Genesys®

Framework 8.1

DB Server

User's Guide

About Genesys

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

Each product has its own documentation for online viewing at the Genesys Ö[&˘ { ^} œæą } 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, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. © 2014 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 Customer Care. Before contacting Ô˘ • ȼ { ^¦ Áˆ Õæ˘ ^ ˇ Ép lease refer to the **_Genesys Care Program Guide_** for complete contact information and procedures.

Ordering and Licensing Information

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

Released by

Genesys Telecommunications Laboratories, Inc. www.genesys.com

Document Version: 81fr_us-dbs_02-2014_v8.1.301.00

Table of Contents

List of Procedures

Genesys

Preface

Welcome to the *Framework 8.1 DB Server User's Guide.* This document provides a high level overview of DB Server, with architecture and configuration information for a DB Server that is a client of Configuration Server.

This document is valid only for the 8.1 release(s) of this product.

Note: For versions of this document created for other releases of this product, visit the Genesys Fqewo gpvcvkqp website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at `orderman@genesys.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 page 89.

About DB Server

DB Server provides the interface between Genesys applications and the database management system on which the databases for the operations of Genesys solutions reside.

Intended Audience

This document is primarily intended for system administrators and database architects. It has been written with the assumption that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications
- Network design and operation
- Your own network configurations

You should also be familiar with:

- Genesys Framework architecture and functions.
- The Genesys configuration environment, including basic operations in Genesys Administrator or Configuration Manager.

Making Comments on This Document

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

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

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

Contacting Genesys 7i gʰca Yfˉ7UfY

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

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

Document Change History

This is the first release of the *Genesys 8.1 DB Server User's Guide*. In the future, this section will list topics that are new or have changed significantly since the first release of this document.

![Genesys]

1

Deploying DB Server

This chapter describes how to configure and install a DB Server that is a client of Configuration Server, and how to configure Database Access Points for the DB Server.

This chapter contains the following topics:

Note: Before deploying DB Server, you must install DBMS client software on the DB Server host computer. Refer to the DBMS vendor's documentation for details. This document also provides some specific recommendations in Chapter 2 on page 27.

Introduction

Within the Framework architecture, DB Server provides the interface between Genesys applications and the database management system (DBMS) you are using to manage the databases required to operate a Genesys solution.

DB Server can run in two modes, as follows:

- Independent of Configuration Server—DB Server provides Configuration Server with access to the Configuration Database; reads its configuration settings from a configuration file.

- As a Configuration Server client—DB Server provides access to various databases, which means you must configure Database Access Points (DAPs). DB Server can therefore be a client of applications other than

Configuration Server. In this mode, DB Server reads its configuration settings via Configuration Server from the Configuration Database.

This chapter contains instructions for installing and configuring DB Server as a client of Configuration Server, and its associated DAPs. For instructions on installing DB Server to run independently of Configuration Server, refer to the *Framework 8.1 Deployment Guide*.

Note: For the rest of this document, the term *DB Server* means *DB Server as a client of Configuration Server*.

DB Server Failure

Starting in release 8.0, the Management Layer can now use hang-up detection to detect an unresponsive DB Server application. Users can then configure appropriate actions, including alarms if required.

To enable hang-up detection for DB Server, use the configuration option `heartbeat-period` to set the time interval in which a heartbeat message must be received before DB Server is considered to be unresponsive. Use the option `hangup-restart` to define the action that Management Layer takes when it deems DB Server to be non-responsive, either automatically restarting DB Server or just generating a notification of the situation.

For more information about this functionality, see the *Framework 8.1 Management Layer User's Guide*. For more information about the configuration options, refer to the *Framework 8.1 Configuration Options Reference Manual*.

Warning! Use this functionality with great care. Failure to use it properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

What Is a Database Access Point?

To cover the variety of ways in which you can interface applications in the Genesys installation with databases, the Configuration Layer uses the concept of a Database Access Point.

A *Database Access Point* (DAP) is an object of the `Application` type that describes both the parameters required for communication with a particular database and the parameters of DB Server through which this database can be accessed. If, according to your configuration, a database can be accessed through multiple DB Servers simultaneously, register as many DAPs as there are DB Servers.

This chapter describes how to configure a DAP when you use DB Server to access a database. When you configure a DAP for DB Server, configure only the fields applicable to a DB Server connection.

> **Note:** No physically installed applications correspond to DAPs.

Task Summary

The following table summarizes the steps required to deploy a DB Server as a client of Configuration Server.

Task Summary: Deploying DB Server as a Client of Configuration Server

Objective	Related Procedures and Information
1. Configure an `Application` object for DB Server.	Use one of the following procedures, as appropriate: • "Configuring a client DB Server Application object using Genesys Administrator" on page 16 • "Configuring a client DB Server Application object using Configuration Manager" on page 17
2. Configure a Database Access Point.	Use one of the following procedures, as appropriate: • "Configuring a Database Access Point for a Client DB Server using Genesys Administrator" on page 19 • "Configuring a Database Access Point for Client DB Server using Configuration Manager" on page 21
3. Install DB Server on the Host specified during configuration.	Do one of the following: • To install on UNIX, use the procedure "Installing Client DB Server on UNIX" on page 23. • To install on Windows, use the procedure "Installing Client DB Server on Windows" on page 24.
4. Start DB Server.	See "Starting DB Server" on page 25.

