log events are sent to the Standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the file is created in the application's working directory. Changes Take Effect: Immediately For example, `interaction = stderr, network`

Table 5: Log Output Options (Continued)

Option	Description
trace	Specifies the outputs to which an application sends log events of `Trace` and higher levels (that is, log events of `Standard`, `Interaction`, and `Trace` levels). You must separate log outputs with commas when you configure more than one output type. Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the Standard output (`stdout`). `stderr` Log events are sent to the Standard error output (`stderr`). `network` Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. `memory` Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the file is created in the application's working directory. Changes Take Effect: Immediately For example, `trace = stderr, network`
debug	Specifies the outputs to which ETL Runtime sends log events of `Debug` and higher levels (that is, log events of `Standard`, `Trace`, `Interaction`, and `Debug` levels). You must separate log output types with commas when you configure more than one output type. Default Value: No default value Valid Values (log output types): `stdout` Log events are sent to the Standard output (`stdout`). `stderr` Log events are sent to the Standard error output (`stderr`). `memory` Log events are sent to the memory output on the local disk. This output is the safest in terms of the application performance. `[filename]` Log events are stored in a file with the specified name. If you do not specify a path, the file is created in the application's working directory. Changes Take Effect: Immediately For example, `debug = stderr, /usr/local/genesys/logfile` **Note:** Log events of `Debug` level are never sent to Message Server or stored in the Log Database.

Log File Extensions

You can use the following file extensions to identify log files that ETL Runtime creates for various types of output:

- `*.log`—Assigned to log files when you configure output to a log file. For example, if you set `standard = etlruntimelog`, ETL Runtime prints log messages into a text file called `etlruntimeloglog.<time_stamp>.log`.
- `*.qsp`—Assigned to temporary (spool) files when you configure output to the network, but the network is temporarily unavailable. For example, if you set `standard = network`, ETL Runtime prints log messages into a file called `etlruntime.<time_stamp>.qsp` during the time the network is unavailable.
- `*.snapshot.log`—Assigned to files containing the output snapshot when you configure output to a log file. The file contains the last log messages that ETL Runtime generates before abnormal termination. For example, if you set `standard = etlruntimelog`, ETL Runtime prints the last log message into a file called `etlruntime.<time_stamp>.snapshot.log` in case of failure.

Note: Provide `*.snapshot.log` files to Genesys Customer Care when reporting a problem.

- `*.memory.log`—Assigned to log files that contain the memory output snapshot when you configure output to memory and redirect the most recent memory output to a file. For example, if you set `standard = memory` and `memory = etlruntime`, ETL Runtime prints the latest memory output to a file called `etlruntime.<time_stamp>.memory.log`.

Examples

This section presents examples of a `log` section you might configure for an application operating in production mode and in two lab modes, debugging and troubleshooting.

Production Mode Log Section

```
[log]
verbose = standard
standard = network, etlruntimelogfile
```

With this configuration, ETL Runtime generates log events of the `Standard` level only, and sends them to the standard output, to Message Server, and to a file named `etlruntimelogfile`, which ETL Runtime creates in its working directory. Genesys recommends that you use this or a similar configuration in a production environment.

Warning! Directing log output to the console (by using the stdout or stderr settings) can affect application performance. Avoid using these log output settings in a production environment.

Lab Mode Log Section

```
[log]
verbose = all
all = stdout, /usr/local/genesys/etlruntimelogfile
trace = network
```

With this configuration, ETL Runtime generates log events of the Standard, Interaction, Trace, and Debug levels, and sends them to the standard output and to a file named etlruntimelogfile, which ETL Runtime creates in the /usr/local/genesys/ directory. In addition, ETL Runtime sends log events of the Standard, Interaction, and Trace levels to Message Server. Use this configuration to test new interaction scenarios in a laboratory environment. Be sure to appropriately set the debug-level option in the ETLRuntime section.

Failure-Troubleshooting Log Section

```
[log]
verbose = all
standard = network
all = memory
memory = etlruntimefile
memory-storage-size = 32 MB
```

With this configuration, ETL Runtime generates log events of the Standard level and sends them to Message Server. It also generates log events of the Standard, Interaction, Trace, and Debug levels, and sends them to the memory output. The most current log is stored to a file named etlruntimelogfile, which the application creates in its working directory. An increased memory storage enables ETL Runtime to save more log information generated before a failure. Use this configuration when trying to reproduce an application failure. The memory log file would contain the snapshot of ETL Runtime's log at the moment of failure. This should help you and Genesys Customer Care identify the reason for the failure. Be sure to appropriately set the debug-level option in the ETLRuntime section.

Log-Filter Section

This section must be called `log-filter`. Table 6 describes the option you configure in this section.

Table 6: Log-Filter Option

Option	Description
default-filter-type	Specifies the default way of presenting KVList information (or, possibly, User Data) in the log. The selected option is applied to all KVList pairs of the User Data, except the ones that are explicitly defined in the `log-filter-data` section. Default Value: `copy` Valid Values: `copy` The keys and values of the KVList pairs are copied to the log. `hide` The keys of the KVList pairs are copied to the log; the values are replaced with strings of asterisks. `skip` The KVList pairs are not copied to the log. Changes Take Effect: Immediately

Example

```
[log-filter]
default-filter-type=copy
```

Here is an example of a log using the default log filter settings:

```
message RequestSetCallInfo
    AttributeConsultType            3
    AttributeOriginalConnID         008b012ece62c8be
    AttributeUpdateRevision         2752651
    AttributeUserData               [111] 00 27 01 00..
            'DNIS'                  '8410'
            'PASSWORD'              '111111111'
            'RECORD_ID'             '8313427'
    AttributeConnID                 008b012ece62c922
```

Log-Filter-Data Section

This section must be called `log-filter-data`. Table 7 describes the options you configure in this section.

Table 7: Log-Filter-Data Options

Option	Description
<key name>	Specifies the way of presenting the KVList pair defined by the key name in the log. Specification of this option supersedes the default way of KVList presentation as defined in the `log-filter` section for the given KVList pair. Default Value: `copy` Valid Values: `copy` The key and value of the given KVList pair are copied to the log. `hide` The key of the given KVList pair is copied to the log; the value is replaced with a string of asterisks. `skip` The KVList pair is not copied to the log. Changes Take Effect: Immediately

Example

```
[log-filter-data]
PASSWORD=hide
```

Here is an example of the log with option `PASSWORD` set to `hide`:

```
message RequestSetCallInfo
    AttributeConsultType         3
    AttributeOriginalConnID      008b012ece62c8be
    AttributeUpdateRevision      2752651
    AttributeUserData            [111] 00 27 01 00..
            'DNIS'               '8410'
            'PASSWORD'           '****'
            'RECORD_ID'          '8313427'
    AttributeConnID              008b012ece62c922
```

Genesys

Chapter

4 Runtime Parameters

You can affect how the various ETL Runtime modules run by specifying certain command-line parameters when the ETL processes are invoked. This chapter describes those parameters:

The .properties Files

A 7.6 Data Mart installation deploys four properties files which store runtime parameters associated with certain ETL Runtime modules and initial Data Mart configuration:

- `etl.properties`—Use this file in conjunction with the Transformation, Aggregation, Configuration Tracking, and Tenant Alias Tracking modules.
- `createdb.properties`—Use this file if you choose not to initialize the Data Mart during configuration of your `Data Mart` Application object.
- `purge.properties` —Use this file in conjunction with the Purging module.
- `tenants_alias_update.properties`—Use this file (or `etl.properties`) in conjunction with the Tenant Alias Tracking module.

Using properties files is convenient when the command line for your ETL process exceeds 255 characters.

List of Runtime Parameters

Table 8 describes the ETL Runtime parameters you can modify to control Data Mart aggregation, transformation, purging, tenant tracking, and database creation. They are presented in alphabetical order, which differs from the order actually encountered in the various initialization and property files. These parameters are case sensitive. In addition, when a parameter accepts a value, the following information may be specified for that parameter:

- Default values if the parameter is absent
- Recommended values installed by the Historical Reporting Wizard
- Valid values

When a parameter accepts no values, the following is specified:

- Default behavior (set or not set)
- Description of ETL Runtime behavior given the parameter's presence.

You can find these parameters in various `.properties` files located in the directory where ETL Runtime is installed. You can also issue any runtime parameter at the command line when invoking ETL Runtime manually. Parameters you issue at the command line override settings in the `.properties` files. The legend in Table 8 provides names of the files in which ETL Runtime parameters are recognized.

Note: ETL Runtime does not ignore tabs and spaces in the properties files.

Table 8: ETL Runtime Parameters

Parameter*				Description
T	P	C	L	
`agg`			✓	This parameter takes no values. If it is present, ETL Runtime performs data aggregation for all defined report views. During the aggregation process, ETL Runtime reads data directly from fact tables in the Data Mart—not from the source ODS(s). Default Behavior: The installation process does not set this parameter.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*				Description
T	P	C	L	
aggOnly				This parameter takes no values. If it is present, ETL Runtime performs data aggregation for all defined report views. During the aggregation process, ETL Runtime reads data directly from fact tables in the Data Mart—not from the source ODS(s). With this parameter, data transformation, if specified by the transOnce parameter, is performed separately. Default Behavior: The installation process sets this parameter.
			✓	
aggQuota				This parameter specifies the number of aggregations that ETL Runtime can simultaneously perform. Each aggregation process occupies the resources of one writer. If your environment uses high-performance RDBMSs, you may notice that ETL Runtime processes aggregations more quickly upon increasing the value of this parameter. Default Value: 1 Valid Values: 1 to the number of writers (specified by the numberOfWriters runtime parameter.)
			✓	
CfgAppName				This parameter specifies the Configuration Server Application object of type ETL Proxy for use by ETL Runtime. **Note:** In a multiple Configuration Server environment, the application specified by this parameter should exist in every Configuration Server and have the same name. For example, if CfgAppName = MyConfig in one Configuration Server, then set CfgAppName to MyConfig in every Configuration Server. Default Value: ETL_Proxy Valid Values: applications of type ET_Proxy
✓			✓	
CfgAppPassword				This parameter specifies the password for the Configuration Server Application object specified by CfgAppName and is mandatory if you have defined a password for the Configuration Server Application object. **Note:** In situations where ETL Runtime connects to more than one Configuration Server, all Configuration Servers must use the same password (as specified in CfgUserPassword). Default Value: <*empty string*> Valid Value: Valid password for the application specified by CfgAppName.
✓			✓	

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
CfgUser or CfgUserName	✓			✓	This parameter specifies the name of a Configuration Server user. This user must have, at minimum, read access to all CC Analyzer applications and to all tenants (and their objects) that ETL Runtime handles. **Note:** In a multiple Configuration Server environment, for each Configuration Server, specify a user with the same name as that specified by this option. Default Value: `default` Valid Values: Valid Configuration Server user names
CfgUserPassword	✓			✓	This parameter (or the `CfgUserPasswordEncrypted` parameter) specifies the password for the Configuration Server user specified by `CfgUser`. **Note:** In a multiple Configuration Server environment, for each Configuration Server, specify a user with the same name as that specified by this parameter. Default Value: `password` (This value is set when you install Data Mart.) Valid Values: The valid password assigned to the user specified in `CfgUser` This parameter is provided for backward compatibility.
CfgUserPassword Encrypted	✓			✓	This parameter specifies the encrypted password for the Configuration Server user specified by `CfgUser`. Valid Values: The valid encrypted password assigned to the user specified in `CfgUser`
chunkBufferQuota				✓	This parameter specifies the threshold for the number of statistics that ETL Runtime may buffer in memory. ETL Runtime ceases reading data chunks when this threshold is crossed and resumes reading when the write operation frees enough space to a level below the threshold. Environments containing a significant number of data chunks or excessive backlogging of non-transferred chunks may benefit from increasing this parameter. Default Value: `786432` (statistics) Valid values: Positive integers from `1024` to `16777216`
chunkQuota				✓	This parameter specifies the number of parsed data chunks that ETL Runtime may buffer in memory. Increasing the value of this parameter may speed up data transformation in environments containing many small unread chunks. Default value: `1536` (data chunks) Valid values: `16` to `8192`

