Genesys

Reporting 7.6

Data Sourcer

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.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support 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

Genesys and the Genesys logo are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other company names and logos may be trademarks or registered trademarks of their respective holders. © 2012 Genesys Telecommunications Laboratories, Inc. All rights reserved.

The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

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

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 7. For complete contact information and procedures, refer to the *Genesys Technical Support Guide*.

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.genesyslab.com

Document Version: 76rt_us_datasourcer_07-2012_v7.6.102.00

Table of Contents

Preface

Welcome to the *Reporting 7.6 Data Sourcer User's Guide*. This document introduces you to the concepts, terminology, and procedures that are relevant to historical reporting. It is valid only for the 7.6.*x* release(s) of Data Sourcer.

Note: For versions of this document that have created for other releases of this product, please visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on .

About Data Sourcer

Data Sourcer is a component of the Data Collection Services. It services both CC Analyzer and CCPulse+ in the collection of historical data for contact centers that use the Genesys T-Server. This guide describes the role that Data Sourcer plays in your Reporting environment, including:

- Which Configuration Server objects Data Sourcer tracks.
- How Data Sourcer organizes data.
- How to fine-tune Data Sourcer configuration options.
- How to customize Data Sourcer.
- How to perform basic troubleshooting, should you experience network problems.

- What measures Data Sourcer takes to prevent loss of data when your servers are down.

Intended Audience

This guide, which is primarily intended for system administrators, assumes that you have a basic understanding of:

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

You should also be familiar with Genesys Framework architecture and functions.

Chapter Summaries

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

- Chapter 1, "Overview of the Data Collection Services," on page 11, describes Data Sourcer's role in the Data Collection Services.
- Chapter 2, "Fine-Tuning Data Sourcer Operation," on page 19, helps you fine-tune Data Sourcer's configuration using the Genesys Configuration Manager.
- Chapter 3, "Tracking Configuration Server Objects," on page 31, describes how Data Sourcer tracks the tenant and object changes that occur in the Configuration Server.
- Chapter 4, "Common Log Options," on page 35, describes log configuration options that are common to all Genesys server applications and applicable to any Framework server component.
- Chapter 5, "Customizing Data Sourcer," on page 57, describes how to use Data Sourcer to connect to two or more Genesys solutions, how to change the default time profile, and how to set alarm conditions to alert you when your Reporting environment is stressed.
- Chapter 6, "Troubleshooting," on page 61, describes Data Sourcer's safeguards for data recovery in the event of dropped server connections, database unavailability, or network problems. This chapter also instructs you on how to correct common problems that can occur in your reports as a result of missing or unrecognized statistical types.

- Chapter 7, "Emergency-Save Files," on page 65, describes the circumstances under which Data Sourcer writes to an emergency save file instead of to ODS; how to manually move the data from this file to the database; and how to tune your Reporting environment to minimize Data Sourcer's regular use of this file.

- Chapter 8, "Application Files," on page 69, describes the files that are deployed during Data Sourcer 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@genesyslab.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 Technical Support 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 Technical Support

If you have purchased support directly from Genesys, contact Genesys Technical Support at the regional numbers below.

Note: The following contact information was correct at time of publication. For the most up-to-date contact information, see the Contact Information on the Tech Support website. Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.

Genesys Technical Support Contact Information

Region	Telephone	E-Mail
North America and	+888-369-5555 (toll-free)	support@genesyslab.com
Latin America	+55 11 2948 5776	la-support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-127-645-7002	support@genesyslab.co.uk

Genesys Technical Support Contact Information (Continued)

Region	Telephone	E-Mail
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
India	000-800-100-7136 (toll-free) +61-7-3368-6868	support@genesyslab.com.au

Document Change History

This section lists topics that are new in the current release of *this document*, or that have changed significantly from the preceding release.

New in Document Version 7.6.001.00

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

- A previously undocumented configuration option (sql-save-path) is now described on page 28.

- Several configuration options that were introduced in the Data Sourcer 7.2.*x* releases are now described in Chapter 2, including:
  - auto-create-new-tenant-layouts (page 21).
  - conf-check-interval (page 23).
  - max-write-attempts (page 27).
  - max-write-reschedule-attempts (page 27).
  - verbose-request-statistics (page 28).

- Two options were updated to provide the range of valid values, as well as Genesys-recommended values:
  - data-flow-check-interval (page 24)
  - data-flow-timeout (page 25)

  The values of these options apply to both primary and backup (if configured) Stat Server applications.

- Data Sourcer's support of the db-request-timeout configuration option for database access point Application objects is described on page 30.

- Additional common log options, introduced in the 7.6 release, are described in this document in Chapter 4, including those in the following new sections:
  - "Log-Extended Section" on page 51
  - "Common Section" on page 55

- Two alarm conditions were added to Table 11 on page 59 to describe the timeliness of data that Data Sourcer receives from a backup Stat Server.

- A workaround was added to the "Troubleshooting" chapter on page 64 to address a special circumstance on Oracle relational database-management systems (RDBMSs) that involves regional settings that define a decimal-point character different from that defined by the RDBMS.

- The listing of files that are deployed during a Data Sourcer installation was updated in Chapter 8, "Application Files," on page 69.

For other Data Sourcer changes that are introduced in the 7.6 release, such as newly supported operating systems and RDBMSs and installation changes, refer to the *Reporting Deployment Guide.*

Note: This release no longer supports the operations of Data Sourcer in Cartridge mode. Discussions about IS Data Sourcer, including the description of the jdbcurl configuration option, have been removed from this document.

New in Document Version 7.6.101.00

- The new-data-processing configuration option has been removed from the collector section. This option has been deprecated and replaced by the chunk-composition-mode configuration option, which more precisely controls the algorithm used by Data Sourcer to compose chunks from incoming data in special situations, such as, high-volume data environments and single non-High Availability Stat Server environments. The chunk-composition-mode configuration option can be found on page 22.

- The "Dropped Stat Server Connection" section on page 62 has been updated to describe the correct behavior of the primary Data Sourcer after it loses connectivity to both primary and backup Stat Server. Data Sourcer releases ODS lock and assumes backup mode.

New in Document Version 7.6.102.00

- SAP Crystal Reports replaces Hyperion Intelligence Designer and Hyperion Interactive Reporting Studio in this release. See page 14.

- RG Assistant is no longer supported in this release. References to this component have been removed from this document.

1

Overview of the Data Collection Services

This chapter defines the components of the Data Collection Services and specifically describes Data Sourcer's functionality in relation to its sister components.

This chapter also briefly describes the other components of Reporting's Data Collection Services, such as Stat Server. Refer to the *Framework Stat Server User's Guide* and other documents of this documentation set for further information about Stat Server functionality.

This chapter contains the following sections:

What Is a Collection Unit?

The foundation for the historical reporting functionality of both CC Analyzer and CCPulse+ is their Data Collection Services, which consist of one or more *collection units*—a grouping of servers and databases that collect and store data. For CTI-based contact centers. Each collection unit comprises a Genesys Stat Server, Genesys Data Sourcer, DB Server, and an industry-standard database that is referred to as ODS (for Operational Data Storage) in Figure 1.
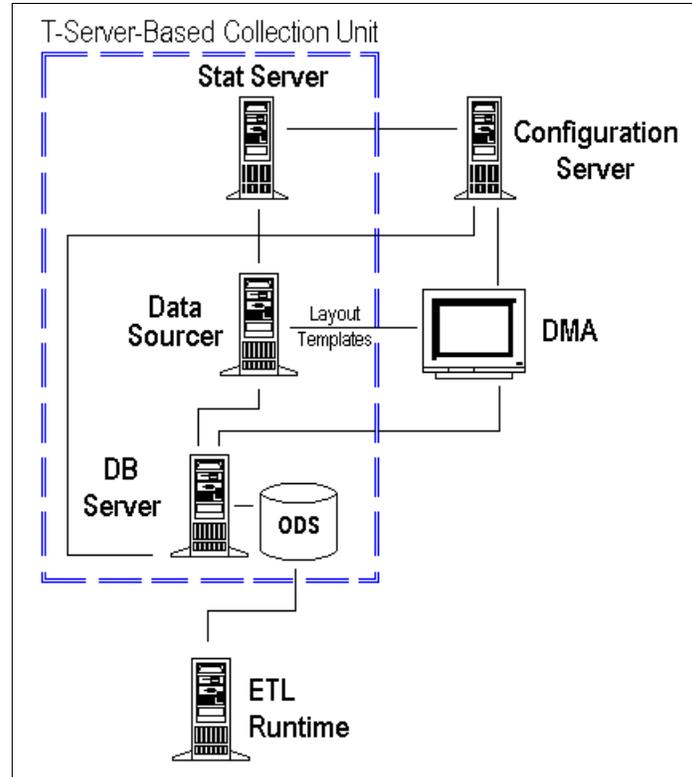
Figure 1: T-Server–Based Collection Unit

To enable implementation of the high availability reporting architecture, the definition of the collection unit includes primary and backup Data Sourcer applications and primary and backup Stat Server pairs. Refer to "Data Sourcer in a High-Availability Environment" on page 14 for further information.

To collect 6.5 (and prior) Internet-based data from Internet-based environments, use Data Sourcer release 7.2 in ICS Cartridge mode. Data Sourcer 7.6 no longer supports this method of operation.

Layout Templates

Data Sourcer uses *layout templates* to store statistics that are to be collected for defined contact-center objects. Each layout template is stored in ODS and specifies:

*   Statistics to be included in each new report layout.

*   Statistical type parameters.

*   Time profiles for the schedules to be created.

*   Default name for report layouts.

*   Default description.