Configuring DB Server

For a DB Server to function as a client of Configuration Server, you must configure an `Application` object of type `DB Server`, using Genesys Administrator or Configuration Manager.

Procedure:
Configuring a client DB Server Application object using Genesys Administrator

Purpose: To configure DB Server to function as a client of Configuration Server.

Prerequisites

- Genesys Administrator is installed and running.
- The Host object upon which this DB Server will reside has been configured. Refer to the *Framework 8.1 Deployment Guide* to obtain instructions for configuring Host objects.
- A database exists to which this DB Server will provide access.

Start of procedure

1. In Genesys Administrator, go to Provisioning > Environment > Applications, and select New in the toolbar. This opens a Browse dialog box that lists the available application templates. If a DB Server template file is not listed, do one of the following:
   - Import the DBServer_<current-version>.apd file from the Management Framework 8.1 product CD.
   - Create a new template using the instructions provided in the *Framework 8.1 Deployment Guide*, and repeat this step.

2. In the Browse dialog box, select the DB Server template file. The Configuration tab for the new DB Server Application object appears in the Details panel.

3. In the General section, enter a descriptive name in the Name field

4. In the Server Info section:
   a. In the Host field, click the magnifying glass icon to select the Host object on which this DB Server is running.
   b. For each listening port that an application must use to connect to this DB Server:
      i. In the Connections list, click Add.
      ii. Enter the port properties in the Port Info dialog box. To secure connections to this port, refer to the *Genesys Security Deployment Guide*.
      iii. Click OK.

    c. For the `Working Directory`, `Command Line`, and `Command Line Arguments` fields, do one of the following:

- Enter the appropriate information in the three text boxes. For information about command-line parameters, see the *Framework 8.1 Deployment Guide*.

- Type a period (`.`) in the `Working Directory` and `Command Line` text boxes, and leave the `Command Line Arguments` text box blank. The information will be filled in automatically when you install this DB Server, but only if the Installation Package can connect to Configuration Server.

5. On the `Options` tab:

    a. In the drop-down list in the top-right corner, select `Advanced View (Options)` if not already selected.

    b. In the `dbserver` section:

        i. Change the value of the `dbprocess_name` option to the value of the option `<DBMS you are using>_name`. For example, if you are using Microsoft SQL Server DBMS, set the value of `dbprocess_name` to `./dbclient_msql`.

        ii. Change the value of the `management-port` option to the management port number for this DB Server.

    c. Change existing, or define new, configuration options as required. For option descriptions, see Chapter 3 on page 33 and Chapter 5 on page 45.

6. Click `Save and Close` to save the new DB Server `Application` object.

End of procedure

Procedure:
Configuring a client DB Server Application object using Configuration Manager

Purpose: To configure DB Server to function as a client of Configuration Server.

Prerequisites

- Configuration Manager is installed and running.

- The `Host` object upon which this DB Server will reside has been configured. Refer to the *Framework 8.1 Deployment Guide* to obtain instructions for configuring `Host` objects.

- A database exists to which this DB Server will provide access.

Start of procedure

1. In Configuration Manager, right-click the `Environment > Applications` folder and select `New > Application`, which opens the `Browse` dialog box that lists the available application templates. If a DB Server template file is not listed, do one of the following:

    - Import the `DBServer_<current-version>.apd` file from the Management Framework 8.1 product CD.

    - Create a new template using the instructions provided in the *Framework 8.1 Deployment Guide*, and repeat this step.

2. In the `Browse` dialog box, select the DB Server template file, which opens the `Properties` dialog box for the new DB Server `Application` object.

3. On the `General` tab, enter a descriptive name in the `Name` text box

4. On the `Server Info` tab:

    a. Click the `Browse` button next to the `Host` drop-down list, and select the host on which this DB Server will run.

    b. For each listening port to which clients can connect to this DB Server:

        i. In the `Ports` section, click `Add Port`. The `Port's Properties` dialog box opens.

        ii. Specify the `Port ID` and the `Communication Port` that applications will use to connect to this DB Server.

        iii. To secure connections to this port, refer to the *Genesys Security Deployment Guide*.

    c. Leave the rest of the fields at their default values.

5. On the `Start Info` tab, do one of the following:

    - Enter the appropriate information in the three text boxes. For information about command-line parameters, see the *Framework 8.1 Deployment Guide*.

    - Type a period (`.`) in the `Working Directory` and `Command Line` text boxes, and leave the `Command Line Arguments` text box blank. The information will be filled in automatically when you install this DB Server, but only if the Installation Package can connect to Configuration Server.