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
chunksAtOnce				✓	This parameter specifies the number of data chunks that ETL Runtime can read in a single read operation from a single ODS source. Setting a higher value for this parameter speeds up data transformation, especially if data chunks are relatively small. Default Value: 16 (chunks) Valid Values: 1 to 64
communication-alarm				✓	If present, this parameter indicates that ETL Runtime will send one of two events to the Solution Control Server upon detecting changes in connectivity to constituent ODSs: 58-30002–Communication problem has been detected 58-30005–Communication to all servers has been restored Default Behavior: The installation process does not set this parameter. This parameter takes no values.
conf					This parameter specifies the file name from which ETL Runtime reads parameter values that are not specified at the command line. Precede this file by a relative or absolute path if the file is not located in the same directory as the ETL Runtime executable. If absent, ETL Runtime uses only those parameters issued at the command line.
createdb			✓		This parameter takes no values. If it is present, ETL Runtime creates the Data Mart database and tables, populates them with metadata, then exits. If ETL Runtime cannot create tables, the application generates an exception and exits. This parameter overrides almost all other parameters such as agg and aggOnly. Default Behavior: The installation process sets this parameter.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
data-excess-alarm				✓	If present, this parameter indicates that ETL Runtime will send one of two events to the Solution Control Server regarding variations in the data transfer rate between ODS and the Data Mart: 58-30001–Too much data was written to the database within the specified threshold 58-30004–Data write rate is within the threshold The trigger for sending either event is determined by a threshold you set (specified by the `data-excess-threshold` runtime parameter) and the frequency with which ETL Runtime checks whether this threshold has been breached (specified by the `data-excess-check-interval` runtime parameter). Default Behavior: The installation process does not set this parameter. This parameter takes no values.
data-excess-check-interval				✓	This parameter specifies how often, in minutes, ETL Runtime checks for overage in the number of data rows transferred, the maximum of which is specified by the `data-excess-threshold` runtime parameter. Default Value: `2` Valid Values: Positive integers from `1` to `60`
data-excess-threshold				✓	This parameter specifies the maximum number of data rows that ETL Runtime can transfer during the interval specified by the `data-excess-check-interval` runtime parameter. Default Value: `1` Valid Values: Positive integers **Note:** Consider changing the default value to better align with the performance expectations of your environment. ETL Runtime can write as much data into the Data Mart as Data Sourcer is able to produce, so set a value accordingly. If you use the default value, expect one or two alarms to be raised in the Solution Control Server along with every data transmission every 15 minutes (using the default time profile). Using this parameter in this manner can reassure you that ETL processes have started and are transferring data.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
data-flow-alarm				✓	If present, this parameter indicates that ETL Runtime will send one of two events to the Solution Control Server regarding changes in data flow between ODS and the Data Mart: 58-30000–No data has been transferred within the specified threshold 58-30003–Data transfer has resumed The trigger for sending either event is determined by the frequency with which ETL Runtime checks for data flow (specified by the `data-flow-check-interval` runtime parameter) and the period of time that can elapse before the condition of no data flow is no longer acceptable (specified by the `data-flow-timeout` runtime parameter). Default Behavior: The installation process does not set this parameter. This parameter takes no values.
data-flow-check-interval				✓	This parameter specifies the frequency, in minutes, with which ETL Runtime checks for data flow from ODS to the Data Mart. Default Value: `2` Valid Values: Positive integers, `1–60`
data-flow-timeout				✓	This parameter specifies how much time, in minutes, can pass before ETL transformation must resume a previously suspended data transfer before logging a message indicating a data flow problem. Default Value: `2` Valid Values: Integers greater than `1`
dba_user	✓			✓	This parameter specifies an administrative user name for the RDBMS where the Data Mart is located and is required for the Tenant Alias–Tracking module, which creates new accounts in the RDBMS. Default Value: No value (for security reasons)
dba_pass	✓			✓	This option specifies the password for the administrative RDBMS user specified by `dba_user`. You can also specify this option from the command line (as you can with any other option). You are responsible for keeping the administrative password secure.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
deadlock Threshold				✓	This parameter specifies how often, in seconds, ETL Runtime checks its threads (the number of which is specified by the `numberOfWriters` runtime parameter) for deadlocks. Default Value: `900` (15 minutes) Valid Values: Any positive integer
dontUpdateStats				✓	This parameter takes no values. Prior to each aggregation, ETL Runtime checks how much time has elapsed since the statistics of the parent report view were last updated. If this period is more than 14 days, ETL Runtime initiates an update of the statistics. If this parameter is present, however, ETL Runtime updates no statistics. This parameter is valid for Oracle and Sybase RDBMSs only. Default Behavior: The installation process does not set this parameter.
dropTransferred Tables				✓	This parameter takes no values. If it is present, ETL Runtime drops tables containing statistical data that is transferred from the source ODS(s) within hours of ascertaining that all data transferred successfully. Genesys recommends using this parameter to keep ODS small and that you implement DBA procedures and tools to maintain data integrity. These tools include disk mirroring, database backups, transaction logging, database recovery procedures, and so on. Default Behavior: The installation process does not set this parameter. **Note:** On Sybase RDBMSs, you must set the `ddl in tran` Sybase option to `true` for `dropTransferredTables` to function properly. You can set this option by issuing the following command: `sp_dboption <dbname>,"ddl in tran",true`
jdbcurl	✓	✓	✓	✓	This parameter specifies the string sent to the JDBC driver to indicate which Data Mart that ETL Runtime is to use. This string is mandatory for ETL Runtime to function, and must be formatted as follows: `jdbc:oracle:thin:@`*`dbhost`*`:`*`dbport`*`:`*`ORACLE_SID`* (for Oracle) `jdbc:jtds:sqlserver://`*`dbhost`*`:`*`dbport`*`;DatabaseName=`*`dbname`* (for Microsoft SQL Server) `jdbc:sybase:Tds:`*`dbhost`*`:`*`dbport`*`/`*`dbname`* (for Sybase) `jdbc:db2://`*`dbhost`*`:`*`dbport`*`/`*`dbname`* (for DB2) For Windows NT, this parameter may initially get its value from the data you provide during installation.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
log	✓	✓	✓	✓	This parameter specifies the path and file name of the ETL Runtime log file. If not specified, ETL Runtime sends log data to the Management Layer in `stdout`. Default Values: • `createdb.log` for `createdb.properties` • `tenants_alias_update.log` for `tentants_alias_update.properties` Valid Values: Valid filename and path if not the current path for your operating system. **Note:** ETL Runtime cannot process data if you specify an invalid value such as a nonexistent path, or a file with inadequate permissions, or if there is no available disk space. Default Behavior: The installation process does not set this parameter for `purge.properties`.
levelOfLog		✓	✓	✓	This parameter specifies the detail level of log messages that ETL Runtime generates for the Transformation, Aggregation, and Purge modules. (Refer to the `loglevel` runtime parameter to specify this information for the Configuration Tracking and Tenant Alias Tracking modules.) Default Value: `.:INFO` (Signifying that ETL Runtime logs all categories with an informational level of verbosity.) Valid Values: `<category>:[<`*`value`*`>][,<`*`category`*`>:[<`*`value`*`>]...]` where *`category`* is one of the values specified in Table 9 on page 56 and *`value`* is one of the following corresponding to the level of information desired: `OFF`—ETL Runtime logs no messages from the corresponding category. `SEVERE`—ETL Runtime logs only severe messages from the corresponding category. `WARNING`—ETL Runtime logs severe and warning messages. `INFO`—ETL Runtime logs severe, warning, and informational messages. `CONFIG`—ETL Runtime logs severe, warning, informational, and configuration messages. `FINE`—Same as `CONFIG` plus an even finer level of messages. `FINER`—Same as `FINE` plus an even finer level of messages. `FINEST`—Same as `FINER` plus an even finer level of messages. `ALL`—ETL Runtime logs all messages. The higher the `value`, the greater the detail that ETL Runtime logs. When you specify no value, ETL Runtime uses `INFO`.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*	T	P	C	L	Description
loglevel	✓			✓	This parameter specifies the detail level of log messages that ETL Runtime generates for the Configuration Tracking and Tenant Alias Tracking modules. (Refer to the `logOflevel` runtime parameter to specify this information for the Transformation, Aggregation, and Purging modules.) This parameter's value must consist of a comma-separated string of entries in the format `action:value` or just `action`. *`value`* must be an integer from `0` to `9`. A zero (`0`) value signifies that no messages from the corresponding action are sent to the log; `9` signifies that the most detailed messages are sent to the log—the higher the value, the greater the detail. When you do not specify a `value`, ETL Runtime uses `9`. See Table 10 on page 57 for a listing and description of the options you can specify for this parameter.
noSrcData				✓	This parameter takes no values. If it is present, ETL Runtime does not transform statistical data from source ODSs. This parameter does not affect the reading of new contact center objects from ODSs. Default Behavior: The installation process does not set this parameter.
numberOfIntervalsToWaitFor_AGG_BY_HOUR				✓	This parameter specifies the number of 15-minute intervals that must elapse before statistical data is aggregated up to the hour aggregation level. Valid Values: Positive integers Default Behavior: The installation process does not set this parameter.
numberOfIntervalsToWaitFor_AGG_BY_DAY				✓	This parameter specifies the number of hour intervals that must elapse before statistical data is aggregated up to the day aggregation level. Default Behavior: The installation process does not set this parameter. Valid Values: Positive integers Default Behavior: The installation process does not set this parameter.
numberOfIntervalsToWaitFor_AGG_BY_WEEK				✓	This parameter specifies the number of day intervals that must elapse before statistical data is aggregated up to the week aggregation level. Valid Values: Positive integers Default Behavior: The installation process does not set this parameter.
numberOfIntervalsToWaitFor_AGG_BY_MONTH				✓	This parameter specifies the number of day intervals that must elapse before statistical data is aggregated up to the month aggregation level. Valid Values: Positive integers Default Behavior: The installation process does not set this parameter.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*	T	P	C	L	Description
numberOfIntervalsToWaitFor_AGG_BY_QUARTER				✓	This parameter specifies the number of month intervals that must elapse before statistical data is aggregated up to the quarter aggregation level. Valid Values: Positive integers Default Behavior: The installation process does not set this parameter.
numberOfIntervalsToWaitFor_AGG_BY_YEAR				✓	This parameter specifies the number of month intervals that must elapse before statistical data is aggregated up to the year aggregation level. Valid Values: Positive integers Default Behavior: The installation process does not set this parameter.
numberOfWriters				✓	This parameter specifies the number of threads that ETL Runtime dedicates to writing data chunks to the Data Mart. Each thread opens one connection to the Data Mart. Upon startup, ETL Runtime statically assigns report folders to these threads. If your environment uses high-performance RDBMSs, you may notice performance improvements upon increasing the value of this option. Default value: `2` Valid values: Positive integers from `1` to `8`
ORACLE_TablespaceNameForDataIndexes				✓	This parameter specifies a tablespace name in which ETL Runtime creates indexes for data tables. The account that ETL Runtime uses to connect to the database must either have enough quota in this tablespace or have some special permissions such as “dba role”. When absent, ETL Runtime creates all indexes for data tables in the user’s default tablespace. Starting with release 7.0.1, ETL Service Starter passes this parameter to ETL Runtime directly from the command line instead of reading the value from a `.properties` file. You can still use this parameter, however, in the indicated `.properties` files. Default Behavior: The installation process does not set this parameter.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*				Description
T	P	C	L	
ORACLE_TablespaceNameForDataTables				This parameter specifies a tablespace name in which ETL Runtime creates tables for data tables. The account that ETL Runtime uses to connect to the database must either have enough quota in this tablespace or have some special permissions such as “dba role”. When absent, ETL Runtime creates all tables in the user’s default tablespace. Starting with release 7.0.1, ETL Service Starter now passes this parameter to ETL Runtime directly on the command line instead of reading the value from a `.properties` file. You can still use this parameter, however, in the indicated `.properties` files. Default Behavior: The installation process does not set this parameter.
			✓	
pass				This parameter specifies the password of the user who is connecting to the Data Mart. Either this or the `passEncrypted` parameter must be present on the command line or in the referenced property file. ETL Runtime fails to start if both parameters are absent. This parameter is provided for backward compatibility. Default Behavior: The installation process does not set this parameter.
✓	✓	✓	✓	
passEncrypted				This parameter specifies the encrypted password of the user who is connecting to the Data Mart. Either this or the `pass` parameter must be present on the command line or in the referenced property file. ETL Runtime fails to start if both parameters are absent. This parameter initially gets its value from the data you provide during installation. Valid Value: Valid password for the specified Data Mart
✓	✓	✓	✓	
reverse_password				This parameter is used in the Tenant Alias–Tracking (TAT) algorithm for generating passwords for tenant accounts in a multi-tenant environment. If set, ETL Runtime reverses the generated password. Default Behavior: The installation process does not set this parameter. Valid Value: `on.` Any value other than `on,` or this parameter’s absence from the `.properties` file, instructs ETL Runtime not to reverse the password. **Note:** This parameter applies only for DB2 Data Marts.
✓				

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
setFirstDayOfWeek				✓	This parameter specifies the start day of the week for week-level aggregations and affects how week-level aggregation is performed for *all* report views in the Data Mart. If you change this setting, reaggregation is not performed automatically. To reaggregate data, Genesys recommends that you contact Genesys Customer Care for assistance in manipulating the REP_REBUILD_LOG Data Mart table. This parameter overrides locale settings, which, by default, set Sunday as the first day of the week for US-based locales and Monday as the first day of the week for European-based locales. Default Behavior: The processes that are deployed during configuration using the Historical Reporting Wizard all explicitly call this parameter, which sets Sunday as the first day of a week. If you remove this setting, the start day that ETL Runtime uses for Data Mart weeks depends on your locale settings. Valid Values: 1, 2, 3, 4, 5, 6, 7 where 1-7 correspond to Sunday–Saturday respectively. **Warning!** If you elect to use this parameter, Genesys recommends that you use it for all ETL Runtime processes you run and set it in all properties files. Refer to “What Constitutes a Week?” on page 67 for an in-depth discussion on how ETL Runtime determines the week number within year and week boundaries.
setMinimalDaysInFirstWeek				✓	This parameter overrides your locale settings that determine which days of the new year comprise the first Data Mart week. A setting of 1, for example, instructs ETL Runtime to always include one day of the new year in the first Data Mart week. This convention coincides with that used by US locales which always include January 1 in the first week of a year. A setting of 4, for example, instructs ETL Runtime to include a minimum of 4 days of the new year in the first Data Mart week. This setting coincides with the convention used by European locales where the day January 4 is always included in the first week of the year. Specifying this parameter only has significance when you also specify the wholeWeeks parameter. Default Value: none Valid Values: 1, 2, 3, 4, 5, 6, 7 Refer to “What Constitutes a Week?” on page 67 for additional information.
sourceDefReReadTimeout				✓	This parameter specifies the time interval, in seconds, for checking for new ODS sources or for changes in the state of an existing ODS source (started or stopped). Default Value: 300 Valid Values: Positive integers greater than 29