Data Modeling Assistant

You use the Data Modeling Assistant (DMA) to import, export, edit, create, and delete layout templates. Using DMS, you import layout templates that are specific to your solution. Layout templates for Genesys Outbound Contact, for instance, contain a set of statistics that include, among other outbound-specific activities, indicators for campaign performance: the number of callbacks that are completed, the number of unsuccessful dial attempts, how long a campaign was active, and so forth.

Report Layouts

Data Sourcer uses layout templates to activate, at your option, *report layouts* automatically for newly detected tenants within your contact center. Report layouts specify:

*   The objects that are to be monitored.

*   The statistics that are to be collected for these objects.

*   How often statistical data is to be collected.

When report layouts are first created, they are automatically activated for each new tenant if the Data Sourcer `auto-create-new-tenant-layouts` configuration option (described on is set to `true`. Data Sourcer collects statistical data only for activated report layouts. You can activate or deactivate report layouts at any time by using DMA. The number of report layouts Data Sourcer can create is bound by database limitations.

Refer to "Predeployment Measures" in the *Reporting Deployment Guide* and "Performance Measurements" in the *Reporting Reference Manual* for further information.

Time Profile

Data Sourcer collects statistical information about contact center objects on a periodic basis, which is defined by the *time profile* that is associated with the report layout. This time profile defines the timing and frequency in consecutive time intervals of constant duration when Data Sourcer is to send requests for statistics to Stat Server. For instance, in the default time profile, which is named `CollectorDefault`, Data Sourcer sends requests to Stat Server every 15 minutes starting at midnight (`0:00+00:15`).

ODS

When Stat Server sends the requested statistics to Data Sourcer, Data Sourcer, in turn, sends that information to a database that is named the Operational Data Storage (ODS). Data Sourcer can access ODS through the Genesys DB Server from any of the following relational database types:

*   Oracle   •   Sybase
*   Microsoft SQL Server   •   DB2

Data Mart Services

The components of a collection unit work together to collect and provide temporary storage for historical data until ETL Runtime (ETL, for Extraction, Transformation, and Loading) can transform and transfer the data to the Data Mart, where data is permanently housed for reporting purposes. Once the data is transferred, ETL Runtime, at your option, deletes the information from ODS. Following data transformation and aggregation, you can use SAP Crystal Reports, CCPulse+, or other various third-party tools to access and report on the data.

ETL Runtime and the Data Mart are both components of the Data Mart Services. Crystal Reports is a component of the Information Delivery Services.

Data Sourcer in a High-Availability Environment

To ensure high availability of historical reporting data, a high availability architecture was implemented beginning with Reporting release 7.2. This architecture employs two Data Sourcer applications that operate in Hot Standby mode, each of which is connected dynamically to two Stat Server applications, two DB Server applications, a high-availability (HA) T-Server, one ODS, and one Data Mart. See Figure 2 for an illustration of this architecture.

In an HA environment, the primary and backup Data Sourcer applications, which are operating in Hot Standby mode, collect the identical set of statistics from the Stat Server pair. The primary Data Sourcer writes this information to ODS. If the backup Data Sourcer determines that the information has been written successfully to ODS, it discards the data that is received from Stat Server; otherwise, the backup Data Sourcer retains these data chunks in memory and continues to check for the presence of this data in ODS.

The backup Data Sourcer also continually checks whether the primary Data Sourcer is connected to ODS—ready to assume the role of primary, if needed.

The operating mode of the Data Sourcer pair in this HA architecture can be different from the mode that is specified in Data Sourcer's configuration. Upon startup, both the primary and backup Data Sourcer applications compete for control of ODS. Regardless of the initial configuration, the Data Sourcer that acquires control of updates to ODS takes on the role of the primary Data Sourcer. The other Data Sourcer then becomes the backup.

Note: Throughout this document, the terms "primary Data Sourcer" and "backup Data Sourcer" refer to the operating mode of Data Sourcer, and not to how Data Sourcer is configured.
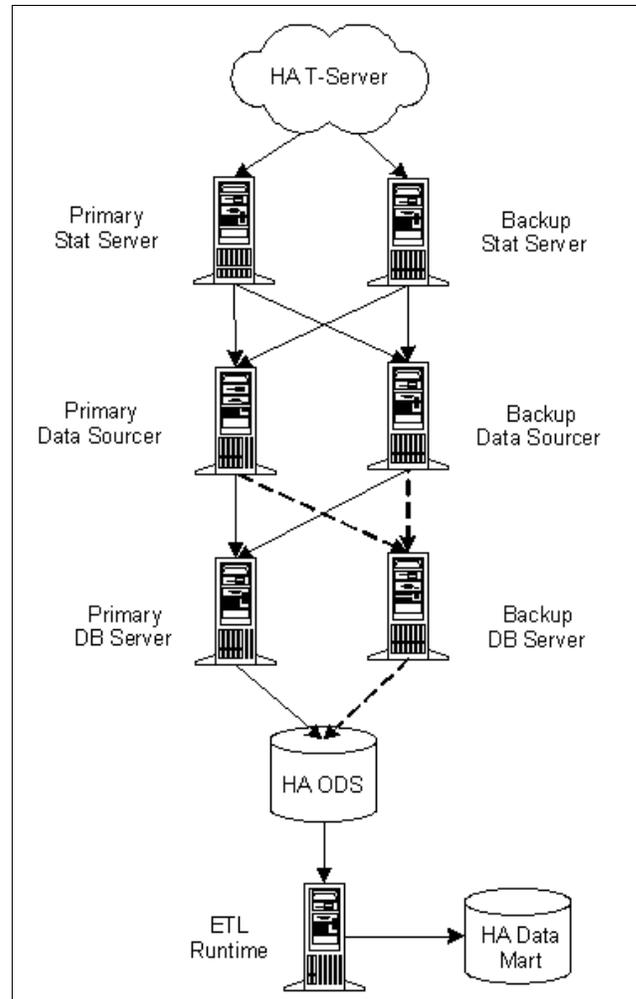
Figure 2: High-Availability Reporting Architecture

In this HA configuration, both primary and backup Stat Server applications must be configured to accept client connections when they are running in backup mode. This functionality enables both primary and backup Stat Server applications to collect the same set of TEvents—ensuring no loss of data, in the event that the primary Stat Server application fails.

Important   To enable this connectivity in backup mode, you must set the `accept-clients-in-backup-mode` Stat Server configuration option in both the primary and backup applications. If this option is not set, you will not achieve a high-availability environment. Refer to the *Framework Stat Server Deployment Guide* set for more information about this option.

A Closer Look at Data Sourcer Connections

A server itself, Data Sourcer is a client of:

• Configuration Server.

• Stat Server.

• DB Server.

Data Sourcer is a server for DMA.

Configuration Server

As a client of Configuration Server, the primary and backup Data Sourcer receive information about contact-center objects for which statistics are collected. Configuration Server provides identity information about these objects (for example, agents, tenants, places, calling lists, and campaigns), as well as information about changes to them (including whether they have been removed from the environment). This information is required for creating and maintaining the appropriate report layouts. Chapter 3 discusses in detail the information that Data Sourcer collects about your contact center.

Beginning with release 7.2.*x,* upon notification, Data Sourcer immediately processes all changes to reportable objects that are received from Configuration Server. If Data Sourcer loses connectivity to Configuration Server, it requests all of the data again upon reconnection. This ensures that Data Sourcer processes any changes that occurred during the disconnect.

Stat Server

Data Sourcer connects to Stat Server in *real-time mode*. This is not to be confused with *real-time statistics,* which, like historical statistics, Stat Server also maintains. You indirectly determine the mode when you specify Data Sourcer application connections during configuration. Specifying that Data Sourcer is to connect to a Stat Server sets the mode at real-time.

Beginning with release 7.2.*x,* a primary and backup Data Sourcer pair connects to a primary and backup Stat Server pair. Before this release, a backup Stat Server could not accept requests from any client, and it did not accommodate the backup mode of Data Sourcer. Beginning with release 7.2, you can set the `accept-clients-in-backup-mode` Stat Server configuration option to enable Data Sourcer to connect to the backup Stat Server application. This enables both Stat Server applications to collect the same set of statistics—thus ensuring minimal loss of data in the event that the primary Stat Server loses connectivity with Data Sourcer.

Note: Under normal conditions, the primary Data Sourcer will use only data that is received from the primary Stat Server.

As a client of Stat Server, Data Sourcer requests only historical statistics for objects that belong to particular report layout and periodically receives statistical values, calculated metrics, and information about whether the reported values are valid. The statistics measure counts and durations over a past time interval, instead of single moments of time. For example, Data Sourcer may issue the following request:

How many calls were answered in the past 15 minutes?

but not:

Which call was answered at 10:02?—Data about a specific interaction

and not:

How many calls are currently in queue?—Nonhistorical data

Data Sourcer does not track individual calls, but instead collects aggregated predefined statistics from Stat Server. You must create and activate report layouts before data collection or the reporting interval begins.

The statistics that Data Sourcer requests for unions of intervals must be calculable from the values for the individual intervals in the union. Such statistics are referred to in SQL as *cumulative* and the categories to which they belong are called *cumulative categories.* For example, `Total Time` and `Total Number` are two cumulative statistical categories; `AverageTime` is not.

Data Sourcer uses only those statistical types that are defined in Stat Server. As additional statistical types are configured within Stat Server, Data Sourcer inherits the ability to use them. Refer to the *Framework Stat Server User's Guide* and the *Reporting Deployment Guide* for more information.

DB Server

As a client of DB Server, Data Sourcer reads information about activated report layouts, and writes to ODS the statistics that it receives from Stat Server.

In a high availability reporting architecture, the primary and backup Data Sourcer is able to connect to a primary and backup DB Server pair.

DMA

Data Sourcer acts as a server for the DMA and receives commands from DMA on the activation and deactivation of report layouts that specify the statistical types (stat types) for which Data Sourcer must collect data. When you activate a report layout within DMA, DMA notifies Data Sourcer, which starts collecting data for that report layout. When you deactivate a report layout, DMA notifies Data Sourcer to stop collecting data for that report layout.

