



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Interaction Server Administration Guide

Event Logger Options

12/17/2025

Contents

- 1 Event Logger Options
 - 1.1 Preparing DAP configuration objects

Event Logger Options

This section provides short descriptions of the DAP options that configure the Event Logger's behavior. See the [eServices 8.1 Reference Manual](#) for full details.

logger-settings Section

`delivery-protocol`—Defines type of the event logger. Possible values are:

- `odbc` — For using Event Logger database scripts
- `mq-series` — For the MQ-Series message queue system
- `msmq` — For the MSMQ message queue system
- `jms` — For a JMS queue
- `groovy` — For the Groovy Event Logger

Important

Since Interaction Server 8.5.3 value for this option must be specified always. And since this version value “event-log” is not supported.

`batch-size`—Defines the minimum number of records to store in internal memory before flushing to the database. Valid values are 1-5,000; the default is 500.

`storing-timeout`—Defines a time interval, in milliseconds, between operations of writing to the database. Valid values are 500-60,000; the default is 1,000.

Important

`storing-timeout` and `batch-size` define limits that trigger writing to the database: writing takes place as soon as one or the other is reached.

`max-queue-size`—Defines the maximum number of records that are kept in memory while waiting to be written to the database. If the number of records exceeds this maximum, the data are discarded from memory and are not written to the database. Valid values are 10,000-100,000; the default is 20,000.

`schema-name`—Specifies the name of the schema used to access the database.

custom-events Section

This section contains options that list custom events by their identifiers and specify which table (interaction, agent or custom) stores them.

event-filtering Section

This section contains seven options, six of which are named for one of the endpoint types that is referred to in the [classification of events](#):

log-agent-state
log-agent-activity
log-queue
log-strategy
log-esp-service

With the value false, events associated with the named endpoint type are filtered out. For example, setting log-queue to a value of false prevents the events EventPlacedInQueue, EventPlacedInWorkbin, EventTakenFromQueue, and EventTakenFromWorkbin from being stored. The remaining two options in this section are:

- log-userdata—With the value false, data from custom fields is filtered out.
- event-filter-by-id—A list of comma-separated event identifiers. Only these events are stored in Event Logger. If this option is not present or contains no event identifiers, event filtering by identifier is not applied.

These event identifiers are listed in the [Platform SDK API Reference for .NET \(or Java\)](#). For example, the identifier of EventRejected is 168.

Custom Data Sections

The five sections contain options specifying a list of events that are to be stored in custom fields of the event logger database. All five work identically, the differences being (a) the events from which the user data is taken and (b) the database table that stores them. These differences are shown in the following table.

Section	Source Event	Logger Database Table
itx-custom-data	UserData and EventContent attributes of interaction-related reporting events	rpt_interaction
esp-custom-data	UserData attribute of EventExternalServiceRequested and EventExternalServiceResponded	rpt_esp
esp-service-data	Envelope3rdServer attribute of	rpt_esp

Section	Source Event	Logger Database Table
	EventExternalServiceRequested and EventExternalServiceResponded	
agent-custom-data	EventContent attribute of EventCustomReporting	rpt_agent
custom-customdata	EventContent attribute of EventCustomReporting	rpt_custom

For an explanation of the `Envelope3rdServer` attribute, see [Platform SDK API Reference for .NET \(or Java\)](#).

To use these options, you must first add a field to the appropriate Event Logger database table. Its data type must be the same as that of the mapped user data key. In these sections, the options have the following characteristics:

- The name is a user data key name (case-sensitive).
- The value is three semicolon-separated strings, which specify the following:
 1. The name of the field that you added to the database table. This value is required.
 2. The data type: string, integer, or timestamp. The default is string, with default length 64. If your data type is other than string, or if it is string and you want to specify a non-default length (next item), this value is required.
 3. Optionally, the length. The default for the string type is 64. There are no default values for integer and timestamp.

For example, if you have a data key called `CustomerSegment`, you can add a custom field to store this data as follows:

1. Add a field called `customer_segment` to the `rpt_interaction` table.
2. In the `itx-custom-data` section, create an option called `CustomerSegment`.
3. Give it this value: `customer_segment;string;64`.

Since string and 64 are the default values for type and length respectively, the value of this option could also be simply `customer_segment`.

Preparing DAP configuration objects

New environment

For a new environment, create a Database Access Point (DAP) object as described in the [Framework Database Connectivity Reference Guide](#) and add the DAP to the connections of your Interaction Server.

To configure the database-oriented [Event Logger](#) for Interaction Server, create a second DAP object as described on the [Deploying Event Logger](#) page.

Existing environment

For an existing environment with Interaction Server 8.5.1 - 8.5.3 and DB Server, you should configure them to use ODBC to connect to database before the upgrade to version 9.0.0. Use Genesys Administration Extension (GAX) to create copies of the existing DAP objects used by Interaction Server. You can use these copies to switch back to connecting via DB Server while testing.

Configure the DAP

This section applies to Interaction Server and **Event Logger**. Related information about Database Capture Point is in a **separate location**.

For an ODBC connection, you must configure the Database Access Point (DAP) associated with the Application in question, as follows:

1. For Database Info configuration, in Genesys Administrator, enter the following in the **[DB Info]** section of the **Configuration** tab (in Configuration Manager, on the **DB Info** tab):
 - DBMS Name - host name of the database server
 - DBMS Type - type of the database
 - Database Name - name of the database
 - User Name - user name to use to connect to the database
 - Password - password for the user name account
2. Configure the options of Database Access Point application to use ODBC.
 - For Interaction Server database DAP [settings]
`dbprotocol=odbc`
`connection-string = <your connection string>`
 - For Database Event Logger DAP [logger-settings]
`delivery-protocol =odbc`
`connection-string = <your connection string>`
3. Determine the Connection string value.

Configure ODBC driver in connection string

To make configuration easier, Interaction Server will try to build the ODBC connection strings itself based on the information that is available in the DAP. During this process, Interaction Server uses the following default names for the drivers:

Database Type	Default Driver
MS SQL	{SQL Server Native Client 10.0}
Oracle	{Oracle in OraClient11g_home1}
PostgreSQL	{PostgreSQL ANSI(x64)}

Important

Genesys strongly recommends that you use latest versions of ODBC drivers that were released officially for your database. The main purpose of default drivers is backward compatibility with old database.

If your configuration uses a different driver name, you must explicitly provide the actual driver name in value of **connections-string** option in section **[settings]** (or **[logger-settings]** for the Event Logger DAP).

The value of **connection-string** is a list of key-value pairs separated by semicolons (;).

- To specify the actual driver, give it a value of `driver=<driver_name>`.
- If your database server is not running on the default port (1433 for MSSQL and 1521 for Oracle), you must also provide the port number by adding `port=<actual_port>` pair to the the value of connection-string. For example, *Driver=SQL Server Native Client 11.0;port=1433*.

Important

For Oracle, ensure the `LOB=T` option is present in the connection string if it's not specified in DSN (see the next section).

DSN

You can also have Interaction Server use a DSN that has been configured in your system. To do this, locate or create the **connection-string** option in the **[settings]** section of the DAP (the **[logger-settings]** section in the Event Logger DAP), and set it to `DSN=<name_of_the_dsn>`. With this method, the user name and password to connect to the database are taken from the settings on the **DB Info** tab and do not need to be provided in the DSN properties.