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*					Description
	T	P	C	L	
srcObjectsReread-Timeout				✓	This parameter specifies the time interval, in seconds, for rereading ODS object tables and searching for new layout templates. Default Value: `3600` (1 hour) Valid Values: Positive integers greater than `29`
srcRefreshTimeout				✓	This parameter specifies the time interval, in seconds, for checking for new data chunks. Default Value: `300` Valid Values: Positive integers greater than `29`
surviveLossOfA Source				✓	This parameter determines ETL Runtime's response to exceptions in communication with constituent ODSs. Default Value: `"MOST"` Valid Values: `"ALWAYS", "MOST", "NEVER"` (values are case sensitive) `"ALWAYS"` signifies that ETL Runtime will exit whenever any ODS exception occurs. `"MOST"` signifies that ETL Runtime will not exit while at least one constituent ODS operates without exception. `"NEVER"` signifies that ETL Runtime ignores any and all exceptions that might be generated by its constituent ODSs.
tenants_shortcut	✓				This parameter specifies how the Tenant Alias–Tracking module generates account names. If set, the letter *t* prefixes the tenant user name instead of *tenant_*. Refer to Chapter 8 on page 79 for more information. Default Behavior: The installation process does not set this parameter Valid Values: `on, off`
transOnce				✓	This parameter takes no values. If it is present, ETL Runtime exits after aggregating all available data. If absent, ETL Runtime periodically checks ODS for new data and never exits. This option is useful for external scheduling of ETL Runtime. Default Behavior: The installation process does not set this parameter.
updateAllStats				✓	This parameter instructs ETL to issue the update statistics command to both the parent and child views. Without this option, ETL Runtime updates only the minute-, hour-, day-, and month-level views. Issuing this parameter enables ETL Runtime to update the week-, quarter-, and year-level views as well following aggregation. This parameter is valid only for Oracle and Sybase RDBMSs. Default Behavior: The installation process does not set this parameter.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Table 8: ETL Runtime Parameters (Continued)

Parameter*				Description
T	P	C	L	
updateStatsAfter			✓	This parameter allows you to control the number of days between statistic updates. This parameter is valid only for Oracle and Sybase RDBMSs. Default Value: `14` Valid Values: Nonnegative integers. `0` means update statistics each time Default Behavior: The installation process does not set this parameter.
user ✓	✓	✓	✓	This parameter specifies the user name for connecting to the Data Mart and is mandatory. This user must have appropriate database privileges for the tablespaces that ETL Runtime populates. Default Behavior: The installation process assigns the value you provide to this parameter.
wholeWeeks			✓	This parameter specifies that all Data Mart weeks be comprised of seven days. This parameter takes no values. If absent, some Data Mart weeks may consist of fewer than seven days. Refer to “What Constitutes a Week?” on page 67 for an in-depth discussion on how ETL Runtime determines the week number within a year and week boundaries. Default Behavior: The installation process does not set this parameter.

***Legend:**
T=tenants_alias_update.properties **P=purge.properties**
C=createdb.properties **L=etl.properties**

Because of improvements in runtime optimization, many runtime parameters that were necessary to tune ETL Runtime prior to the 7.0.2 release are now obsolete. Such is the case for the following options which are no longer described in Table 8:

- `maxLevelOfAgg`
- `maxNumberOfChunksInReadQueue`
- `maxNumberOfChunksInWriteQueue`
- `maxNumberOfChunksInReadQueue`
- `maxNumberOfChunksInWriteQueue`
- `maxNumberOfConnToDist`
- `maxNumberOfWriteMonitorsPerView`
- `minNumberOfWriteMonitorsPerView`
- `noCheckTablesForFolderViews`
- `noSrcObjects`
- `numberOfDataReadersPerSource`
- `reinitAfterExceptionTimeout`
- `srcStartReadingTimeout`
- `syncDimWithGlobalCatalogTimeout`

Log-Level Categories

Log-level categories specify the amount and type of detail you see in messages that ETL Runtime writes to its various logs. These categories differ from the options you can specify in the `log` section of a `Data Mart` Application object. (Refer to the “Log Section” on page 25 for the latter.)

For the Transformation and Aggregation modules, you can set the `levelOfLog` parameter or rely on its default value, `.:INFO,` for an informational level of verbosity. For the Configuration Tracking and Tenants Alias Tracking modules, you can set the `logOflevel` parameter or use the default value, `INFO:9,` for an informational level of verbosity. Table 9 describes the categories you can specify for the `levelOfLog` parameter; Table 10 describes the options you can set for the `loglevel` parameter. As with all ETL Runtime parameters, these categories and options are case sensitive.

Table 9: ETL Runtime Log Categories

Category	Values
LFolder	Displays information about ETL Runtime’s check of tables and views of report layout folders.
LFolder.SynonymRegister	Displays information about ETL Runtime’s check and creation of aliases.
LFolder.TableRegister	Displays information about ETL Runtime’s creation of tables and views of report layout folders.
LFolder.ViewFactory	Displays information about ETL Runtime’s deletion and creation of report views.
Main	Displays information about operation of the Transformation module and certain runtime parameters for accessing the Data Mart. Also displays information about Data Mart creation.
Object.Catalog	Logs a message whenever ETL Runtime waits five or more minutes for information about a particular object in ODS. If you see this message, verify that your ODS is consistent. ETL Runtime will continue to wait until the information becomes available or until ETL Runtime detects a deadlock.
Object.Monitor	Displays information about ETL Runtime’s progress in maintaining objects.
Source.SourceDef	Displays information about ETL Runtime’s attempts to connect ODS sources.
SqlExec	Logs the sql statements and files that ETL Runtime processes during Data Mart creation and during upgrades.
Src.ChunkData.<sourceID>	Displays information about ETL Runtime’s progress in reading data chunks from an ODS source.
Src.ChunkLog.<sourceID>	Displays information about the sets of data chunks ETL Runtime reads and of the tables that ETL Runtime drops from the ODS source.

Table 9: ETL Runtime Log Categories (Continued)

Category	Values
Src.Scheyout.<sourceID>	Displays information about ETL Runtime's deletion of report layouts from the ODS source. *Scheyout* is a neologistic portmanteau for scheduled layout.
Thread.<threadName>	Displays information about ETL Runtime's thread management and some of the commands executed.
WatchDog	Displays information about ETL Runtime's attempts to detect second instances of ETL Runtime that are performing the same functions. The *WatchDog* is the thread ETL Runtime dedicates for this purpose.
Write.Heartbeat	Periodically displays information about the memory ETL Runtime consumes. This category also logs information about suspected deadlocks.
Write.Job.Agg.<viewID>	Displays information about the aggregation process.
Write.Job.Trans.<viewID>	Displays information about the transformation process.

Table 10: Log-Level Options

Option	Description
ALL	A catchall parameter used to display information about all actions.
AGG	Displays information about the individual aggregation steps ETL Runtime performs in the `trans_and_agg` and `agg_only` modes.
ALIAS	Displays information about aliases created for tenant accounts and encountered errors.
COMMAND	Displays information by individual threads about execution status of commands in the `trans_only`, `trans_and_agg`, `agg_only`, and `trans_once` modes.
CREATE-SQL	Displays dynamic SQL statements that are sent to the database server.
DEBUG	Displays debugging information.
DIST-POOLING	Displays information about accessing resources from shared pools (for example, connection pool, preparsed statement pool).
DROP	Displays information about ODS-dropped tables.
ERROR	Displays abnormal events that are within the ETL or generated during communication with external resources (for example, database, or network).
EVENT	Displays information about events that are sent and received between monitors.

Table 10: Log-Level Options (Continued)

Option	Description
INFO	Displays general informational messages.
INIT	Displays initialization information about internal objects (monitors, managers, and so forth).
LISTENER	Displays information about new listeners added to the monitor.
NEW_ ACCOUNT	Displays information about the creation and validation of tenant accounts.
PURGE	Displays information about data chunks that are marked `transferred`.
READ-SRC	Displays information (data chunks, objects, layouts, and so forth) about data that is obtained from ODS source(s).
SQL	Displays static SQL statements sent to ODS or the Data Mart.
SQL-STATS	Displays information about updated database statistics (table names, indexes).
START_AGG	Displays information about the aggregation steps taken by ETL Runtime in the `trans_and_agg` and `agg_only` modes.
START_WRITE_ MON	Displays information about invoking new write monitors.
THREAD-WAIT	Displays those threads that are waiting for other thread events.
WAKE	No longer in use.
WARNING	Displays information about noncritical errors.
WRITE	Displays information about data written to the Data Mart.

Setting Alarm Conditions

Reporting 7.0 and higher releases contain four runtime parameters and corresponding log messages that, when used in conjunction with alarm conditions in the Solution Control Interface, automatically alert you when your Reporting environment becomes stressed and when an acceptable level of activity has resumed. The parameters, described in detail beginning on page 45, are:

- `communication-alarm`
- `data-excess-alarm`
- `data-flow-alarm`

These parameters, in conjunction with some others, indicate under which conditions ETL Runtime should send certain messages to its log. Table 11 lists the corresponding log messages:

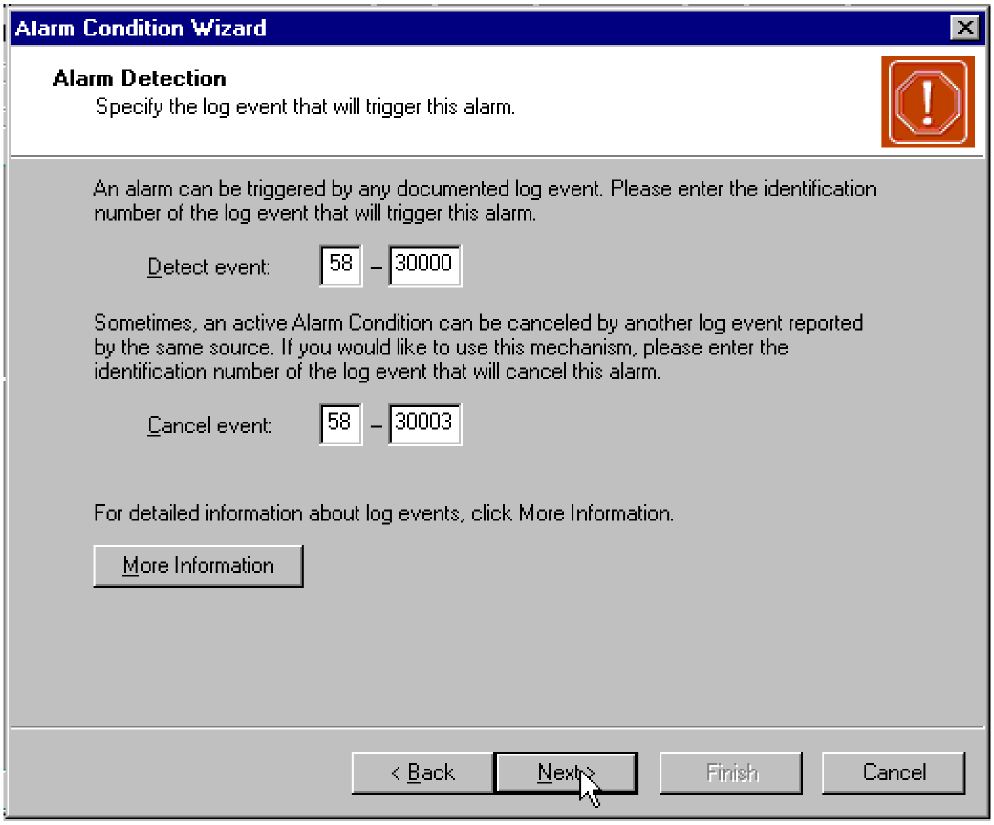
Table 11: ETL Runtime Log Messages

Trigger Event	Corresponding Clearing Event
58-30000 No data has been transferred within the specified threshold.	58-30003 Data transfer has resumed.
58-30001 Too much data was written to the database within the specified threshold.	58-30004 Data write rate is within the threshold.
58-30002 Communication problem has been detected.	58-30005 Communication to all servers has been restored.

Note: *Framework Combined Log Events Help* describes all Genesys log events including those of ETL Runtime. You can invoke this file directly from the SCI application or open it from the Genesys Customer Care website.

ETL Runtime default behavior automatically sends predefined messages to its log whenever predetermined situations occur. However, in addition to this notification, you may want to define alarm conditions if your environment operates many Data Mart applications simultaneously. Setting alarm conditions for the log messages that are most important to you enables you to monitor—within one application—exceptional activity in all applications. The alternative is to review the messages in each Data Mart log individually.

Framework Solution Control Interface Help explains how to create alarm conditions using the Alarm Condition Wizard. When you arrive at the `Alarm Detection` page in this Wizard, specify a `Detect` log event and a corresponding `Cancel` (or clearing) event, if one exists. Figure 7 illustrates how you might specify a `Detect` and `Cancel` event for a ETL Runtime–specific alarm condition.



Alarm Condition Wizard

Alarm Detection

Specify the log event that will trigger this alarm.

An alarm can be triggered by any documented log event. Please enter the identification number of the log event that will trigger this alarm.

Detect event: 58 – 30000

Sometimes, an active Alarm Condition can be canceled by another log event reported by the same source. If you would like to use this mechanism, please enter the identification number of the log event that will cancel this alarm.

Cancel event: 58 – 30003

For detailed information about log events, click More Information.