![Genesys]

2 Fine-Tuning Data Sourcer Operation

This chapter describes the options that you can use and change to fine-tune Data Sourcer operation. It covers the following topics:

Refer to and Chapter 4, "Common Log Options," on page 35, for descriptions of log configuration options that are common to most Genesys server applications.

Configuring the Data Sourcer Application Object

You can modify configuration options for your Data Sourcer application within the Configuration Manager after initial Data Sourcer configuration. To add, modify, or delete Data Sourcer configuration options:

1. Within the Applications folder in Configuration Manager, double-click the corresponding Data Sourcer Application object to open its Properties dialog box.

2. Select the Options tab, as illustrated in Figure 3 on page 20.

3. Open the collector and/or log sections, and modify options as necessary.

You do not need to stop Data Sourcer in order to change most option values; however, changes to some options become effective only after you restart Data Sourcer.
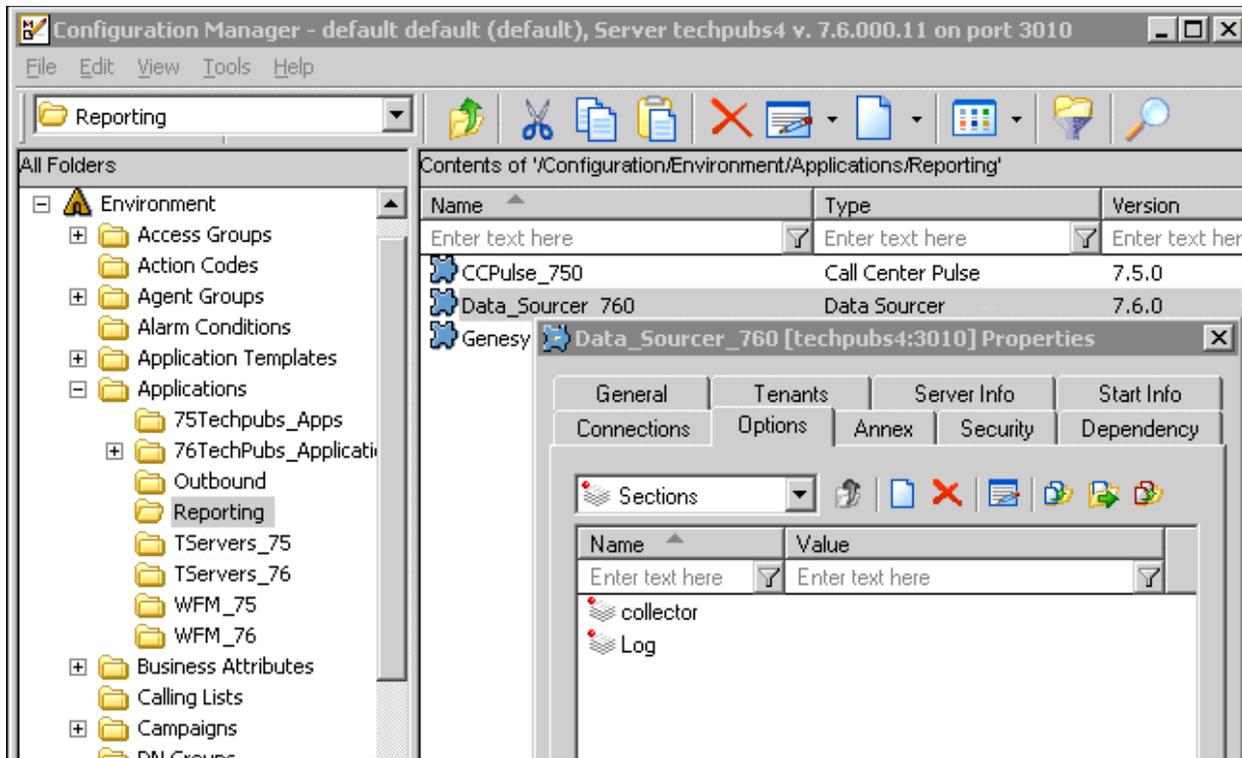
Figure 3: Editing the Data Sourcer Application Object Within Configuration Manager

Collector Section

The collector section holds configuration options with which you can specify how Data Sourcer is to collect data (see Table 1). For those configuration options for which you specify true/false values, any of the following additional values are also valid:

• yes/no

• y/n

• 1/0

• on/off

These alternate values are not indicated in Table 1

Table 1: Configuration Options for Collector Section

Option	Description
activate-new-layouts	Specifies if new report layouts (created by default for new tenants) are to be automatically activated. This option prevents automatic activation where a large number of objects can overload Data Sourcer and cause Data Sourcer to malfunction.
	For a large contact center (for example, more than 1,000 agents), set this option to `false` before Data Sourcer starts *for the first time* because predefined report layouts that are activated for all agents can overload Data Sourcer. If Data Sourcer has already been started, set this option to `false` and then manually deactivate report layouts using DMA.
	Refer to the *Reporting 7.6 Data Modeling Assistant Help* for information about deactivating report layouts.
	Default Value: `false`
	Valid Values: `true, false`
	Change Takes Effect: Immediately
auto-create-new-tenant-layouts	Controls automatic creation of report layouts from existing layout templates when a new tenant is connected to the Data Sourcer.
	Default Value: `true`
	Valid Values: `true, false`
	Change Takes Effect: Immediately

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
chunk-composition-mode	This option specifies the method of receiving and writing data from the primary and backup Stat Server pair. Default Value: legacy Valid Values: legacy, fast, normal, safe legacy—When this value is configured, Data Sourcer overwrites a chunk from the backup Stat Server with the chunk from the primary Stat Server, if it is available. In this mode, this release of Data Sourcer performs the same as Data Sourcer 7.2.100.49 and does not use the Receive queue. Receive Queue—A new queue that is introduced for chunks that are received from both the primary and backup Stat Servers. The Receive Queue increases the reliability of the statistical data captured in high availability (HA) mode. The memory consumption of Data Sourcer increases because this queue requires additional memory. To activate the Receive Queue, set the chunk-composition-mode option to one of the following valid values: fast—When this value is configured, Data Sourcer uses the Receive Queue. In this mode, Data Sourcer writes the chunk, regardless of whether it originated from the primary or backup Stat Server. However, if the received chunk is incomplete, Data Sourcer waits for a subsequent chunk to complete and then decides which copy of the incomplete chunk to submit—the primary or backup. Genesys recommends *only* configuring the fast value in your test environment or in environments that do not include a backup Stat Server. normal—When this value is configured, Data Sourcer uses the Receive Queue. In this mode, Data Sourcer overwrites a chunk from the backup Stat Server with the chunk from the primary Stat Server. In this mode, if a chunk from the backup Stat Server is received, Data Sourcer waits to receive a chunk from the primary Stat Server (unless the chunk from the primary Stat Server was complete—then it does not wait). If a chunk from the primary Stat Server is received but is incomplete, Data Sourcer waits for the chunk from the backup Stat Server and then decides which chunk to submit next, primary or backup. Genesys recommends using the normal value in all configurations. safe—When this value is configured, Data Sourcer uses the Receive Queue. In this mode, Data Sourcer waits to receive chunks from both the primary and backup Stat Servers and once received, decides which chunk to submit. Genesys recommends setting the safe value only in environments with a high-availability (HA) configuration of Stat Server. Setting this value in a non-HA Stat Server configuration may cause constant data writing delays for the one time profile period. Changes Take Effect: After a restart

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
chunk-file-format	Specifies one of two supported file formats for the chunk file. If a value for this option is not specified, or the value is not valid, the default value will be used. Default Value: text Valid Values: binary, text Change Takes Effect: Upon restart
conf-check-interval	Specifies the interval, in minutes, that Data Sourcer checks Configuration Server for updates to those configuration objects that Data Sourcer monitors. Default Value: 60 (minutes) Valid Values: 1–1440 Change Takes Effect: After the nearest configuration-recheck cycle
create-sample-templates	Specifies if Data Sourcer is to create sample templates within ODS. Sample templates are taken from layout_templates.sql, which is a file that the Data Sourcer Configuration Wizard generates based on your solution. Default Value: true Valid Values: true, false Change Takes Effect: Upon restart
data-excess-high-threshold	Specifies the maximum amount of data, expressed as a percentage of memory queue size, that Data Sourcer is to hold in memory before alarming conditions are reached. The default memory queue size is two times the number of active schedules. This option is effective only if Data Sourcer contains active report layouts and has a live connection to Stat Server. Default Value: 100 Valid Values: 0-100000. This value must be higher than the value that is specified by the data-excess-low-threshold configuration option. Change Takes Effect: Immediately **Note:** Although the Data Sourcer Configuration Wizard does not add this option, the default behavior sets its value at 100 percent of memory queue size. **Warning!** Data Sourcer will not evaluate queue size if the value that is specified by this option—the high boundary—is less than or equal to the low boundary (defined by the data-excess-low-threshold configuration option). Make sure to specify meaningful values for both options to ensure that queue-size evaluation can occur.

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
data-excess-low-threshold	Sets the threshold for aborting the condition that is detected by the data-excess- high threshold. Expressed as percentage of memory queue size, this option specifies to what level the amount of data shall drop in order for normal conditions to be considered restored. This option is effective only if Data Sourcer contains active report layouts and has a live connection to Stat Server. Default Value: 65 Valid Values: 0-100000. This value must be lower than the value that is specified by the data-excess-high-threshold configuration option. Change Takes Effect: Immediately **Note:** Although the Data Sourcer Configuration Wizard does not add this option, the default behavior sets this option at 65 percent of memory queue size. **Warning!** Data Sourcer will not evaluate queue size if the value that is specified by this option—the low boundary—is greater than or equal to the high boundary (defined by the data-excess-high-threshold configuration option). Make sure to specify meaningful values for both options to ensure that queue size evaluation can occur.
data-flow-check-interval	Specifies how often, in minutes, Data Sourcer is to check for data flow from Stat Server, provided that Data Sourcer contains active report layouts and has a live connection to Stat Server. This setting applies to the primary Stat Server application as well as to its backup, if one has been configured. Default Value: 2 Valid Values: 1–60 Recommended Values: Positive integers that are less than the value that you set for the time profile. The default CollectorDefault time profile is 15. Change Takes Effect: Immediately **Note:** Although the Data Sourcer Configuration Wizard does not add this option to the collector section, the default behavior is that Data Sourcer checks for data flow every two minutes.