Figure 7: Specifying Detect/Cancel Events for ETL Runtime Within SCI

Chapter

5 Transformation Module

The ETL Runtime Transformation copies to the Data Mart, data from the ODS sources that are defined to it. This chapter covers these topics:

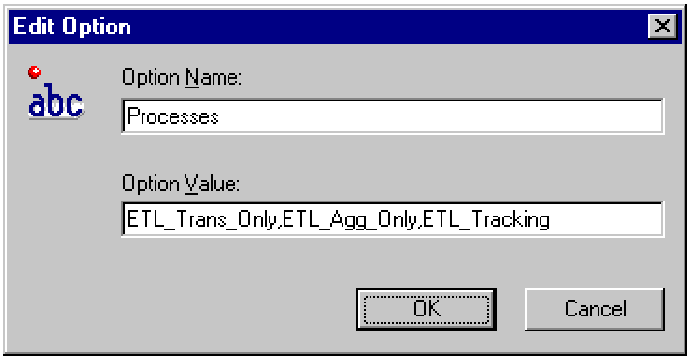
The ODS–Data Mart Connection

ETL Runtime connects to ODS and the Data Mart using a continuous Java Database Connectivity (JDBC) connection. The data-loading process in ETL Runtime functions by detecting new chunks of data since performing its last load. A *chunk* is a set of statistics that Data Sourcer collects for the objects belonging to one particular report layout over the period of time that is defined by the Data Sourcer collection time profile (by default, every 15 minutes).

The `srcRefreshTimeout` runtime parameter (described on page 54) determines the frequency of detection. The ETL Runtime transformation process creates tables in the Data Mart to hold this data as necessary, but does not manipulate the data itself. Once data transfers successfully, the transformation process deletes the data from ODS only if you have set the `dropTransferredTables` runtime parameter (page 48)—the default behavior does not set this option. An inherent fail-safe mechanism maintains data integrity if server problems occur. You can specify a one-time data transformation or an ongoing data transformation.

Scheduling Ongoing Transformation

If you use the `Genesys ETL Service Starter` to start ETL Runtime processes, a continuous data transformation process is already scheduled. The process name is `ETL_Trans_Only` as depicted in Figure 8.

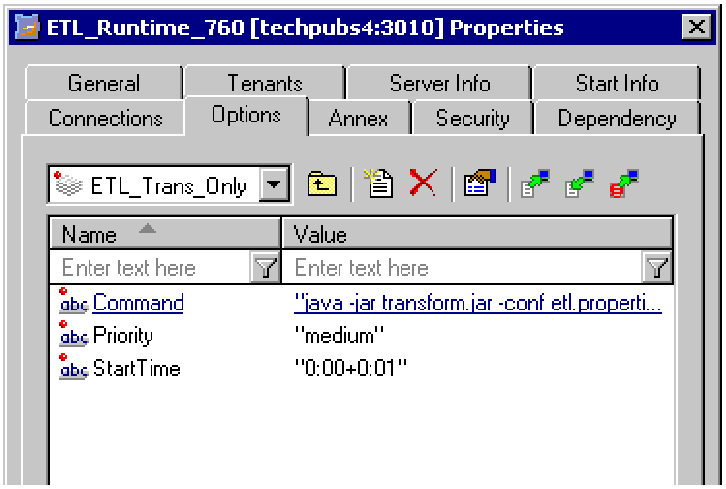


Option Name: Processes

Option Value: ETL_Trans_Only,ETL_Agg_Only,ETL_Tracking

Figure 8: The ETL Service Starter Includes ETL_Trans_Only

Figure 9 shows this process as defined in Configuration Manager.



Name	Value
Command	"java -jar transform.jar -conf etl.properti...
Priority	"medium"
StartTime	"0:00+0:01"

Figure 9: ETL_Trans_Only Process Definition

If you used the Historical Reporting Wizard to configure the `Data Mart` Application object, the Transformation module is automatically scheduled to run one minute past midnight and one minute after each hour thereafter. The Wizard also configures your `etl.properties` file with the values you specified for the `CfgAppName`, `CfgAppPassword`, `CfgUser`, and `CfgUserPasswordEncrypted` runtime parameters for connection to your Configuration Server.

Note: The Wizard no longer writes to the `etl.properties` file values specifying connection parameters to your Data Mart, including `user`, `pass`, and `jdbcurl`. Instead, the starter application adds these parameters and their values to the command line immediately prior to starting a Data Mart application.

Scheduling a Onetime Transformation

To schedule a one-time transformation, edit the properties of your `Data Mart` Application to include an `ETL_Trans_Once` process (the name you choose for this process is arbitrary). Define this `transOnce` process in the same manner as `ETL_Trans_Only,` only include the `transOnce` runtime parameter on the command line. For example:

```
Command =   java -jar transform.jar -conf etl.properties
            -transOnce
```

Refer to the example in "Creating a New Process" on page 21, which provides a complete definition if you are creating a one-time transformation.

If an ongoing data transformation process is already running, be sure to stop it before running a one-time transformation process. You should not run these two processes simultaneously.

Running the Transformation Module Manually

To manually run the Transformation module, issue the following command from the directory in which ETL Runtime is installed:

```
java -jar transform.jar -conf [properties]
```

where

`[`*`properties`*`]` is the name of the file containing a listing of runtime parameters you can use to effect data transformation.

Refer to Table 8 on page 42 for a complete listing and description of these parameters.

Warning! Again, do not simultaneously run both one-time and continuous data transformation processes.

Chapter

6 Aggregation Module

The ETL Runtime Aggregation module logically pools data that was transferred from ODS to create aggregated and composite statistics for the defined aggregation levels. This module connects to the Data Mart to perform this task. You can run this module separately or in conjunction with the Transformation module, described beginning on page 61. The aggregation process organizes data and stores it in aggregate tables; the data is based on seven default aggregated time periods ranging from the 15-minute to the year level. Based on these tables, ETL Assistant creates report folders containing report views for each report layout in ODS.

This chapter includes these sections:

How ETL Runtime Aggregates Data

The Data Mart aggregation levels are presummarized fact tables containing data of different aggregation periods. This architecture optimizes report-generation performance because each report's underlying queries are run against their corresponding fact tables instead of against a larger single table. The ETL Runtime aggregation process derives its data for the following seven levels:

- *Hour-level data* is derived from aggregations of *15-minute data chunks;* hour-level aggregation is performed after the default number of 15-minute intervals (specified by `numberOfIntervalsToWaitFor_AGG_BY_HOUR`) has elapsed.

- *Day-level data* is derived from aggregations of hour-level data. Likewise, day-level aggregation is performed after the default number of hour intervals (specified by `numberOfIntervalsToWaitFor_AGG_BY_DAY`) has elapsed.
- *Week-* and *month-level data* is derived from aggregations of day-level data and is aggregated after the default number of day intervals (specified by `numberOfIntervalsToWaitFor_AGG_BY_WEEK` and `numberOfIntervalsToWaitFor_AGG_BY_MONTH` respectively) has elapsed.
- *Quarter-* and *year-level data* is derived from aggregations of month-level data and is aggregated after the default number of month intervals (specified by `numberOfIntervalsToWaitFor_AGG_BY_QUARTER` and `numberOfIntervalsToWaitFor_AGG_BY_YEAR`) has elapsed.

To obtain partial-period data, set the respective `numberOfIntervalsToWaitFor_` parameter to a value lower than the default. When data for the next interval arrives, ETL Runtime aggregates the previous interval and aggregates part of the next. For instance, setting the value of `numberOfIntervalsToWaitFor_AGG_BY_YEAR` to `1` directs ETL Runtime to maintain aggregated data for all past months of the current year. Year-level reports would then contain partial-year information.

Note: The `numberOfIntervalsToWaitFor_` parameters have significance only during the transformation-and-aggregation process. At each start of `trans_and_agg,` ETL aggregates at least one interval, if available, up to the number specified by the `numberOfIntervalsToWaitFor_` parameter.

The relationship that defines which longer intervals are aggregated on the basis of which shorter intervals forms a part of the metadata that is stored in the Data Mart, which you cannot change using ETL Assistant. You would have to alter Data Mart structure to effect new relationships, and Genesys discourages this.

ETL Runtime always aggregates data by applying a statistical aggregation function (addition, maximum, or minimum) to all data for shorter intervals whose starting times lie within the longer intervals. As a result, the collection interval should be an integral fraction of an hour. For instance, don’t base month-level aggregations on 30.4167 day intervals (365 days/12 months). On the other hand, you could base hour-level aggregations on any of the following:

- 30-minute intervals [60/30=2]
- 20-minute intervals [60/20=3]
- 15-minute intervals [60/15=4]

You should not base them on 45-minute intervals [60/45=1?], and, it makes little sense to make the collection interval greater than one hour.

ETL Runtime aggregates data regardless of the order that transformed data chunks appear in the Data Mart, ensuring correct results especially in those environments where the underlying data source (ODS) is populated from a

Data Sourcer application that operates under high availability—in this mode, Data Sourcer logs records with IDs that need not increase monotonically.

What Constitutes a Week?

The boundaries of 15-minute, hour, day, month, quarter, and year aggregation levels are very well defined within any given Gregorian year because each denomination represents an integral fraction of that year—there are four whole quarters in a year, 12 full months, 365 (or 366) complete days, and one and only one year in a year. No single hour splits in such a way that part of the hour resides in one year and the other part in the next, as is precisely the case for the beginning and/or ending weeks of any given year. Over and above this dual membership of each year, your system locale settings specify your preferred date-related conventions, which include the definition of a week and on which day the week begins. Different cultures observe different date conventions. As such, these variations in what constitutes a week merit special discussion for how ETL Runtime processes week-level aggregations.

ETL Runtime assigns week number 1 to the first week in a given year and stores this value in the `WEEK_N_IN_YEAR` column of the various time dimension tables in the Data Mart. But what constitutes Week 1 varies based partly on convention and partly on the configuration of your Data Mart application.

The Conventional First Week of A Year

By convention, the first week of a year in the United States and Russia countries always includes January 1, regardless of the day of the week January 1 turns out to be. The rest of the days that make up this first week depend on the day that is defined as the first day of the week in your environment. By convention, this first weekday is:

- Sunday in US-based locales.
- Monday in Russian-based locales.

Therefore, if your first weekday is Sunday and January 1 falls on a Saturday (as was the case for year 2005), then the conventional first week of the year, for US-based locales, consists of the seven days, December 26 (a Sunday) through January 1. The second week consists of the seven days, Sunday, January 2 through Saturday, January 8, and so forth.

Most European countries follow ISO 8601, which defines (among other things) the first week of a year as the week that includes the first Thursday of the new year; or put another way, the week that contains January 4. This standard also defines Monday to be the first day of a week.

The composition of Week 1 is the same in US, European, and Russian locales when this week contains both January 1 and January 4.

The Conventional Last Week of A Year

The conventional last week of a year can start as early as December 22nd (for US locales) and end as late as January 3rd of the subsequent year (for European locales). You can deduce which days make up this last week from the rules that determine the composition of the first week of the next year; namely, the last week of a year cannot contain:

- January 1 for US locales.
- January 4 for European (ISO 8601) locales.

This means that for European locales, January 1 can belong to the last week of the previous year if this day falls on a Friday, Saturday, or Sunday. For example, because January 1, 2005 fell on a Saturday, the latter half of the week, the seven days, December 27, 2004 through January 2, 2005, belong to the last week of 2004 (for European locales).

Data Mart Week 1

Data Mart weeks may differ from conventional weeks depending on how, or if, you set the following ETL Runtime parameters:

- `setFirstDayOfWeek`
- `wholeWeeks`
- `setMinimalDaysInFirstWeek`

setFirstDayOfWeek You can, in effect, override your locale's settings for the first day of a week, within ETL Runtime, by setting the `setFirstDayOfWeek` runtime parameter to your desired start day. The ETL Runtime processes that are deployed during the configuration of your Data Mart, using the Historical Reporting Wizard, all explicitly call this parameter, which sets Sunday as the first day of a week. To set up your environment so that it is ISO 8601-compliant, change this start day to Monday, but please refer to the description of this parameter, on page 53, for more details before doing so in an existing Reporting environment.

wholeWeeks Prior to the 7.0.2 release, not all Data Mart weeks consisted of seven days! Year boundaries split one week into two partial weeks belonging to two different years. So, any given year could start with one partial week and/or finish with another, thereby allowing some years to comprise 53 Data Mart weeks—51 normal weeks in the middle and two partial weeks at the beginning and end of the year.

Beginning with release 7.0.2, ETL Runtime provides the `wholeWeeks` runtime parameter enabling you to control an enforcement of seven-day weeks. The ETL Runtime default behavior, without the use of this parameter, splits a week along year boundaries as described above. The Data Mart Week 1 for 2011 (given a US-based locale that starts its week on Sunday and using partial weeks), for instance, is composed solely of one day—Saturday, January 1, 2011. Week 2 consists of the 7 days, January 2 through January 8, 2011; and Week 53 consists of the 7 days, December 25 through December 31, 2011.

When you use the `wholeWeeks` parameter, Data Mart Week 1 for 2011 consists of the following:

- The 7 days from December 26, 2010 through January 1, 2011 for US locales.
- The 7 days from January 2, 2011 through January 9 for European locales.

setMinimalDaysIn FirstWeek

In conjunction with use of the `wholeWeeks` parameter, you can specify how many days of the new year should appear in your first Data Mart week. Set this parameter to `1` (so that January 1st is always included) to follow US convention and `4` to follow ISO 8601 (so that January 4 is always included). Refer to the description of this parameter, on page 53, for additional details.

Table 12 provides several examples to illustrate what constitutes the first Data Mart week given different settings of the `setFirstDayOfWeek` and `wholeWeeks` runtime parameters in US- and European-based locales.

Table 12: The First Week of A Year

January 1 Is A ...	setFirstDay OfWeek	whole Weeks	Data Mart Week 1 Consists of...	
			Europe	US
Sunday (as for 2012, 2017)	SAT	Yes	31 1 2 3 4 5 6	same
		No	□ 1 2 3 4 5 6	same
	SUN	Yes	1 2 3 4 5 6 7	same
		No	1 2 3 4 5 6 7	same
	MON	Yes	2 3 4 5 6 7 8	26 27 28 29 30 31 1
		No	□ □ □ □ □ □ 1	same
Monday (as for 2018)	SAT	Yes	30 31 1 2 3 4 5	same
		No	□ □ 1 2 3 4 5	same
	SUN	Yes	31 1 2 3 4 5 6	same
		No	□ 1 2 3 4 5 6	same
	MON	Yes	1 2 3 4 5 6 7	same
		No	1 2 3 4 5 6 7	same
Tuesday (as for 2013, 2019)	SAT	Yes	29 30 31 1 2 3 4	same
		No	□ □ □ 1 2 3 4	same
	SUN	Yes	30 31 1 2 3 4 5	same
		No	□ □ 1 2 3 4 5	same
	MON	Yes	31 1 2 3 4 5 6	same
		No	□ 1 2 3 4 5 6	same

Table 12: The First Week of A Year (Continued)

January 1 Is A ...	setFirstDay OfWeek	whole Weeks	Data Mart Week 1 Consists of...	
			Europe	US
Wednesday (as for 2014, 2020)	SAT	Yes	4 5 6 7 8 9 10	28 29 30 31 1 2 3
		No	1 2 3	same
	SUN	Yes	29 30 31 1 2 3 4	same
		No	1 2 3 4	same
	MON	Yes	30 31 1 2 3 4 5	same
		No	1 2 3 4 5	same
Thursday (as for 2015)	SAT	Yes	3 4 5 6 7 8 9	27 28 29 30 31 1 2
		No	1 2	same
	SUN	Yes	4 5 6 7 8 9 10	28 29 30 31 1 2 3
		No	1 2 3	same
	MON	Yes	29 30 31 1 2 3 4	same
		No	1 2 3 4	same
Friday (as for 2016)	SAT	Yes	2 3 4 5 6 7 8	26 27 28 29 30 31 1
		No	1	same
	SUN	Yes	3 4 5 6 7 8 9	27 28 29 30 31 1 2
		No	1 2	same
	MON	Yes	4 5 6 7 8 9 10	28 29 30 31 1 2 3
		No	1 2 3	same
Saturday (as for 2011)	SAT	Yes	1 2 3 4 5 6 7	same
		No	1 2 3 4 5 6 7	same
	SUN	Yes	2 3 4 5 6 7 8	26 27 28 29 30 31 1
		No	1	same
	MON	Yes	3 4 5 6 7 8 9	27 28 29 30 31 1 2
		No	1 2	same

The Data Mart Last Week of A Year

The last Data Mart week can differ from the conventional last week of a year given different settings of the `setFirstDayOfWeek`, `wholeWeeks`, and `setMinimalDaysInFirstWeek` runtime parameters. Table 13 provides several examples of the composition of this week of a year given different settings of these parameters in US- and European-based locales.

Table 13: The Last Week of A Year

December 31 Is A ...	setFirst DayOf Week	whole Weeks	Last Data Mart Week Consists of ...			
			Week #		Europe	US
			Eur	US		
Sunday (as for 2006)	SAT	Yes	52	...	23 24 25 26 27 28 29	same
		No	53	...	30 31	same
	SUN	Yes	52	...	24 25 26 27 28 29 30	same
		No	53	...	31	same
	MON	Yes	52	53	25 26 27 28 29 30 31	same
		No	53	...	25 26 27 28 29 30 31	same
Monday (as for 2007)	SAT	Yes	52	...	22 23 24 25 26 27 28	same
		No	53	...	29 30 31	same
	SUN	Yes	52	...	23 24 25 26 27 28 29	same
		No	53	...	30 31	same
	MON	Yes	52	...	24 25 26 27 28 29 30	same
		No	53	...	31	same
Tuesday (as for 2002*)	SAT	Yes	53	52	28 29 30 31 1 2 3	21 22 23 24 25 26 27
		No	53	...	28 29 30 31	same
	SUN	Yes	52		22 23 24 25 26 27 28	same
		No	53	...	29 30 31	same
	MON	Yes	52	...	23 24 25 26 27 28 29	same
		No	53	...	30 31	same
Wednesday (as for 2008)	SAT	Yes	53	52	27 28 29 30 31 1 2	20 21 22 23 24 25 26
		No	53	...	27 28 29 30 31	same
	SUN	Yes	53	52	28 29 30 31 1 2 3	21 22 23 24 25 26 27
		No	53	...	28 29 30 31	same
	MON	Yes	52	...	22 23 24 25 26 27 28	same
		No	53	...	29 30 31	same

Table 13: The Last Week of A Year (Continued)

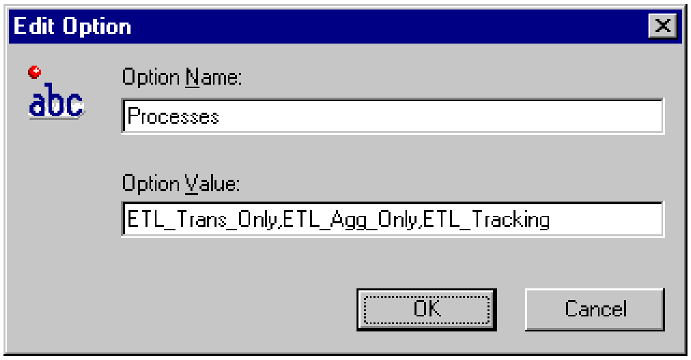
December 31 Is A ...	setFirst DayOf Week	whole Weeks	Last Data Mart Week Consists of ...			
			Week #		Europe	US
			Eur	US		
Thursday (as for 2009)	SAT	Yes	53	52	26 27 28 29 30 31 1	19 20 21 22 23 24 25
		No	53	...	26 27 28 29 30 31 □	same
	SUN	Yes	53	52	27 28 29 30 31 1 2	20 21 22 23 24 25 26
		No	53	...	27 28 29 30 31 □ □	same
	MON	Yes	53	52	28 29 30 31 1 2 3	21 22 23 24 25 26 27
		No	53	...	28 29 30 31 □ □ □	same
Friday (as for 2010)	SAT	Yes	53	53	25 26 27 28 29 30 31	same
		No	53	...	25 26 27 28 29 30 31	same
	SUN	Yes	52	52	26 27 28 29 30 31 1	19 20 21 22 23 24 25
		No	53	...	26 27 28 29 30 31 □	same
	MON	Yes	53	52	27 28 29 30 31 1 2	20 21 22 23 24 25 26
		No	53	...	27 28 29 30 31 □ □	same
Saturday (as for 2005, 2011)	SAT	Yes	52	...	24 25 26 27 28 29 30	same
		No	53	...	31 □ □ □ □ □ □	same
	SUN	Yes	52	...	25 26 27 28 29 30 31	same
		No	53	...	25 26 27 28 29 30 31	same
	MON	Yes	53	52	26 27 28 29 30 31 1	19 20 21 22 23 24 25
		No	53	...	26 27 28 29 30 31 □	same

Weekend

For all locales, ETL Runtime interprets settings for Saturday and Sunday as the weekend.

Scheduling Ongoing Aggregation

If you use the `Genesys ETL Service Starter` to start ETL Runtime processes, a continuous data aggregation process is already scheduled. The process name is `ETL_Agg_Only` as depicted in Figure 10.

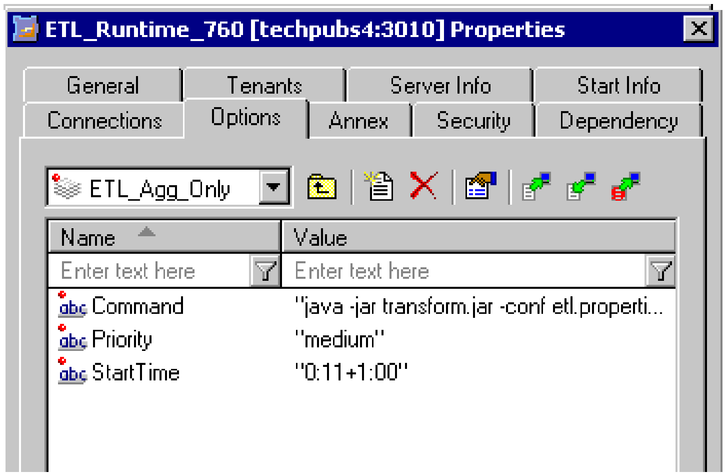


Option Name: Processes

Option Value: ETL_Trans_Only,ETL_Agg_Only,ETL_Tracking

Figure 10: The ETL Service Starter Includes ETL_Agg_Only

Figure 11 shows this process as defined in Configuration Manager.



Name	Value
Command	"java -jar transform.jar -conf etl.properti...
Priority	"medium"
StartTime	"0:11+1:00"

Figure 11: ETL_Agg_Only Process Definition

The cutaway crops the value for `Command`, of which the following is the complete definition:

```
java -jar transform.jar -conf etl.properties -aggOnly
```

If you used the Historical Reporting Wizard to configure your `Data Mart` Application object, the Aggregation module is automatically scheduled to run 11 minutes past midnight and every hour thereafter. The Wizard also configures your `etl.properties` file with the values you specified for the `CfgAppName`, `CfgAppPassword`, `CfgUser`, and `CfgUserPasswordEncrypted` runtime parameters for connection to your Configuration Server.

Note: The Wizard no longer writes to the `etl.properties` file, values specifying connection parameters to your Data Mart, including `user`, `pass`, and `jdbcurl`. Instead, the starter application adds these parameters and their values to the command line immediately prior to starting a Data Mart application.

Running the Aggregation Module Manually

Issue the following command from the directory in which ETL Runtime is installed:

```
java -jar transform.jar -conf etl.properties -aggOnly
```

where [*properties*] is the option list of runtime parameters you can use to effect data aggregation.

Refer to Table 8 on page 42 for a complete listing and description of aggregation and other parameters.

Running Transformation and Aggregation Consecutively

To run the Transformation module followed by the Aggregation module, remove `-transOnly` and/or `-aggOnly` from the command line and add `-agg`:

```
java -jar transform.jar -conf [properties] -agg
```

where [*properties*] is the option list of runtime parameters you can use to effect data transformation and aggregation.

You should note that the available runtime parameters are not the union of transformation-only and aggregation-only parameters. Refer to Table 8 on page 42 for a complete listing and description of transformation-and-aggregation parameters.

Chapter

7 Configuration Tracking Module

The ETL Runtime Configuration Tracking module tracks the relationship between certain objects and their respective groups in the Configuration Server. This tracking can be helpful in generating reports containing information about only those agents who are included in the Customer Service group, for example. In the following sections, this chapter describes the Configuration-Tracking module of ETL Runtime:

The Objects That ETL Runtime Tracks

The Configuration Tracking module connects asynchronously to Configuration Server to track the addition and deletion of objects belonging to the following group objects:

- Group of Agents—ETL tracks the agents included in this group. ETL Runtime does not differentiate virtual agent groups from explicitly-defined agent groups.
- Group of Places—ETL tracks the places included in this group.
- Group of DNs of type `Routing Point`—ETL tracks the routing points, virtual routing points, and routing queues included in the group of directory numbers.
- Group of DNs of type `Queue`—ETL tracks the queues and virtual queues included in the group of directory numbers.

If ETL Runtime is unable to connect to Configuration Server, ETL Runtime does not attempt to reconnect to the Server until its next scheduled time (see "Scheduling Configuration Tracking" on page 77). Under these circumstances, ETL Runtime does not attempt to connect to the backup Configuration Server, if one has been specified. However, you can set up your Data Mart application to initially connect to a backup Configuration Server using the Solution Control Interface if you so desire.

ETL Runtime stores this tracking information in the OBJ_TO_OBJ table of the Data Mart, which contains two columns—PARENT_ID and CHILD_ID—that refer to an entry in the OBJECT table. PARENT_ID refers to the group object, such as Group of Agents; CHILD_ID refers to the objects belonging to this group (Agents in this example).

In the same fashion, ETL Runtime tracks the skills assigned to agents.

Since you can add and remove objects from a group and reassign agents to different skills, the OBJ_TO_OBJ table maintains a history of changes.

When ETL Runtime detects that you have added an object to a group or assigned (or reassigned) a skill to an agent, the Configuration-Tracking module inserts a record in the OBJ_TO_OBJ table, setting ADD_TIME to the time of detection and DELETE_TIME to NULL.

Note: Because ETL Runtime is not a real-time application, the time can indicate a later time than when the object is actually added to the particular group. If ETL Runtime runs every day, for example, the precision of ADD_TIME will be correct to the day but hours and minutes will be incorrect.

When the Configuration-Tracking module detects that you have deleted an object from a group or removed a skill assignment for an agent, it marks the location as deleted by setting DELETE_TIME to the time for which the object was detected for deletion.

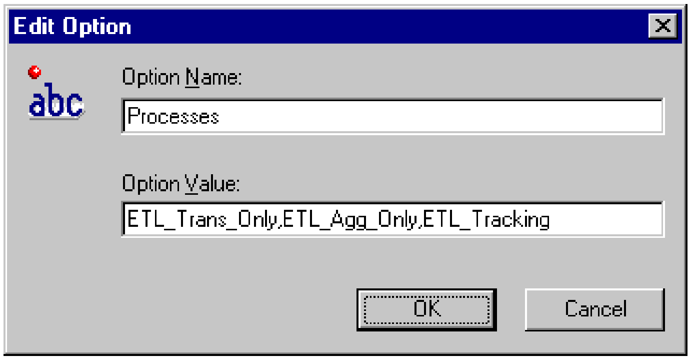
Note: Again, since ETL Runtime is not a real-time application, DELETE_TIME can indicate a later time compared to the actual moment that the object was deleted from the particular group.