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
data-flow-timeout	Specifies how much time can elapse, in minutes, before it is no longer acceptable to suspend the transfer of data from Stat Server (either primary or backup). Data Sourcer must contain active report layouts and have a live connection to Stat Server for this option to be in effect. The timer starts upon the last successful data transfer, which is 0, 15, 30, and 45 minutes after each hour using the default CollectorDefault time profile.
	Default Value: 20
	Valid Values: 1–1440
	Recommended Values: Positive integers that are greater than the value that you specify for data-flow-check-interval and less than two times the value of the time profile that is used. If you use the default time profile, two times its value is 30 (15x2).
	Change Takes Effect: Immediately
	Note: Although the Data Sourcer Configuration Wizard does not add this option to the collector section, the default behavior is that Data Sourcer observes a 20-minute timeout for lack of data flow.
data-table-size	Controls the size, in bytes, of OL_DATA_n tables that store Stat Server statistics. Data Sourcer receives data from Stat Server and writes the data into this table until the number of records exceeds the value that you specify. Then, Data Sourcer creates a new OL_DATA_{n+1} table.
	For more information about OL_DATA_n tables or other ODS tables, refer to the Standard Physical Data Model for your RDBMS (for the location, see Chapter 8 beginning on page 69).
	Default Value: 1,000,000
	Valid Values: Positive integers that are greater than or equal to 1,000
	Change Takes Effect: Immediately

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
emergency-save-directory	Specifies the path to the directory in which Data Sourcer writes historical data if the connection to ODS is lost. This directory must reside on the computer that is running Data Sourcer. **Note:** If Data Sourcer is running as a Windows Service, the Service must have write permissions for this directory. Valid Value: Valid path name or a null value. If a valid path name is not specified, the emergency save files will be saved to Data Sourcer's current working directory. To avoid unexpected results, Genesys recommends that you specify a value for this option. Change Takes Effect: Immediately. **Note:** Genesys recommends that you do *not* change this option while Data Sourcer is running. To change the value of this option safely: 1. Stop Data Sourcer. 2. Create a folder (or verify that one exists) and move any existing emergency save files to this new folder. 3. Set the new option value. 4. Restart Data Sourcer. **Note:** Data Sourcer 7.6.*x* supports both the `emergency-save-directory` and `sql-save-path` configuration options.
management-port	Specifies the TCP/IP port that Data Sourcer reserves for connections that its SNMP Option Management Client establishes. If this option is absent or null, a server for Management Client is not created. Default Value: No default value Valid Values: Positive integers (no other application should use this port) Change Takes Effect: Upon restart **Warning!** You must specify a value for this option if you are using an SNMP connection. Do not change the value for this option while Data Sourcer is running.
max-chunk-size	Specifies the maximum chunk size that Data Sourcer will process. If the maximum chunk size is exceeded, Data Sourcer will not process the chunk, but will log an error message instead. If you do not specify a value for this option, or the value is not valid, the default value will be used. **Note:** The maximum chunk size refers to the maximum number of statistics in a chunk. Default Value: 2000000 Valid Values: 1000–10000000 Change Takes Effect: Upon restart

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
max-write-attempts	Controls how many times Data Sourcer will attempt to write data chunks to ODS. If Data Sourcer is unsuccessful in these attempts, it moves the chunks to the end of the chunk queue. Default Value: 10000 Valid Values: 1–10000 Change Takes Effect: Upon restart **Note:** This option applies only to the Data Sourcer application that is running in primary mode. The attempt-to-write counters are maintained for each Data Sourcer instance. In a high-availability environment, if switchover occurs and the backup Data Sourcer becomes the primary, the counter begins counting attempts anew from 1 or—if this Data Sourcer instance ever served as the primary—from the last value of the counter for this chunk. **Note:** To achieve the behavior that is available in prior releases, set this option to its maximum value.
max-write-reschedule-attempts	Controls how many times Data Sourcer will reschedule writing previously unwritten data chunks to ODS. If the chunks are successfully written to the ODS within the limits that are specified by this option, Data Sourcer will remove the chunks from the chunk queue and write them to the bad_chunks folder that is located in the emergency-save folder. Default Value: 1000 Valid Values: 1–1000 Change Takes Effect: Upon restart **Note:** To achieve the behavior that is available in prior releases, set this option to its maximum value.
new-object-delay	This option specifies the delay between the moment Data Sourcer receives notification of a new object in Configuration Server and the moment it actually starts to process this notification. Setting this option to 0 enables Data Sourcer to process new objects without delays, as it did in previous releases. Default Value: 0 Valid Values: 0 - 1440 Changes Take Effect: Immediately

Table 1: Configuration Options for Collector Section (Continued)

Option	Description
person-presentation-format	Specifies the order of agent information that Data Sourcer is to observe when it writes agent information to ODS. Default Value: %l %f (last name followed by first name) Valid Values: Any combination of the following: • %f (first name) • %l (last name) • %u (username) • %b (birth date) • %c (comment) • %i (employee id) Change Takes Effect: Immediately To include the % symbol within formatted data that is written to ODS, repeat the symbol: %%. **Note:** Although the Data Sourcer Configuration Wizard does not add this option to the collector section, the default behavior is that Data Sourcer observes %l %f format for agent information.
receive-queue-size	This option determines the size of the chunk queue that is maintained by Data Sourcer for each schedule, and determines the number of periods for which Data Sourcer is able to keep chunks before it has to submit some of them. This option is useful in high-volume data environments in which Stat Server is supplying data at different speeds. Default Value: 3 Valid Values: 1 - 100 Changes Take Effect: After a restart
sql-save-path	Is the same as emergency-save-directory. Refer to the description of this option on page 26.
verbose-request-statistics	Controls whether Data Sourcer logs verbose messages about the objects for which it requests statistics. Default Value: false Valid Values: true, false Change Takes Effect: Immediately

Configuring the ODS Database Access Point

Data Sourcer 7.2.003.10 (and higher) retrieves certain configuration parameters of the database access point (DAP) `Application` that is used to access ODS. To enable Data Sourcer to retrieve these parameters, modify the ODS DAP `Application` and add a new section and option as follows:

1. In Configuration Manager, open the `Application Properties` window of the ODS DAP.

2. On the `Options` tab, create a new section that is named `ODS-USER-SCHEMA`.

3. Within this new section, create and configure the `<option name>` configuration option (see Table 2).

ODS-USER-SCHEMA Section

The `ODS-USER-SCHEMA` section holds one user-defined configuration option which is described in Table 2.

Table 2: Configuration Option for ODS-USER-SCHEMA Section

Option	Description
<option name>	Specifies the name of the user that is used by the DAP to log in to ODS. The name of this option must match the user name exactly. Create this option if the object schema name (on SQL Server 2005) or object owner (on SQL Server 2000) differs from the user (`User Name`) that is configured in the ODS DAP `Application`. **Note:** Do not create an option named `sa` with a value of `dbo`. (Data Sourcer automatically uses the schema name `dbo` for the user `sa,` so that additional configuration is not required in this case.) Default Value: No default value Valid Values: The option value must be the same as the schema name (on SQL Server 2005) or owner name (on SQL Server 2000) that holds ODS objects in ODS. Change Takes Effect: Upon restart

dbserver Section

Beginning with the 7.6 release, Data Sourcer can read the value of the `db-request-timeout` configuration option of the DAP configured to access ODS. It uses the value as the timeout for database queries that Data Sourcer makes to ODS. This option was introduced in the Framework 8.0 release and is defined

in the [dbserver] section on the Annex tab of an 8.0 (or later) DAP Application object.

Table 3: Configuration Option for dbserver Section

Option	Description
db-request-timeout	Specifies the maximum time, in seconds, that Data Sourcer will wait for one DBMS request to be completed. If a request takes longer than the specified timeout, Data Sourcer generates an error and treats the current operation with database as failed. Default Value: 600 (10 minutes) Valid Values: 0–604800 (7 days) Change Takes Effect: Immediately **Note:** If a value or zero is specified, Data Sourcer enforces a 10-minute timeout that differs from what is documented in the *Framework 8.0 Configuration Options Reference Manual*. Data Sourcer also uses this value if the DAP holds an invalid value.

![Genesys]

3

Tracking Configuration Server Objects

Since the release of Reporting 7.2, both the primary and backup Data Sourcer applications maintain a constant connection to Configuration Server. Any changes to reportable objects are detected and processed immediately. The primary Data Sourcer then updates ODS accordingly.

Only those Configuration Server objects that have been included in *activated* report layouts are monitored. Data Sourcer monitors changes to those tenants that you have assigned to Data Sourcer.

Note: Some objects might not be visible to Data Sourcer because of security settings in Configuration Server.

This chapter describes Data Sourcer functionality and contains the following sections:

Note: Throughout this chapter, for simplicity, the term Data Sourcer is used to denote both the primary and backup Data Sourcer applications when both operate identically. When there is a difference in operation, primary or backup will be explicitly stated.

Tracking Tenant Changes

Data Sourcer tracks the following tenant changes within Configuration Server:

- New tenants
- Deleted tenants

> •   Tenant name changes

Data Sourcer does not track any other changes to a tenant's configuration.

Note: To collect data for any report layout with the `Tenant` object type, you must include the `Environment` tenant in your configuration of Data Sourcer and Stat Server. Only under the `Environment` tenant is it possible to have such report layouts.

Detecting New Tenants

When a new tenant is assigned to Data Sourcer, Data Sourcer adds this tenant to ODS and records the time. The recorded time reflects the time at which the tenant was added to ODS, not to Configuration Server. Data Sourcer then creates report layouts from all active layout templates that are available for this tenant. If you have set the `auto-create-new-tenant-layouts` configuration option to `true`, Data Sourcer automatically creates report layouts from existing layout templates. If `activate-new-layouts` is set to `true`, Data Sourcer automatically activates the report layouts and begins to collect data

Detecting Deleted Tenants

When a tenant is deleted from Configuration Server, Data Sourcer performs the following operations:

- Marks the tenant as deleted within ODS and records the time. Again, the recorded time is the time at which the tenant was marked for deletion within ODS, not the time at which the tenant was removed from Configuration Server.

- Stops collecting information for this tenant.

- Marks all report layouts for this tenant as deleted.

Detecting Changed Tenants

When a tenant name changes within Configuration Server, Data Sourcer substitutes the new name for the old one within ODS.

Tracking Object Changes

Data Sourcer tracks the following object changes within Configuration Server:
- New objects
- Object-name changes
- Deleted objects

Data Sourcer tracks no other changes to an object's configuration.

Deleting New Objects

When Data Sourcer detects new objects within a particular group in Configuration Server, the primary Data Sourcer adds these objects to ODS and records the time. The recorded time is the time at which the object was updated within ODS, not Configuration Server. Data Sourcer also notes when an object is removed from a particular group (without being deleted from Configuration Server) and then restored to the same group (to maintain a change history for groups).

Likewise, Data Sourcer updates ODS if an object is added to a report layout within DMA.

In the HA reporting architecture, both the primary and backup Data Sourcer applications try to add new configuration objects to ODS; only one, however, can be successful. In this way, if the primary Data Sourcer fails, the backup Data Sourcer can add any new configuration objects to ODS—thus, ensuring minimal loss of data. Both the primary and backup Data Sourcer applications perform a periodic check of ODS, when new objects are detected, they start collecting data on them.

Deleting Changed Objects

When an object name changes within Configuration Server, either the primary or backup Data Sourcer updates the corresponding fields in ODS, replacing the old name with the new. Data Sourcer also notes the update time in the database. Again, the recorded time is the time at which the object was updated within ODS, not within Configuration Server.

Detecting Deleted Objects

When an object is deleted from Configuration Server, the primary Data Sourcer marks this object as deleted within ODS and records the time. Note that the primary Data Sourcer does not remove the corresponding records from ODS, but instead *marks* them as deleted. The recorded time is the time at which the object was marked for deletion, not the time at which the object was removed from Configuration Server.

Likewise, either the primary or backup Data Sourcer updates ODS if an object is removed from a report layout within DMA.

![Genesys logo]

4

Common Log Options

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

Mandatory Options

You are not required to configure any common log options to start Data Sourcer.

Log Section

You must call this section `log`. Table 4 lists the log configuration options that are available to you. To use these options, note that you must actively set them, either by using the Data Sourcer Wizard or manually within Configuration Manager. The default Data Sourcer application template includes the `verbose,` `all,` and `buffering` options.

Table 4: Log Options

Option	Description
verbose	Determines whether a log output is created. If it is, this option specifies the minimum level of log events that are generated. The log-event levels, starting with the highest-priority level, are Standard, Interaction, Trace, and Debug. Refer to "Log Output Options" on page 45 for more information. Default Value: all 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 statserver 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 *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 45). Setting this option to true increases output performance. **Note:** When you enable buffering, log messages might appear in the log after a delay. Default Value: false 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 which the segment stays 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 Data Sourcer 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). Data Sourcer 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—that is, 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, Data Sourcer looks for the file in its current working directory. Default Value: `statserver.lms` Valid Value: `<string>.lms` Changes Take Effect: After Data Sourcer restarts if Data Sourcer finds `statserv.lms` at startup or immediately if Data Sourcer cannot find this file at startup **Warning!** If Data Sourcer cannot find its message file upon startup, it will not be able to generate Data Sourcer–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: Regardless of whether the `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 Data Sourcer. Default Value: `false` Valid Values: `true, false` Changes Take Effect: Immediately
check-point	Specifies how often, in hours, Data Sourcer generates a `check-point` log event to divide the log into sections that are equal in time. By default, Data Sourcer 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 Data Sourcer regularly prints a snapshot of the memory output (see page 45). The new snapshot overwrites previously written data. If Data Sourcer 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, Data Sourcer 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 45 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 Data Sourcer creates temporary log-related files. If you change this value while Data Sourcer 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 Data Sourcer uses 6.*x* output logic.
	
	Default Value: `false`
	Valid Values:
	
	`true` — The log of the level that is specified by one of the log-output options that are described on is sent to the specified output.
	
	`false` — The log of the level that is specified by one of the log-output options that are described on 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 that you change 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 6.*x* releases. Setting this option to `true` affects log consistency.

The configuration options that are listed in Table 5 enable you to generate debug logs that contain information about specific Data Sourcer operations. You designate these options in the `log` section of the Data Sourcer application.

Warning! Genesys advises that you use these options only when you are requested to do so by Genesys Technical Support.

Table 5: Debug-Log Options

Option	Description
x-conn-debug-open	Generates debug log-records about "open connection" Data Sourcer operations. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-select	Generates debug log-records about "socket select" Data Sourcer operations. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-timers	Generates debug-log records about Data Sourcer operations that are involved in creating and deleting system timers. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-write	Generates debug-log records about Data Sourcer write operations. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart

Table 5: Debug-Log Options (Continued)

Option	Description
x-conn-debug-security	Generates debug-log records about security-related operations, such as Transport Layer Security (TLS) and security certificates. **Note:** This option has no effect on Data Sourcer, which does not support TLS operations. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-api	Generates debug-log records about connection library-function calls. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-dns	Generates debug-log records about DNS operations. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart
x-conn-debug-all	Generates debug-log records about open-connection, socket-select, timer creation-and-deletion, write, security-related, DNS-operation, and connection library-function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<optype>` options. Default Value: 0 Valid Values: 0 Log records are not generated. 1 Log records are generated. Changes Take Effect: After restart

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). See Table 6.

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 to log 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 49.

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

Warnings! 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`) if 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 6: Log-Output Options

Option	Description
all	Specifies the outputs to which Data Sourcer 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 Data Sourcer 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 6: 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 6: 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 Data Sourcer 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 (see Table 7) to identify log files that Data Sourcer 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 = statservlog`, Data Sourcer prints log messages into a text file called `statservlog.<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`, Data Sourcer prints log messages into a file called `statserv.<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 Data Sourcer generates before abnormal termination. For example, if you set `standard = statservlog`, Data Sourcer prints the last log message into a file called `statserv.<time_stamp>.snapshot.log` in the event of failure.

Note: Provide `*.snapshot.log` files to Genesys Technical Support 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 = statserv`, Data Sourcer prints the latest memory output to a file called `statserv.<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, statservlogfile
```

With this configuration, Data Sourcer generates only log events of the `Standard` level and sends them to the standard output, to Message Server, and to a file named `statservlogfile`, which Data Sourcer 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/statservlogfile
trace = network
```

With this configuration, Data Sourcer generates log events of the Standard, Interaction, Trace, and Debug levels, and sends them to the standard output and to a file named statservlogfile, which Data Sourcer creates in the /usr/local/genesys/ directory. In addition, Data Sourcer 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 statserver section.

Failure-Troubleshooting Log Section

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

With this configuration, Data Sourcer 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 statservlogfile, which the application creates in its working directory. An increased memory storage enables Data Sourcer 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 Data Sourcer's log at the moment of failure. This should help you and Genesys Technical Support identify the reason for the failure. Be sure to appropriately set the debug-level option in the statserver section.

Note: If you are operating Data Sourcer on UNIX and do not specify any files in which to store the memory output snapshot, the core file that Data Sourcer produces before terminating contains the most current Data Sourcer log. Provide the Data Sourcer's core file to Genesys Technical Support when reporting problems.

Log-Extended Section

This section must be named log-extended.

Table 7: Extended Log Options

Option	Description
level-reassign-<eventID>	Specifies one of five log levels for the *<eventID>* log event, which may differ from its default level, or disables logging of the named event altogether. This option is useful if you want to change the behavior of what Data Sourcer logs for the specified log event ID. If no value is specified, the named log event retains its default level. You can deactivate these options by using the level-reassign-disable configuration option, as described below. Default Value: Default value of the *<eventID>* log event. Refer to the *Common Log Events Help* or datasourcer.lms (located in the directory in which Data Sourcer is installed) for a listing of each of Data Sourcer's default levels. Valid Values: alarm — The log level of the *<eventID>* log event is set to alarm. standard — The log level of the *<eventID>* log event is set to standard. interaction — The log level of the *<eventID>* log event is set to interaction. trace — The log level of the *<eventID>* log event is set to trace. debug — The log level of the *<eventID>* log event is set to debug. none — The *<eventID>* log event is not recorded in a log. Changes Take Effect: Immediately
level-reassign-disable	When this option is set to true, the original (default) log level of all log events in the [log-extended] section are restored. This option is useful when you want to use the default levels and keep the customizations. Default Value: false Valid Values: true, false Changes Take Effect: Immediately Defined In: Options tab of Application object

Warning!   Use caution when you make these changes in a production environment.

Depending on the log configuration, changing the log level to a higher priority might cause the log event to be logged more often or to a greater number of outputs. This could affect system performance.