If an object is added to the group or a skill is assigned to an agent later, a new entry is created in the OBJ_TO_OBJ table. As a result, the database keeps a history of relationships among objects/groups and agents/skills, making it possible to find out the following:

- How many times an object is included in a particular group, when it entered the group, and when it left the group.
- When different skills are assigned to agents.

Scheduling Configuration Tracking

If you use the Genesys ETL Service Starter to start ETL Runtime processes, configuration tracking is already scheduled. The process name is ETL_Tracking as depicted in Figure 12.

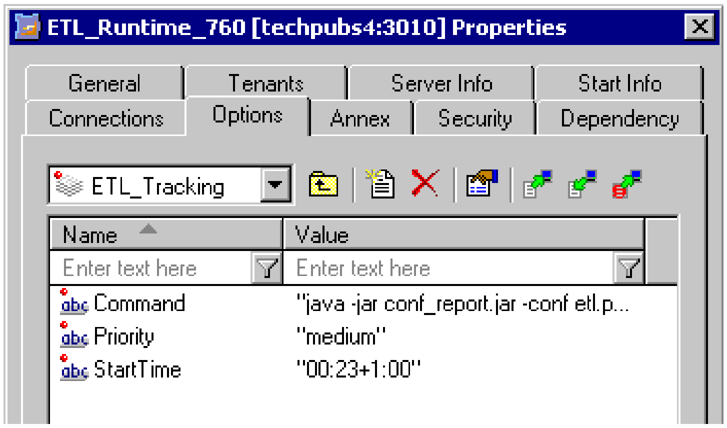


Option Name: Processes

Option Value: ETL_Trans_Only,ETL_Agg_Only,ETL_Tracking

Figure 12: The ETL Service Starter Includes ETL_Tracking

Figure 13 shows how this process is defined in Configuration Manager.



ETL_Tracking

Name	Value
Command	"java -jar conf_report.jar -conf etl.p...
Priority	"medium"
StartTime	"00:23+1:00"

Figure 13: ETL_Tracking Process Definition

If you used the Historical Reporting Wizard to configure your Data Mart Application object, the Configuration-Tracking module is automatically scheduled to run 23 minutes past midnight and every hour thereafter. The Wizard also configures your etl.properties file with the values you specified for the CfgAppName, CfgAppPassword, CfgUser, and CfgUserPasswordEncrypted runtime parameters.

Note: The Wizard no longer writes to the etl.properites file values specifying connection parameters to your Data Mart, including user, pass, and jdbcurl.

Either edit this file to contain the appropriate parameters and their values or specify them at the `Command` option of the `ETL_Tracking` process. For example:

```
java -jar conf_report.jar -conf etl.properties -user DU
-pass DP -jdbcurl DJ ...
```

where *DU*, *DP*, and *DJ* are values specific to your Data Mart.

Running the Configuration Tracking Module Manually

If you run ETL Runtime modules manually (or use a third-party scheduler), you should run the Configuration Tracking module as often as object-to-group changes are made in the Configuration Server—for example, when an agent is moved from the Customer Service group to the Technical Support group, when a new directory number is added to a switch, when the composition of virtual groups change, or when an infrequently used queue is removed from the queue group.

To manually run the Configuration Tracking module, issue the following command from the directory in which ETL Runtime is installed:

```
java -jar conf_report.jar [properties]
```

where `[`*`properties`*`]` is the option list of runtime parameters you can use to effect configuration tracking.

Refer to Table 8 on page 42 for a complete listing and description of these parameters.

The Data Mart installation process does not deploy a batch file for manually running the Configuration Tracking module.

Chapter

8

Tenant Alias Tracking Module

The ETL Runtime Tenant Alias Tracking (TAT) module facilitates the setup of views for new and modified tenants within your Oracle, Sybase, Microsoft SQL Server, or DB2 environment by creating and/or using aliases instead of literal tenant names. Executing in the TAT mode:

- Updates synonyms in ETL Runtime for your tenants.
- Creates separate Oracle, Microsoft SQL, and Sybase accounts, where appropriate, for each tenant.
- Grants the appropriate database permissions to each tenant.
- Consolidates data from multiple sources into one canned or custom report. This consolidation is particularly useful for resolving scalability issues, such as when you operate several ODSs in different time zones.

Run the TAT module any time you make tenant-level changes in the Configuration Database. You do not need to stop any other Reporting component to do so.

This chapter has the following sections:

Why Use Tenant Aliases?

Aliases allow you to use the same queries and Crystal Report report layouts in a multi-tenant environment. Each account uses aliases (or views, depending on the database engine—that point to the aliased tables that contain report data. As a tenant, you see and use the same table names as other tenants, but the content of generated reports describes your own activity. Tenants can view the report layouts and folders of other tenants but they cannot access the tables or the data these report layouts and report folders are based on.

Table names follow this convention:

```
dimension_tname_agglevel
```

where

- `dimension` is one of [`S` | `O` | `T` | `V`] (stat, object, time, value dimension).
- `tname` is the name of the layout template as defined in DMA.
- `agglevel` is one of [`DAY` | `HOUR` | `MONTH` | `WEEK` | `QUARTER` | `YEAR` | `NO_AGG`].

Because of the naming convention that is used, do not run the TAT module in two or more Data Marts under the following circumstance:

- One Configuration Server environment contains two or more Data Marts.
- These Data Marts are each created in the same relational database management system.
- Each Data Mart stores information for two or more of the same tenants; for example, data for both `TenantA` and `TenantB` resides in both `DataMart`$_i$ and `DataMart`$_j$.

Under this circumstance, running the TAT module for `DataMart`$_i$, for example, uses the same tenant account name that is created when the TAT module is run for `DataMart`$_j$.

Tenant User Names

The `tenants_alias_update.log` file logs information about created accounts, tenant aliases, tenant user names, and tenant passwords. You cannot rename TAT-assigned tenant user names. These names follow these naming conventions:

```
"tenant_" & ConfigServerID & "_" & Tenant_DBServerID
```

or

`"t" & ConfigServerID & "_" & Tenant_DBServerID` when you have set a value for the `tenants_shortcut` option.

For example, `tenant_248_101` or `t248_101` describes tenant 101 in the 248 Configuration Server.

Use your own tenant account to run the solution-provided canned reports. Even though you can operate the Genesys Report Generation Assistant from any tenant or Data Mart owner account, Genesys does not recommend that the DBA share the Data Mart's owner login information with tenants. Furthermore, you can only access ETL Assistant from the Data Mart owner's account.

Note: For DB2 only, you must create the corresponding operating system accounts for each tenant before running the TAT module. These accounts must strictly adhere to the stated naming convention. In addition, you must set a value for the `tenants_shortcut` option for DB2 Data Marts on any platform.

Algorithm for Tenant Passwords

The TAT module generates passwords for the tenant accounts it creates. The algorithm that is used differs for each DBMS type as depicted in Table 14. Using standard DBA tools, you can change tenant passwords provided that you do not assign the same password to tenants bearing identical names in different Configuration Servers.

Note: As DB2 does not maintain database accounts for each tenant, you must create O/S accounts that strictly adhere to the naming convention described in "Tenant User Names" before running the TAT module.

Table 14: Algorithm for Determining Tenant Passwords

<table>
<tr><td colspan="4"><b>Recommendation:</b> In the <code>tenants_alias_update.properties</code> file, set <code>tenants_shortcut</code> to the following:</td></tr>
<tr><td><b>MS SQL:</b> <code>OFF</code></td><td><b>ORCL:</b> <code>OFF</code></td><td><b>SYB:</b> <code>OFF</code></td><td><b>DB2</b>: <code>ON</code></td></tr>
<tr><td colspan="4">1. TAT takes the password that is assigned in the Configuration Server.</td></tr>
<tr><td><b>MS SQL:</b> <code>True</code></td><td><b>ORCL:</b> <code>True</code></td><td><b>SYB:</b> <code>True</code></td><td><b>DB2</b>: <code>False</code></td></tr>
<tr><td colspan="4">2. If no password has been assigned in the Configuration Server or if the password contains special characters, TAT uses its generated account name as the password for the account.</td></tr>
<tr><td><b>MS SQL:</b> <code>True</code></td><td><b>ORCL:</b> <code>True</code></td><td><b>SYB:</b> <code>True</code></td><td><b>DB2</b>: <code>False</code></td></tr>
<tr><td colspan="4">3. If the Configuration Server password begins with a digit, TAT uses its generated account name as the password for the account.</td></tr>
<tr><td><b>MS SQL:</b> <code>True</code></td><td><b>ORCL:</b> <code>True</code></td><td><b>SYB:</b> <code>True</code></td><td><b>DB2</b>: <code>False</code></td></tr>
<tr><td colspan="4">4. If the Configuration Server password length is greater than eight characters and <code>tenants_shortcut=on</code> appears in the <code>tenant_alias_ update.properties</code> file, TAT uses its generated account name as the password for the account.</td></tr>
<tr><td><b>MS SQL:</b> <code>True</code></td><td><b>ORCL:</b> <code>True</code></td><td><b>SYB:</b> <code>True</code></td><td><b>DB2</b>: <code>False</code></td></tr>
</table>

Table 14: Algorithm for Determining Tenant Passwords (Continued)

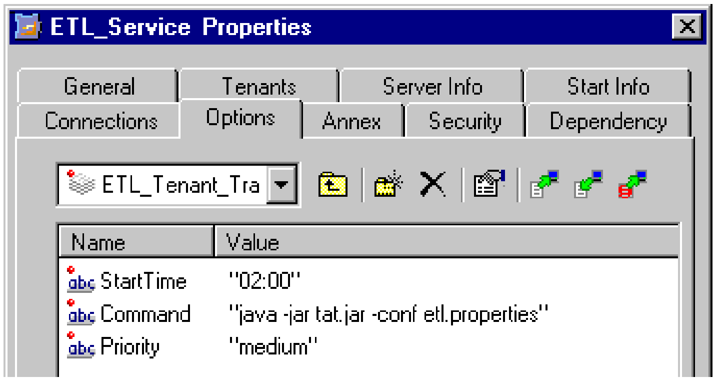
5. If the Configuration Server password length is greater than 30 characters, TAT uses its generated account name as the password for the account.			
MS SQL: `True`	**ORCL:** `True`	**SYB:** `True`	**DB2**: `False`
6. If the Configuration Server password is shorter than the database's minimum password length parameter, TAT uses its generated account name as the password for the account. If the tenant account name is less than this parameter, the password is padded at the end with zeros to achieve minimum length.			
MS SQL: `True`	**ORCL:** `True`	**SYB:** `True`	**DB2**: `False`
7. If `reverse_password=on` appears in the `tenant_alias_update.properties` file, the password is reversed.			
MS SQL: `True`	**ORCL:** `True`	**SYB:** `True`	**DB2:** `False`

Note: Your RDBMS must permit the password lengths you set for each tenant in Configuration Server. The Sybase server, for example, may not accept passwords that are less than five characters in length.

Scheduling the TAT Module

The default configuration of the `Genesys ETL Service Starter` neither invokes the TAT module nor creates a process section for automatic scheduled runs. You must define a new process to the `Genesys ETL Service Starter` application to accomplish this.

Figure 14 illustrates how you might define the `ETL_Tenants_Tracking` process (the name you choose is arbitrary).

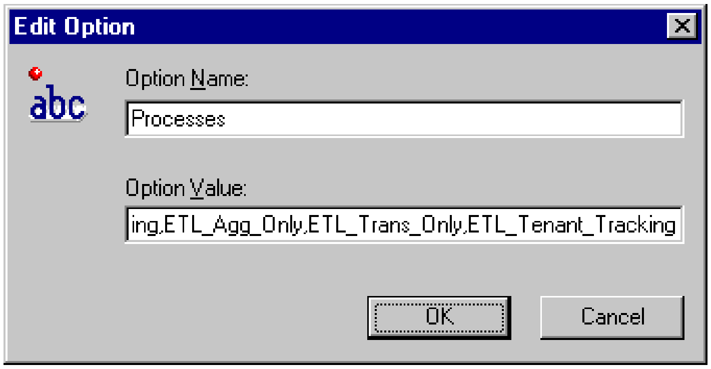


Name	Value
StartTime	"02:00"
Command	"java -jar tat.jar -conf etl.properties"
Priority	"medium"

Figure 14: The Tenant Alias Tracking Process

To schedule the process so that it runs whenever `Genesys ETL Service Starter` runs, add the process to the `Genesys ETL Service Starter` section:

1. From the `Options` tab of the Data Mart `Application` object `Properties` dialog box, open the `Genesys ETL Service Starter` section.
2. Double-click `Processes` to open the `Edit Option` dialog box.
3. Add the process to the `Option Value` string as shown in Figure 15 and click `OK`.



Option Name: Processes

Option Value: ing,ETL_Agg_Only,ETL_Trans_Only,ETL_Tenant_Tracking

Figure 15: Scheduling Tenant Alias Tracking Using ETL Service Starter

Optionally, you can use a third-party scheduler to run the TAT module if you prefer to run the module less frequently than once a day.

Running the TAT Module Manually

Issue the following command from the directory in which you installed ETL Runtime:

```
java -jar tat.jar -conf [properties]
```

where `[`*`properties`*`]` is the option list of runtime parameters you can use to affect this module.

Note: You can only run the TAT module from the Data Mart account.

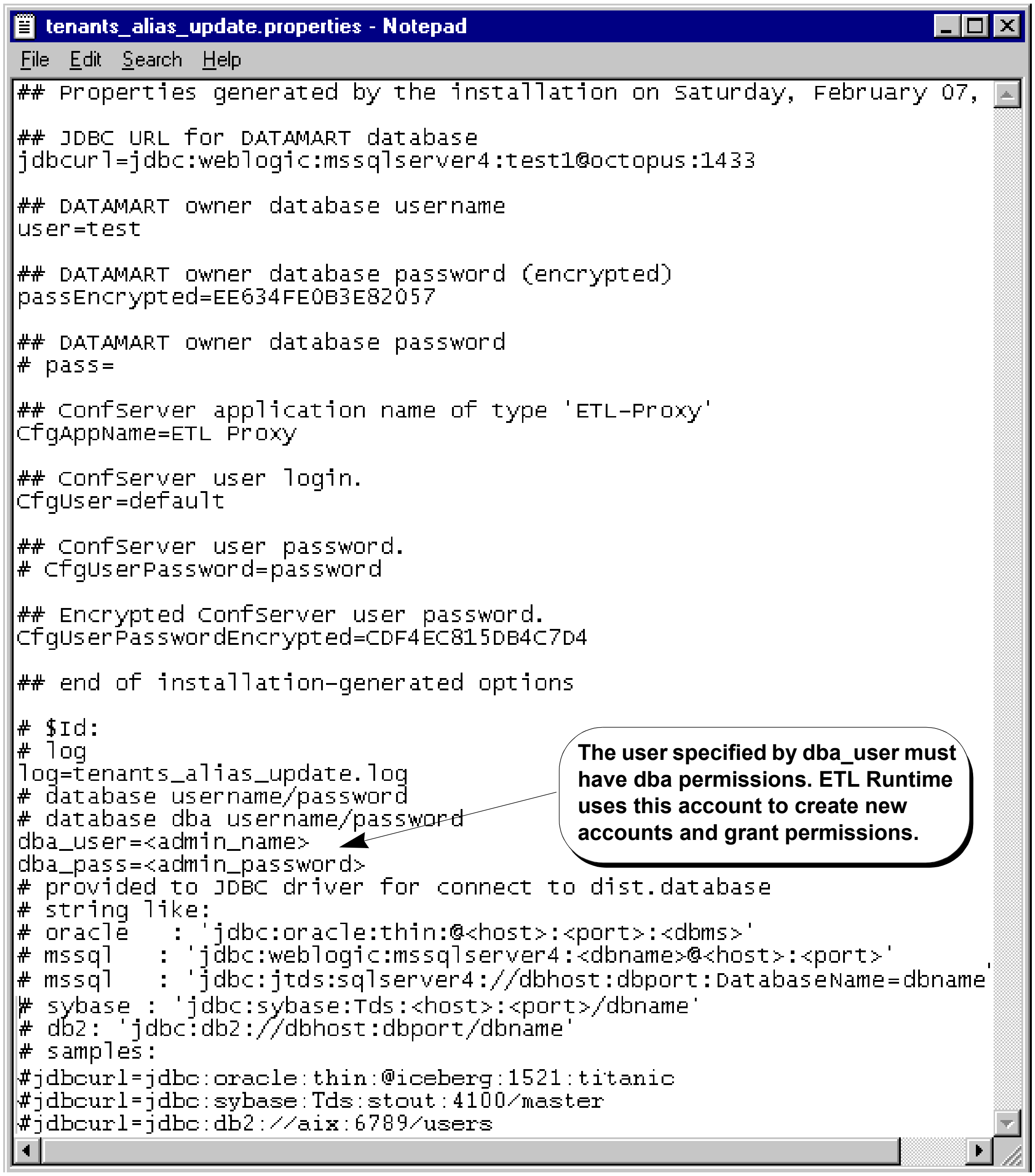
Refer to Table 8 on page 42 for a complete listing and description of these parameters.

If you have set up the `tenants_alias_update.properties` file (a cutaway is shown in Figure 16), you could issue the following command instead:

- `run-tenants_alias_update.bat` (on Microsoft Windows systems)
- `run-tenants_alias_update` (on Unix systems)

Note that the Historical Reporting Wizard populates only the `etl.properties` file with the values for the `user`, `pass`, and `jdbcurl` parameters that you supplied during configuration. Be sure to include values for these three parameters in the `tenants_alias_update.properties` file before running it.

In Figure 16, for example, a Microsoft SQL Data Mart, named `test1,` has been set aside on `octopus` (the host) at port `1433` (the default port for Microsoft SQL). The Data Mart owner’s name is `test` and the corresponding password is encrypted. You must specify the values for additional parameters such as `dba_user` and `dba_pass` to complete the setup of this file.



```
## Properties generated by the installation on Saturday, February 07,

## JDBC URL for DATAMART database
jdbcurl=jdbc:weblogic:mssqlserver4:test1@octopus:1433

## DATAMART owner database username
user=test

## DATAMART owner database password (encrypted)
passEncrypted=EE634FE0B3E82057

## DATAMART owner database password
# pass=

## ConfServer application name of type 'ETL-Proxy'
CfgAppName=ETL Proxy

## ConfServer user login.
CfgUser=default

## ConfServer user password.
# CfgUserPassword=password

## Encrypted ConfServer user password.
CfgUserPasswordEncrypted=CDF4EC815DB4C7D4

## end of installation-generated options

# $Id:
# log
log=tenants_alias_update.log
# database username/password
# database dba username/password
dba_user=<admin_name>
dba_pass=<admin_password>
# provided to JDBC driver for connect to dist.database
# string like:
# oracle   : 'jdbc:oracle:thin:@<host>:<port>:<dbms>'
# mssql    : 'jdbc:weblogic:mssqlserver4:<dbname>@<host>:<port>'
# mssql    : 'jdbc:jtds:sqlserver4://dbhost:dbport:DatabaseName=dbname'
# sybase : 'jdbc:sybase:Tds:<host>:<port>/dbname'
# db2: 'jdbc:db2://dbhost:dbport/dbname'
# samples:
#jdbcurl=jdbc:oracle:thin:@iceberg:1521:titanic
#jdbcurl=jdbc:sybase:Tds:stout:4100/master
#jdbcurl=jdbc:db2://aix:6789/users
```

The user specified by dba_user must have dba permissions. ETL Runtime uses this account to create new accounts and grant permissions.

Figure 16: tenant_alias_update.properties File

Architectural Change

Prior to release 6.5, running reports from the Data Mart account displayed data that was retrieved from the *first report layout encountered* **belonging to the appropriate tenant**. Running reports from a tenant account displayed data that retrieved from all similar report layouts belonging to that tenant. For this reason, Genesys recommends running reports from the tenant account for pre-6.5 releases.

Note: The term *similar* as used here refers to those report layouts that are based on identically named layout templates. (Layout templates are not shared among tenants, who have their own set of layout templates. These templates may be named alike though.)

In 6.5 and later releases, ETL Runtime behavior was changed for the Data Mart account to widen the scope of data displayed. In 6.5$^+$ releases, reports generated from the Data Mart account display data that was retrieved from *all similar report layouts* belonging to the appropriate template *for all tenants.* Genesys recommends that the elements of metagroups on which similar report layouts are based *do not overlap* in time. For example, agent John Smith, should not be defined to two or more tenants—even if this represents two different people. Agent, Place, Queue, and Route-Point objects should be uniquely named across all tenants. Likewise, agent John Smith should not appear both in the `Gold Account` and `Platinum Account` metagroups, for example, where these metagroups form the basis for agent-object report layouts.

On this subject, reports generated from tenant accounts display the same results in release 6.5+ as they did in prior releases. Reports that are generated from the Data Mart reflect a difference as illustrated below:

Given	**Tenant 1**	**Tenant 2**
	John Smith 0 0 0...	John Doe 0 0 0...

A report generated from a pre-6.5 Data Mart might show either of the following:

```
John Smith 0 0 0...
```

or

```
John Doe 0 0 0...
```

depending on which report layout Data Mart encounters first.

In 6.5 and later releases, the same report generates both records:

```
John Smith 0 0 0...
John Doe 0 0 0...
```

Genesys

Chapter

9 Purging Module

The ETL Runtime Purging module permanently deletes data from Data Mart tables using either a default set of rules or the purging rules you specify using ETL Assistant. You specify purge rules at report-view level, which is a combination of report folder (or report layout) and aggregation level. You could, for instance, have ETL Runtime delete tables related to queues containing data that is older than three months, but still keep data for routing points for the same time period. You can schedule this module using ETL Service Starter, run it stand-alone, or run it using a third-party scheduler. The following chapter sections describe the Purging module:

Scheduling the Purging Module

The simultaneous running of other activities against the Data Mart (such as data aggregation or transformation) and the Purging module may slow down the purge operation and increase demand for the temporary database space to keep transactions. For this reason, Genesys recommends that you schedule the Purging module during nonpeak hours.

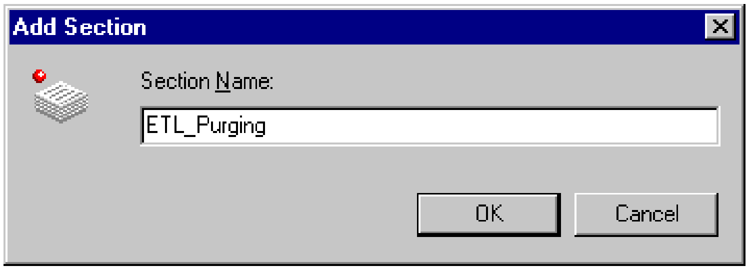
The default configuration of `Genesys ETL Service Starter` does not invoke the Purging module. You must create a process (if you have not already done so) and add it to the Starter definition for automatic scheduling.

To create a purge process:

1. Within Configuration Manager, open the `Properties` dialog box of the Data Mart `Application` object.
2. Click the `Options` tab.

3. Right-click within the blank area of the list view and select `New` from the context menu that appears.
4. In the `Add Section` dialog box that opens, name your new process, and then click `OK`. (Process names must not contain commas or spaces and must be less than 255 characters in length.)

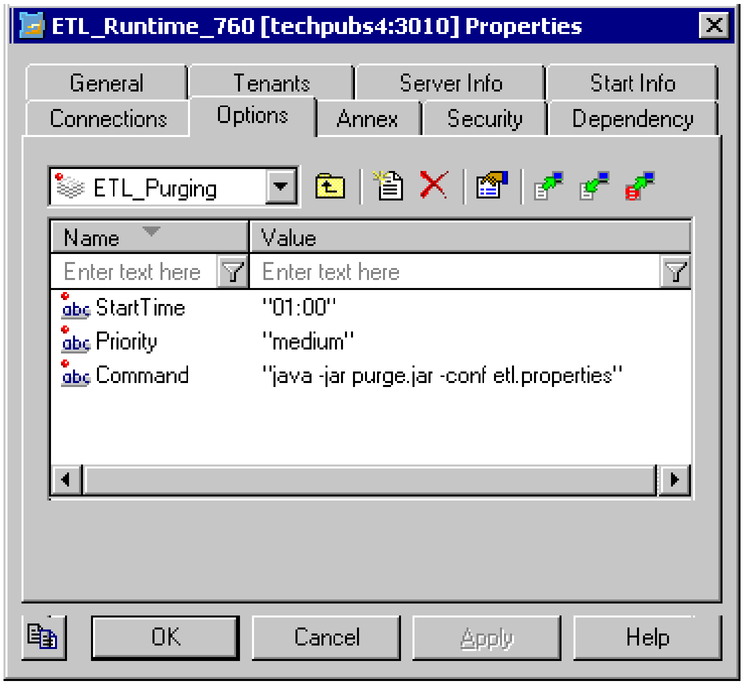
   Figure 17 shows `ETL_Purging` as the process (section) name.



Section Name: ETL_Purging

Figure 17: Setting Up a Purging Process

5. Within the `Options` list view, double-click the name you just created and specify `Priority`, `Command`, and `StartTime` options as appropriate for your environment. The configuration in Figure 18, for example, invokes the Purging module every morning at 1:00 AM.



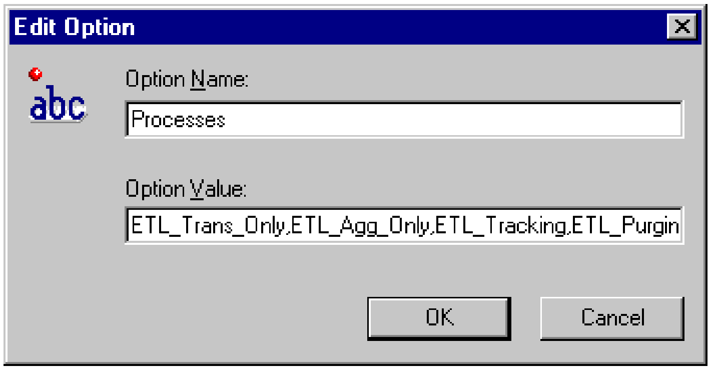
Name	Value
StartTime	"01:00"
Priority	"medium"
Command	"java -jar purge.jar -conf etl.properties"

Figure 18: Defining Option Values for a Purge Process

If you use a properties file, as depicted in Figure 18, be sure it specifies `user`, `passEncrypted` (or `pass`), and `jdbcurl` runtime parameters for connection to your Data Mart.