Likewise, changing the log level to a lower priority might cause the log event to be not logged at all, or not logged to specific outputs—thereby, losing important information. The same applies to any alarms that are associated with that log event.

In addition to the preceding precautionary message, please take note of the following:

• Logs can be customized only by applications of release 7.6 or later.

• When the log level of a log event is changed to any level except none, it is subject to the other settings in the [log] section at its new level. If it is set to none, it is not logged and therefore not subject to any log configuration.

• Changing the log level of a log by using this feature changes only its priority; it does not change how that log is treated by the system. For example, increasing the priority of a log to Alarm level does not mean that an alarm will be associated with it.

• Each application in an HA pair can define its own unique set of log customizations; however, but the two sets are not synchronized with each other. This can result in different log behavior, depending on which application is currently in primary mode.

• This feature is not the same as a similar feature in Universal Routing Server, version 7.2 or later. In this Framework feature, the priority of log events are customized. In the URS feature, the priority of debug messages only are customized. Refer to the *Universal Routing Server 8.0 Reference Manual* for more information about the URS feature.

• You cannot customize any log event that is not in the unified log record format. Log events of the Alarm, Standard, Interaction, and Trace levels feature the same unified log-record format.

Example

This is an example of the use of customized log-level settings, subject to the following log configuration:

```
[log]
verbose=all
all=stderr
interaction=log_file
standard=network
```

Before the log levels of the log are changed:

- Log event 20009—with default level `trace`—is output to `stderr`.

- Log event 20018—with default level `standard`—is output to `stderr` and the log file, and sent to Message Server.

- Log event 20022—with default level `debug`—is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-20009=none
level-reassign-20018=interaction
level-reassign-20022=standard
```

After the log levels are changed:

- Log event 20009 is disabled and is not logged.

- Log event 20018 is output to `stderr` and to the log file.

- Log event 20022 is output to `stderr` and to the log file, and sent to Message Server.

Log-Filter Section

This section must be called `log-filter`. Table 8 describes the option you configure in this section.

Table 8: 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 9 describes the options you configure in this section.

Table 9: Log-Filter-Data Option

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 manner 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
```

Common Section

This section must be named `common`. Table 10 describes the options you configure in this section.

> **Warning!** Use this option only when you are requested to do so by Genesys Technical Support.

Table 10: Common Option

Option	Description
rebind-delay	Specifies the delay, in seconds, between socket-bind operations that are being executed by Data Sourcer. Use this option if the Data Sourcer has not been able to occupy a configured port successfully. Default Value: 10 Valid Values: 0–600 Changes Take Effect: After restart

5

Customizing Data Sourcer

This chapter describes some methods beyond setting configuration options that you can use to customize Data Sourcer. It contains the following section:

- Changing the Default Time Profile, page 57
- Setting Alarm Conditions, page 58

Refer to page 19 for information about setting Data Sourcer configuration options.

Changing the Default Time Profile

There are many reasons that you might want to change Data Sourcer's default time profile so that it does not collect data from Stat Server every 15 minutes. These reasons include:

- Data Sourcer consistently cannot write to ODS.
- The ODS RDBMS is too slow.
- Database space is limited and you want the same information stored in fewer records.

You can change the default time profile in the following manner:

1. Stop Data Sourcer.

2. Within DMA, open the `Statistical Parameters` > `Time Profiles` folder.

3. Select the `CollectorDefault` time profile, open the `Time Profile Constructor` dialog box, and change the time profile's value to one that is more useful.

4. Save your changes by closing the dialog box.

   Note: Stat Server must be running in order for you to save these changes.

5. Restart Data Sourcer.

Genesys does not recommend that you change the `CollectorDefault` value directly within Configuration Manager. Using DMA to accomplish this task automatically synchronizes Data Sourcer's statistical parameter definitions with Stat Server's.

Figure 4 shows that `CollectorDefault` has been redefined to collect data from Stat Server every hour on the hour.
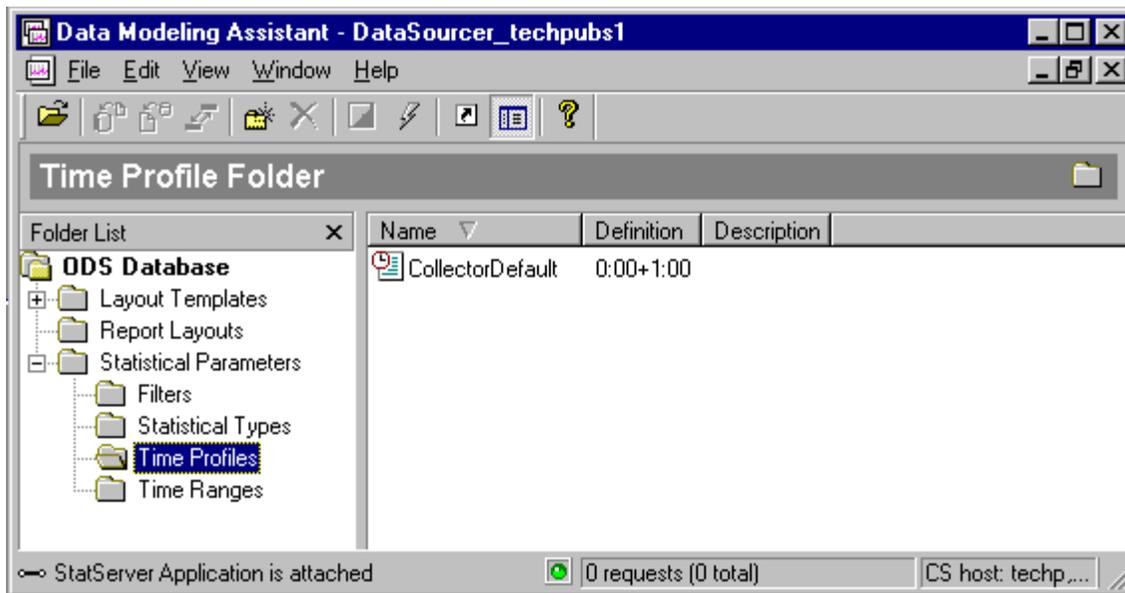
Figure 4: CollectorDefault Redefined

The collection interval that you redefine should be an integral fraction of one hour. For instance, you could redefine `CollectorDefault` to any of:

- `0:00+0:30` (60/30=2)
- `0:00+0:20` (60/20=3)
- `0:00+0:05` (60/5=12)

but not:

- `0:00+0:45` (60/45=4/3)

It makes little sense to define the collection interval as greater than one hour. Such an interval would reduce result details and provide less visibility into contact-center activities.

Setting Alarm Conditions

The 7.0.1 Data Sourcer release introduced four new configuration options and four new corresponding log messages. These options and messages, when used in conjunction with alarm conditions in the Solution Control Interface, will automatically alert you when your Reporting environment becomes stressed and when an acceptable level of activity has resumed.

The options, described in detail beginning on page 23, are:

- `data-excess-high-threshold`
- `data-excess-low-threshold`
- `data-flow-check-interval`
- `data-flow-timeout`

These options indicate under which conditions Data Sourcer should log certain messages. Table 11 lists the corresponding log messages:

Table 11: Data Sourcer Log Messages[*]

Trigger event	Corresponding clearing event
38-30000 No data has been received within the specified threshold from the primary StatServer.	38-30003 Data transfer from the primary StatServer has resumed.
38-30001 Too much data was written to the database within the specified threshold.	38-30004 Data write rate is within the threshold.
38-30006 No data has been received within the specified threshold from the backup StatServer.	38-30007 Data transfer from the backup StatServer has resumed.

[*].   The *Framework Log Events Help* (release 7.2 or higher) describes all Genesys log events, including Data Sourcer's. You can invoke this file directly from the SCI application or open it from the Genesys Technical Support website.

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

The *Framework Solution Control Interface Help* (release 7.2 or higher) 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 5 on page 60 illustrates how you might specify a Detect and Cancel event for a Data Sourcer–specific alarm condition.

Note: You can set alarm conditions for any of the nearly 50 Data Sourcer–specific messages sent to a Data Sourcer log (as you can for any Genesys-specific log message.)
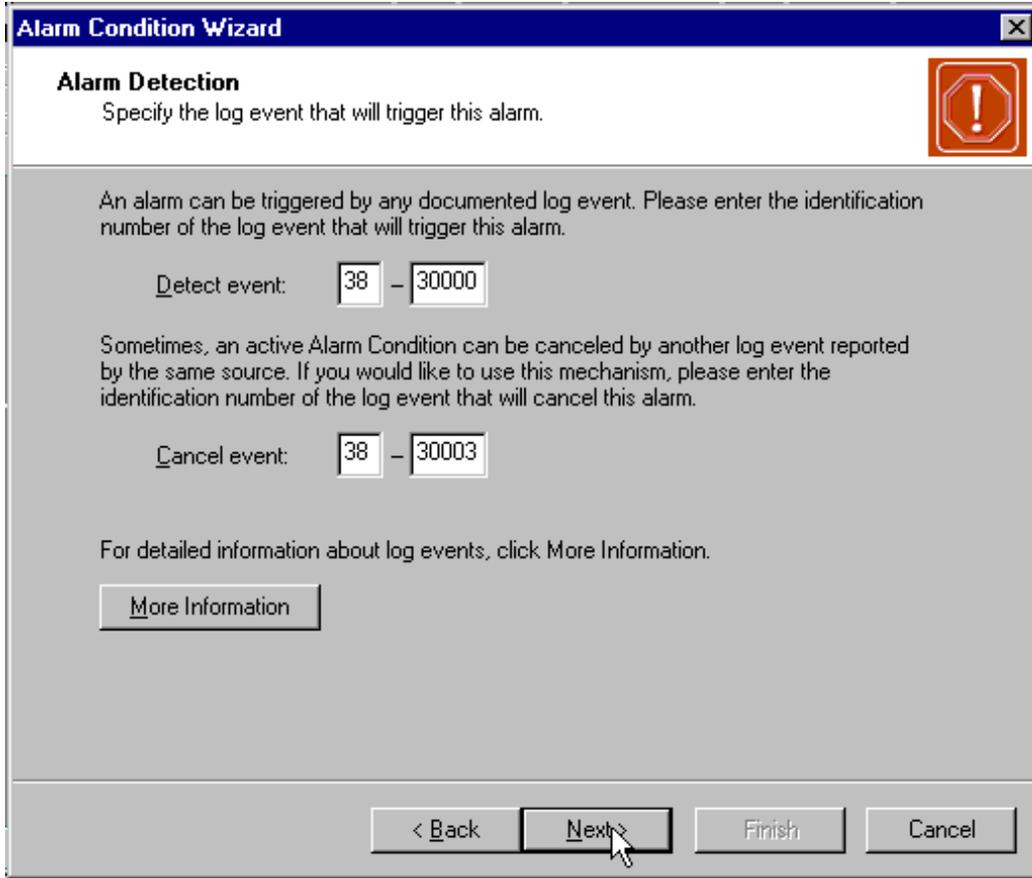
Figure 5: Specifying Detect/Cancel Events for Data Sourcer Within SCI

![Genesys]

Chapter

6

Troubleshooting

This chapter describes the safeguards for data recovery that are incorporated into Data Sourcer for those times when various server connections are dropped. This chapter also provides workaround solutions for some common issues. It contains the following sections:

- Dropped Server Connections, page 61
- Missing Statistics, page 63
- Primary Data Sourcer Failure, page 64
- Out of Memory, page 64
- Invalid Number, page 64

Refer also to the *Framework Log Events Help* for a listing of all possible Data Sourcer log events and actions that you should perform to correct errors.

Dropped Server Connections

Data Sourcer connects to DB Server, Configuration Server (or CS Proxy), and Stat Server. If Data Sourcer becomes disconnected from any of these servers, it attempts to reconnect to it and records an entry in its log file.

Data Sourcer in High-Availability Environment

In the HA reporting architecture, both primary and backup Data Sourcer applications in Hot Standby mode connect to both Stat Server applications, DB Servers, and Configuration Server. If the primary Data Sourcer loses its connection with DB Server, it will stop updating ODS and release control of it. Both the primary and backup Data Sourcer applications will attempt to reconnect to the lost server and gain control of ODS.

A Data Sourcer log file is created in the location that you specified during configuration.

It reports information about the following events:

- Upgrade success/failure

- Server connections

- Data chunks that are stored in emergency save files

- Data transfers from the emergency save files to the database

- Data that is received from Stat Server

- Data that is written to ODS

- Configuration rechecks

- Progress information for various operations

Dropped Stat Server Connection

Both the primary and backup Stat Server applications collect and send the same set of statistics to the primary and backup Data Sourcer applications in the HA reporting architecture. Under normal conditions, the primary Data Sourcer writes to ODS only the data that it receives from the primary Stat Server. If the primary Data Sourcer loses its connection with the primary Stat Server, it will instead write to ODS the data that it receives from the backup Stat Server. This data redundancy in Stat Server means that there will be no loss of historical data in the event of a single dropped Stat Server connection.

In the event of failure to connect with both Stat Server applications, the primary Data Sourcer releases ODS lock and will not try to obtain it again until Data Sourcer restores a valid connection with at least one Stat Server.

If the dropped Stat Server connection affects only the primary Data Sourcer, and the backup Data Sourcer can connect to one of the Stat Server applications, the backup Data Sourcer continues to collect Stat Server data. Backup Data Sourcer obtains ODS lock and the ability to write data to ODS because the primary Data Soucer has released ODS lock.

If both Data Sourcer applications lose their connection to both Stat Server applications, each Data Sourcer will repeatedly attempt to reestablish a connection with the primary and backup Stat Server applications, until a successful connection is established between one of the Data Sourcer applications and one of the Stat Server applications.

Dropped Configuration Server Connection

If Configuration Server (or CS Proxy) becomes disconnected from Data Sourcer, Data Sourcer:

- Attempts to reconnect to Configuration Server a specified number of times (see the application's Server Info properties within the Configuration Manager for the user-defined number of reconnect attempts and timeout).

- Connects to the backup Configuration Server.

A dropped Configuration Server connection is not considered critical in the HA reporting architecture. The primary Data Sourcer continues operating in the same mode, while it tries to reestablish the connection with Configuration Server. There is *no* switchover to the backup Data Sourcer in this case.

Dropped DB Server Connection

If the connection to the DB Server is lost, or if a DB failure is detected, the primary Data Sourcer releases control of ODS. Both the primary and backup Data Sourcer applications will attempt to reconnect to the primary or backup DB Server by using the standard Framework algorithm.

During the time that the connection is lost, each Data Sourcer writes the data that it receives from Stat Server to a queue that it holds in memory, which can be extended to one or more emergency-save files. If the emergency-save-directory configuration option is set within Configuration Manager, the emergency save files are saved to the location that is specified by this option; otherwise, the data is saved to Data Sourcer's current working directory.

When one or both of the Data Sourcer applications is able to reconnect to a DB Server, each Data Sourcer will compete again for control of ODS. The data from the memory queue of the Data Sourcer that becomes primary, is automatically written to ODS. The backup Data Sourcer will delete the data in its queue, after it has determined that all the data that is stored in memory during the DB Server failure has been successfully written to ODS.

Missing Statistics

If for some reason your Stat Server Application object does not have all statistical types (stat types, for short), time ranges, filters, or time profiles that were initialized in ODS, Data Sourcer might experience errors while it collects data from Stat Server. This might occur if, for example, you manually configured your Data Sourcer Application object incorrectly. (For this reason, Genesys recommends that you use the Reporting configuration wizards to configure Reporting Application objects.)

You can restore the missing filters, time profiles, and time ranges to Stat Server by using any of the following methods:

•   Reinitialize your ODS and Data Mart.

•   Use the Synchronize button in DMA.

•   Add the missing stat types manually to Stat Server's options.

•   Remove the missing statistic from your reports.

If you are using one Data Sourcer Application object to report activity for more than one Genesys solution, you might encounter similar errors because Data Sourcer, as packaged, is designed to run against one solution.

Primary Data Sourcer Failure

If the primary Data Sourcer fails, it releases control of ODS. As soon as the backup Data Sourcer detects this failure, it will attempt to gain control of ODS; if it is successful, it becomes primary. The new primary Data Sourcer then checks its memory queue and emergency save files for data chunks and submits them to ODS.

The failure of the primary Data Sourcer and the switchover of the backup to the primary mode do not impact the collection of object data or the appearance of the data on a layout. Both the primary and backup Data Sourcer applications receive notifications from Configuration Server about new, changed, and deleted objects, and they apply these changes immediately.

Data Modeling Assistant also notifies both the primary and backup Data Sourcer applications when report layouts are activated and deactivated, thus, it ensures that if a switchover occurs, the new primary Data Sourcer will have the correct layouts.

Out of Memory

If Data Sourcer encounters a low memory condition, it can exit unexpectedly. If this happens, increase the memory that is located to Data Sourcer by installing additional memory or increasing the swap size.

On UNIX operating systems, make sure that the data segment size limit is appropriate. Before you start your Data Sourcer application, run the following commands:

- For CSH/TCSH shell: `limit datasize unlimited`
- For SH/BASH shell: `ulimit -d unlimited`

For more information about how to set these limits, or make these settings permanent for your system, contact your System Administrator.

Invalid Number

For certain combinations of regional settings of DB Server, you might encounter an `ORA-01722: invalid number` error when Data Sourcer attempts to write data to an Oracle-based ODS. This error can occur when certain regional settings, such as the decimal point character, differ from those that are defined within the RDBMS.

If you encounter this `invalid number` error, either reset your regional settings or set the `NLS_NUMERIC_CHARACTERS` Oracle option to include both a decimal point and a comma—for example,`".,"`.

7

Emergency-Save Files

Data Sourcer writes to one or more emergency save files whenever it is unable to write to ODS, or whenever memory is adversely affected.

In release 7.2.*x* of Data Sourcer, you specify the *location* of the emergency-save files by setting the `emergency-save-directory` option during Data Sourcer configuration.

In the following sections, this chapter describes Data Sourcer functionality with respect to this file:

Algorithm

Data Sourcer uses a FIFO (first-in, first-out) queue of data chunks that consists of the memory queue and one or more emergency-save files. Since the 7.2.003.05 release of Data Sourcer, both the primary and backup Data Sourcer applications store data in their respective memory queues. Any excess data is stored in emergency save files, in a directory that you specify during configuration.

The size of the memory queue is limited only by the amount of memory that is available to the Data Sourcer application. The number of active layouts multiplied by two determines the number of complete data chunks that are stored in the queue. When this number multiplied by two is reached, only the headers of successive data chunks are stored in memory, and collected data (corresponding to each header) is stored in the chunk files that reside in the emergency-save folder. This allows for improved performance of the backup Data Sourcer, as it no longer has to scan the entire contents of the emergency-save folder and read the chunk files when it checks for chunks that are already in ODS.

Note: During manual configuration of the backup Data Sourcer, you must set the `emergency-save-directory` configuration option to specify where the emergency-save files will be stored. If this option is not set, these files will instead be written to Data Sourcer's current working directory. Refer to page 26 for more information about this option.

Each chunk of data is stored in a separate emergency save file—one chunk per file. The primary Data Sourcer checks each chunk for duplicate keys before it submits it to ODS. If the primary Data Sourcer is unable to write a data chunk to ODS, it will continue to try until it is successful.

The backup Data Sourcer deletes the data that is in its memory queue and emergency-save files only after it ensures that the primary Data Sourcer has successfully written the corresponding data chunks to ODS.