Now, you can set up ETL Runtime to schedule this process. To add the Purging process to the `ETL Service Starter` section, do the following:

1. From the `Options` tab of the Data Mart `Application` object `Properties` dialog box, open the `Genesys ETL Service Starter` section.
2. Double-click `Processes` to open the `Edit Options` dialog box.
3. Add the new process you just created to the `Option Value` string as shown in Figure 19. Then click `OK`.



Option Name: Processes

Option Value: ETL_Trans_Only,ETL_Agg_Only,ETL_Tracking,ETL_Purgin

Figure 19: Scheduling Purging Using ETL Service Starter

Optionally, you can use a third-party scheduler to run the Purge process.

Running the Purging Module Manually

Issue the following command from the directory in which you installed ETL Runtime:

```
java -jar purge.jar -conf [propertiesfile] -user x -pass y -jdbcurl z
```

where

x is the user name of the Data Mart owner.
y is his or her password.
z is the URL for the JDBC driver.

If you have set up the `etl.properties` or `purge.properties` files to provide these values you could issue the following command instead:

```
java -jar purge.jar -conf [etl|purge].properties
```

Or simply issue:

- `run-purge.bat` (on Microsoft Windows systems).
- `run-purge` (on Unix systems).

Which properties file you use, or even its name, is not significant so long as it contains the runtime parameters that are necessary for connection to your Data Mart. Properties files are placed in the `ETL_Runtime` folder that is created during ETL Runtime installation. If you used the Historical Reporting Wizard to configure your Data Mart `Application` object, the Wizard updates `etl.properties` to specify these parameters automatically. You must, however, edit `purge.properties` if you wish to use it instead. The batch files point to `purge.properties`, but you can edit the batch file as well.

See Chapter 4 beginning on page 41 for more information about the `user`, `passEncrypted` (or `pass`), and `jdbcurl` runtime parameters. Also refer to "Starting ETL Runtime's Modules Individually" in the *Reporting Deployment Guide* for additional information.

Defining Your Own Purge Rules

The default rules enable you to purge data for all aggregation levels after two complete calendar years have elapsed. For example, if today were July 30, 2012, all data prior to January 1, 2010, would be eligible for purge using the default schedule because as of 7/30/12, 2012 does not represent a complete calendar year. The Purging module looks to 2011 and 2010 as the complete calendar years before which data becomes eligible for purge. Note that data eligibility for purge is not the same thing as data purging. The actual purge operation occurs when you invoke the Purging module. You must use ETL Assistant to set your own rules for each report view.

Note that for the Purging module, log output is always directed to `STDOUT`, *not* to the log value that is specified in the `all` configuration option in ETL Runtime.

If you have set no rules using ETL Assistant and if the Purging module is invoked, the ETL Runtime default purge behavior for all aggregation levels is to purge data after two years have elapsed with the level of granularity set at one year.

Purge Algorithm

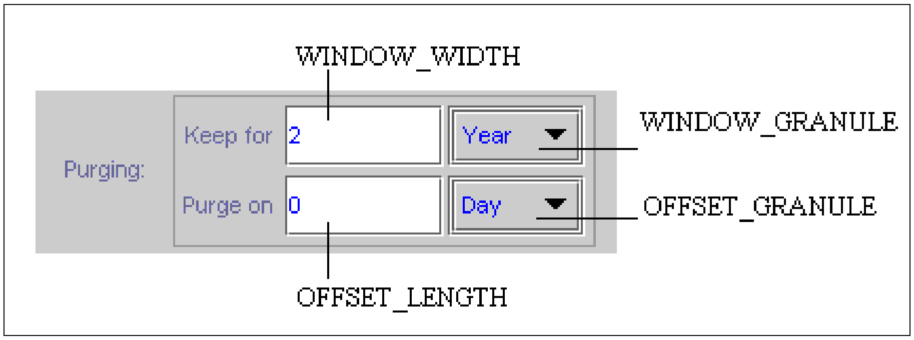
If scheduled within the ETL Runtime Starter, ETL Runtime automatically calls a purging procedure each night. The procedure reads the `PURGING_RULES` table in the Data Mart and determines whether the user-defined amount of time has elapsed for data to be purged from each report view. A default set of rules applies where you have not explicitly defined them.

To correctly calculate metrics in the various reports you may generate, the purge procedure must leave enough data to finish the aggregation of reports that have not been aggregated yet. For example, you should leave at least 32 days' worth of data to complete month-level views and 13 months' worth to complete year-level views. The earliest that data is pruned using the default purge schedule is two full years. The purge procedure prunes those transactions and interactions that are both marked done and are older than the prune date stored in the purge schedule. The time required for the purge operation to complete is dependent on many parameters, including number of users, database size, and your purge schedule.

The PURGING_RULES table in the Data Mart stores the purge rules you define for keeping and purging data. Four fields in this table store these values:

- WINDOW_WIDTH
- WINDOW_GRANULE
- OFFSET_LENGTH
- OFFSET_GRANULE

Figure 20 shows a cutaway of the ETL Assistant user interface when a specific report view is selected from the Report Views tab. The captions show how your settings map to the fields of the PURGING_RULES table in the Data Mart.



Purging:

Keep for 2 Year

Purge on 0 Day

Figure 20: How ETL Assistant Maps to the PURGING_RULES Table

Refer to *Reporting ETL Assistant Help* for more information about selecting report views.

Additional rules apply over and above the rules you set:

- Data is available for purge beginning at 00:00 AM on the day indicated by OFFSET_LENGTH+1 and OFFSET_GRANULE. In the figure above, data older than two years will be eligible for purge on January 1 at 00:00 AM. For the OFFSET_GRANULE value of Week, purging starts on Monday at 00:00 AM.
- Purge transactions are committed in blocks of records (up to 1,000 records/block) in which the block size is determined by the report view's TIME_KEY attribute. If the purge routine is interrupted for any reason, it will resume later where it left off (after the last committed transaction block).
- WINDOW_GRANULE must be greater than or equal to OFFSET_GRANULE. If it is greater than, then the data eligible for purge is met by the following criteria:

  ```
  WINDOW_GRANULE - (WINDOW_WIDTH x WINDOW_GRANULE)
  ```

  If WINDOW_GRANULE is equivalent to OFFSET_GRANULE, then eligible data available meets the following criteria:

  ```
  WINDOW_GRANULE - (WINDOW_WIDTH x WINDOW_GRANULE)
  + (OFFSET_LENGTH x OFFSET_GRANULE)
  ```

Refer to the *Reporting Physical Data Model* document for your RDBMS type for more information about Data Mart schema.

Example 1

To keep data for one full month and purge on the third day of the following month, use ETL Assistant to set the following rules:

```
WINDOW_GRANULE = MONTH
WINDOW_WIDTH = 1
OFFSET_GRANULE = DAY
OFFSET_LENGTH = 2
```

That is, on the third day of every month at 00:00 AM, data that is eligible to be purged includes all data prior to, but not including, the first day of the last month. Based on these rules, if today were June 3, 2012, all data stored before May 1 would be eligible for ETL Runtime to purge.

Example 2

To keep data for one full week and purge on the third week following, use ETL Assistant to set the following rules:

```
WINDOW_GRANULE = WEEK
WINDOW_WIDTH = 1
OFFSET_GRANULE = WEEK
OFFSET_LENGTH = 3
```

That is, three full weeks (starting from Monday at 00:00 AM) following a particular week must elapse before that week is purged from the Data Mart. Hence, the maximum amount of data stored under this scenario is just under four weeks' worth. On Sunday, July 31, 2011, at 10:00 PM, for example, you could initiate the purge routine, which, if completed before Monday August 1, 2011, at 00:00 AM, would purge all data prior to July 3, 2011, roughly four weeks prior. You see, at 10:00 PM on July 31, the week of July 25 through 31 is not a complete week (it's two hours short), so it cannot be included in the three-week count. Running and completing the purge routine on Monday, July 25 at 1:00 AM purges the same set of data. Incidentally, you could set `WINDOW_WIDTH` and `OFFSET_LENGTH` to `4` and `0`, or `3` and `1`, or `2` and `2`, or `0` and `4` respectively, to achieve the same results.

Genesys

Chapter

10 Application Files

During installation, an ETL Runtime application folder is created and populated with several files. This chapter lists each file and describes its purpose.

Warning! Do not attempt to run any script manually because of the potential for data loss. The `.sql` files in the following tables are intended for internal use by ETL Runtime. They are provided for highly skilled database administrators.

Table 15 describes the files in the `ETL_Runtime` directory.

Table 15: Contents of the ETL_Runtime Directory

File Name	Description
bcel-license.txt	The Apache Software License.
cfg_lib.jar	Configuration library.
comlib.jar	Java common library.
common.lms	Common file of Genesys log messages.
common_lib.jar	Java common library provided for backward compatibility. (Same contents as comlib.jar.)
conf_report.jar	Configuration-Tracking module. This module is also known as the ETL Tracking module.
createdb.properties	File containing runtime parameters for Data Mart database creation. This file is specified: • On the command line when running ETL Runtime manually. • Within `run-createdb.bat`.

Table 15: Contents of the ETL_Runtime Directory (Continued)

File Name	Description
DataMart.lms	Log messages file for Data Mart–specific log events.
db2_jcc.jar	JDBC drivers for a DB2 IBM system.
db2jcc_license_cu.jar	License file for for Linux, UNIX, and Windows servers.
etl.properties	File containing common properties for ETL Runtime.
gutils.jar	Java library.
jtds_1.2.2.jar	JDBC drivers for a Microsoft SQL Server system.
LICENSE.jTDS	The GNU lesser general public license.
oracle_jdbc_oracle_ classes12.jar	JDBC drivers for an Oracle system.
purge.jar	Purging module.
purge.properties	File containing runtime parameters for purging the Data Mart. This file is specified: • On the command line when running ETL Runtime manually. • Within `run-purge.bat.`
retroweaver-rt.jar	Retroweaver runtime library that supports Java 1.5 language features. Used for purging, if you are running Java 1.4.
run-agg_only.bat	Sample batch file run to aggregate existing data within the Data Mart.
run-createdb.bat	Batch file run to create and initialize the Data Mart.
run-purge.bat	Sample batch file that calls `purge.jar.`
run-tenants_alias_update.bat	Sample batch file to update tenant aliases.
run-trans_and_agg.bat	Batch file run to transfer data from source ODSs and aggregate it within the Data Mart. ETL Runtime continues running and periodically checks source ODSs for new data to pull and aggregate.
run-trans_once.bat	Batch file run to transfer data from source ODSs to the Data Mart. ETL Runtime exits after one round of data transformation.
run-trans_only.bat	Batch file run to transfer data from source ODSs to the Data Mart. ETL Runtime continues running and periodically checks source ODSs for new data.
starter.exe	Application executable for Windows platforms.

Table 15: Contents of the ETL_Runtime Directory (Continued)

File Name	Description
startServer.bat	Batch file that starts ETL Runtime as a regular application, not as a Windows Service.
tat.jar	Tenant Alias–Tracking module.
tenants_alias_update.properties	File containing sample runtime parameters for the Tenant Alias–Tracking module. Modify this file to include your environment's specific information before using it.
tkv.jar	Internal file that is part of the Genesys Framework libraries.
transform.jar	Transformation and Aggregation module.

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Solution Reporting

- *Reporting 7.6 Deployment Guide,* to learn how to configure and install ETL Runtime, and other Reporting components, using the Historical Reporting Wizard.
- Standard Physical Data Model for ODS (Operational Database Storage) for your database type, a Microsoft Word file deployed during Data Sourcer installation in the `database` subdirectory. This document describes ODS table structure and field definitions.
- *Framework Configuration Manager Help* for information about setting configuration options.
- *Reporting 7.6 Data Sourcer User's Guide* for information about the Data Collection Services.
- *Reporting 7.6 ETL Assistant Help,* to learn how to set Data Mart purging rules.
- *Reporting 7.6 Physical Data Model for a Sybase Data Mart, which will help you understand the table structure and field definitions of a Sybase structure.*
- *Reporting 7.6 Physical Data Model for an Oracle Data Mart,* which will help you understand the table structure and field definitions of an Oracle structure.
- *Reporting 7.6 Physical Data Model for a DB2 Data Mart,* which will help you understand the table structure and field definitions of a DB2 structure.
- *Reporting 7.6 Physical Data Model for a Microsoft SQL Data Mart,* which will help you understand the table structure and field definitions of a Microsoft SQL structure.

- *Reporting 7.6 Reference Manual,* to find out about ETL Runtime performance and to learn from some configuration-tuning examples that will help you set up your environment for partial-period reporting.
- *Reporting 7.6 Master Index,* which will help you find where other related topics are documented.

Genesys

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Customer Care website at `http://genesys.com/customer-care`.

Information about supported hardware and third-party software is available on the Genesys Customer Care website in the following documents:

- *Genesys Supported Operating Environment Reference Guide*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which provides information about Genesys hardware sizing guidelines for the Genesys 7.x releases.
- *Genesys Interoperability Guide,* which provides information on the compatibility of Genesys products with various Configuration Layer Environments; Interoperability of Reporting Templates and Solutions; and G*plus* Adapters Interoperability.
- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures relevant to the Genesys licensing system.
- *Genesys Database Sizing Estimator 7.6 Worksheets,* which provides a range of expected database sizes for various Genesys products.

For additional system-wide planning tools and information, see the release-specific listings of System-Level Documents on the Genesys Documentation website (docs.genesys.com).

Genesys product documentation is available on the:

- Genesys Customer Care website at `http://genesys.com/customer-care`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesys.com`.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

76rt_us_etlruntime_08-2012_v7.6.101.00

You will need this number when you are talking with Genesys Customer Care about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 16 describes and illustrates the type conventions that are used in this document.

Table 16: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 100).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 16: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index