Note: The queue size of the backup Data Sourcer can become larger than that of the primary Data Sourcer, because it retains its data in the queue longer while it waits for the primary Data Sourcer to write the corresponding data to ODS.

When connection to DB Server is lost, the size of the memory queue will increase continuously as long as Data Sourcer remains connected to Stat Server and continues to receive statistical data from it.

Chunk File Format

You can specify the format of chunk files as either `text` or `binary`. In earlier releases of Data Sourcer (prior to 7.2.003.05), chunk files were stored in text format only, which produced comma-separated-value (CSV) formatted files. Data Sourcer now supports an additional binary-file format that provides improved processing speed. The binary-file format produces non-readable (to users) binary-data files that are specific to the operating system and the CPU type.

Important Do not use chunk files that have been generated on different operating systems, because this might cause unpredictable errors and incorrect data in ODS. For example, chunk files that have been generated on a Pentium 4 computer that is running Windows 2003 cannot be used with a Data Sourcer application that is running on a computer that has an UltraSPARC CPU and Solaris 10.

The format of the chunk files must be specified before you start the Data Sourcer application by setting the `chunk-file-format` configuration option (see page 23). If this option is not set, or an incorrect value is specified, Data Sourcer uses the default `text` format. It is important that you take care when you change the format of the chunk file. Make sure that there are no unprocessed chunk files before you change this option because Data Sourcer will not recognize chunk files in a format that is different from that which was specified at startup.

> **Note:** The binary file format is not compatible with releases of Data Sourcer prior to 7.2.003.05. Please consult Genesys Technical Support if you need to convert chunk files from binary to text format, or vice versa.

Resolving Regular Use of the Emergency Save Files

If Data Sourcer regularly writes data chunks to the emergency-save files, you have a number of options for resolving the problem, including the following:

- Check ODS availability. If ODS size has reached its limit, flush data to ETL Runtime by using the `dropTransferredTables` transformation parameter. (This feature is different from ETL Runtime's Purge utility.)

- Reduce the number of open statistics or increase the interval in the time profile (see "Changing the Default Time Profile" on) because your ODS network might not be fast enough to handle the specified statistical load. You can also reduce the number of open statistics by deactivating some report layouts or deleting some statistics from custom report layouts.

Extracting Data from the Emergency Save Files

You *cannot* manually extract data from emergency-save files. Under normal operating conditions, Data Sourcer automatically restores data from emergency-save files into ODS. No manual intervention is required.

> **Note:** Please contact Genesys Technical Support for analysis of your problem and assistance in recovering data.

Deleting the Emergency-Save Files

The emergency-save files are automatically deleted after the data has been transferred to ODS.

![Genesys logo]

8 Application Files

The Data Sourcer installation routine creates a Data Sourcer application folder that contains the following two subfolders:

- DataSourcer
- database

Tables 12 and 13 in this chapter describe the files that comprise each subfolder.

Warning! Do not attempt to run the SQL scripts manually because of the potential for data loss. They are intended for Data Sourcer's internal use and for advanced database administrators.

Table 12: Contents of the DataSourcer Subfolder

File name	Description
common.lms	File that stores common log messages
COPYING	Licensing information for the pthreads-win32 library for Windows platforms.
COPYING.LIB	Text of the LGPL 2.0 license that covers the pthreads-win32 library for Windows platforms.
data_sourcer.exe	Application executable for Windows platforms
data_sourcer	On UNIX platforms, link to the Data Sourcer executable.
data_sourcer_32	On UNIX platforms, Data Sourcer executable for 32-bit platforms.
data_sourcer_64	On UNIX platforms, Data Sourcer executable for 64-bit platforms.
DataSourcer.lms	File that stores Data Sourcer–specific log messages
database	Subfolder. (See Table 13.)

Table 12: Contents of the DataSourcer Subfolder (Continued)

File name	Description
im_statserver_template.cfg	Stat Server configuration file that contains application time profile options and statistical types for Data Sourcer to function properly. This file is used for installation.
ip_description.xml	File that contains the IP descriptions of common IP parameters including: • Name, version, build date, build number and release type of IP • Language that is supported by the product [for example, English (United States), Korean, and so on]. • A short functional description of the product. • A list of supported computer types and operating systems. • Files names and locations of product release notes, wizard advisory and deployment procedure.
ospatchlist.txt	On UNIX platforms, list of patches that are used for the build.
pthreadVC2.dll	POSIX threads library for Microsoft Windows. Pthreads-win32 is covered under the terms of the GNU Lesser General Public License.
read_me.html	File that contains general information about the installation package
run.sh	On UNIX platforms, shell script that launches Data Sourcer.
startServer.bat	On Windows platforms, batch file that starts Data Sourcer as a regular application, not as a Windows Service.

Table 13: Contents of the database Subfolder

File name	Description
alter_1_51502.sql alter_1_610.sql alter_1_650.sql alter_1_700.sql alter_1_70100.sql alter_1_720.sql alter_1_723.sql alter_1_724.sql alter_1_725.sql alter_1_726.sql alter_1_727.sql alter_1_728.sql	SQL scripts for internal use only against a Sybase database

Table 13: Contents of the database Subfolder (Continued)

File name	Description
alter_2_51501.sql alter_2_51502.sql alter_2_610.sql alter_2_650.sql alter_2_700.sql alter_2_70100.sql alter_2_720.sql alter_2_723.sql alter_2_724.sql alter_2_725.sql alter_2_726.sql alter_2_727.sql alter_2_728.sql	SQL scripts for internal use only against an Oracle database
alter_3_610.sql alter_3_650.sql alter_3_700.sql alter_3_70100.sql alter_3_720.sql alter_3_723.sql alter_3_724.sql alter_3_725.sql alter_3_726.sql alter_3_727.sql alter_3_728.sql	SQL scripts for internal use only against an IBM DB2 database
alter_6_51502.sql alter_6_610.sql alter_6_650.sql alter_6_700.sql alter_6_70100.sql alter_6_720.sql alter_6_723.sql alter_6_724.sql alter_6_725.sql alter_6_726.sql alter_6_727.sql alter_6_728.sql	SQL scripts for internal use only against a Microsoft SQL Server database
cc_templates.sql	SQL script that inserts the layout templates that are used by an IS Data Sourcer (Data Sourcer in Cartridge mode) into ODS
crebas_1.sql	SQL script that creates tables and indexes for a Sybase database
crebas_2.sql	SQL script that creates tables and indexes for an Oracle database

Table 13: Contents of the database Subfolder (Continued)

File name	Description
crebas_3.sql	SQL script that creates tables and indexes for an IBM DB2 database
crebas_6.sql	SQL script that creates tables and indexes for a Microsoft SQL Server database
cretrg_1.sql	SQL script that creates triggers and procedures for a Sybase database
cretrg_2.sql	SQL script that creates triggers and procedures for an Oracle database
cretrg_3.sql	SQL script that creates triggers and procedures for an IBM DB2 database
cretrg_6.sql	SQL script that creates triggers and procedures for a Microsoft SQL Server database
init_time_profiles.bin	SQL script that initializes Contact Server time profiles
layout_templates.sql	SQL script that, for new tenants, creates default report layouts from corresponding layout templates
ods.rtf	"Standard CDM [Conceptual Data Model] Report"
ods-db2.rtf	"Standard PDM [Physical Data Model] Report" for an IBM DB2 database
ods-mssql.rtf	"Standard PDM Report" for a Microsoft SQL Server database
ods-oracle.rtf	"Standard PDM Report" for an Oracle database
ods-sybase.rtf	"Standard PDM Report" for a Sybase database
solution_types.sql	SQL script that redefines the solution type ID of 15 to Multimedia
stattypes_detail.sql	SQL script that updates tables with statistical type descriptions

Genesys®

Related Documentation Resources

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

Solution Reporting

- *Reporting Deployment Guide*, for information about configuring and installing Data Sourcer, as well as other Reporting components, using the Historical Reporting Wizard. This guide also provides guidelines for appropriately sizing your ODS, and for starting and stopping Data Sourcer on Windows and UNIX platforms.

- Standard Physical Data Model for your database type, for ODS table structure and field definitions. This Microsoft Word file is deployed during Data Sourcer installation, in the `database` subdirectory.

- *Reporting 7.6 Data Modeling Assistant Help,* for information about importing layout templates and activating and deactivating report layouts.

- *Reporting 7.6 ETL Runtime User's Guide*, for information about the Data Mart Services and, specifically, the `dropTransferredTables` runtime parameter used for transferring data from ODS to the Data Mart.

- *Reporting Reference Manual,* for information about what you can expect with regard to Data Sourcer performance and how to compute the maximum number of requests that your Reporting environment can handle.

- *Reporting 7.6 Master Index,* which will help you find where other related topics are documented in the Reporting 7.6 documentation set.

Management Framework

- *Framework Combined Log Events Help*, for a complete listing of Data Sourcer log messages, and for information about what to do when you encounter errors in Data Sourcer logs.

- *Framework Solution Control Interface Help,* for information about defining alarm conditions for Data Sourcer log events.
- *Framework Configuration Manager Help,* for information about configuration sections and how to edit configuration options.

Real-Time Metrics Engine

- *Framework Stat Server User's Guide,* for information about time profiles and the role that Stat Server plays in fulfilling requests for statistics made from the Data Collection Services.

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 Technical Support for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support`.

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

- *Genesys Supported Operating Environment Reference Manual*
- *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 8.*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.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the `system level documents by release` tab in the Knowledge Base `Browse Documents` Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at `http://genesyslab.com/support`.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesyslab.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_datasourcer_07-2012_v7.6.102.00

You will need this number when you are talking with Genesys Technical Support 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 14 describes and illustrates the type conventions that are used in this document.

Table 14: 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 76).	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 14: 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

Symbols

A

B

C