6. On the `Options` tab, in the `dbserver` section:

    a. Change the value of the `dbprocess_name` option to the value corresponding to the `<DBMS you are using>_name` option. For example, if you are using Microsoft SQL Server DBMS, set the value `dbprocess_name` to `./dblient_msql`.

    b. Change the value of the `management-port` option to the number of the management port for this DB Server.

    c. Change existing, or define new, configuration options as required. For option descriptions, see Chapter 3 on and Chapter 5 on .

7. Click OK.

End of procedure

Configuring Database Access Points

For any application to access the Database, you must configure one or more Database Application Points (DAP) as Application objects, using Genesys Administrator or Configuration Manager.

Procedure:
Configuring a Database Access Point for a Client DB Server using Genesys Administrator

Prerequisites

*   DB Server is installed and running.
*   Genesys Administrator is installed and running.

Start of procedure

1. In Genesys Administrator, go to Provisioning > Environment > Applications, and select New in the toolbar. This opens a Browse dialog box with a list of available application templates. If a Database Access Point template file is not listed, do one of the following:
    *   Import the Database_Access_Point_<current-version>.apd file from the Management Framework 8.1 product CD.
    *   Create a new template using the instructions provided in the *Framework 8.1 Deployment Guide*, and repeat this step.

2. In the Browse dialog box, select the DAP template file. The Configuration tab for the new DAP Application object appears in the Details panel.

3. In the General section, enter a descriptive name in the Name field.

    A DAP can have the same name as the database itself. However, it is recommended that you make their names unique if you are using multiple access points for the same database.

4. In the Host field of the Server Info section, click the magnifying glass icon to select the Host object to which this DAP is assigned.

5. In the DB Info section, provide information about the database in the following fields:
    *   Connection Type—The type of connection to the DBMS. Select Default.

> **Note:** Do not select a `Connection Type` of `JDBC` for Database Access Points.

- `DB Server`—The `Application` object corresponding to the database to which this DAP will provide access.

  > **Note:** A DAP has the same listening ports as the DB Server to which it provides access.

- `DBMS Name`—The name or alias identifying the DBMS that handles the database. The value of this option is communicated to DB Server so that it connects to the correct DBMS:
  - For DB2, set this value to the name or alias-name of the database specified in the db2 client configuration.
  - For Informix, this value is the name of SQL server, specified in the `sqlhosts` file.
  - For Microsoft SQL, set this value to the SQL server name (usually the same as the host name of the computer on which Microsoft SQL runs).
  - For Oracle, the value is the name of the Listener service.
  - For PostgreSQL, set this value to the SQL server name (usually the same as the host name of the computer on which PostgreSQL runs).
  - For Sybase, this value is the server name stored in the Sybase interface file.
- `DBMS Type`—The type of DBMS that handles the database. You must set a value for this property.
- `Database Name`—The name of the database to be accessed, as it is specified in the DBMS that handles this database. You must set a value for this property unless `oracle` or `db2` is specified as the `DBMS Type`. For Sybase, Informix, and Microsoft SQL, this value is the name of the database to which the client will connect.
- `User Name`—The user name established in the SQL server to access the database. You must set a value for this property.
- `User Password`—The password established in the SQL server to access the database.
- `Re-enter Password`—Confirmation for the value entered for `Password`.
- `Case Conversion`—Case conversion method for key names of key-value lists coming from DB Server. This value specifies whether and how a client application converts the field names of a database table when receiving data from DB Server. If you select `upper`, field names are converted into uppercase; if you select `lower`, field names are converted into lowercase; and if you select `any`, field names are not

converted. This setting does not affect the values of key-value lists coming from DB Server. That is, actual data is being presented exactly as in the database tables.

> **Note:** For the Case Conversion option, use the default value (`any`) unless directed to do otherwise by Genesys Ewuvqo gt'Ectg.

6. Click `Save and Close` to save the new DAP `Application` object.

End of procedure

To interface an application with a database through a certain DAP, add this DAP to the `Application` object's `Connections` list.

Procedure:
Configuring a Database Access Point for Client DB Server using Configuration Manager

Prerequisites

*   Configuration Manager is installed and running.
*   Client DB Server is installed and running.

Start of procedure

1. In Configuration Manager, right-click the `Environment > Applications` folder and select `New > Application`, which opens the `Browse` dialog box with the available application templates. If a DAP template file is not listed, do one of the following:
   *   Import the `Database_Access_Point_<current-version>.apd` file from the Management Framework Product CD.
   *   Create a new template using the instructions provided in the *Framework 8.1 Deployment Guide*, and repeat this step.

2. In the `Browse` dialog box, select the DAP template file, and click `OK`, which opens the `Properties` dialog box for the new DAP `Application` object.

3. On the `General` tab:
   *   Enter a descriptive name.
       A DAP can have the same name as the database itself. However, it is recommended that you make their names unique if you are using multiple access points for the same database.
   *   In the `DB Server` field, use the `Browse` button to select the `Application` object corresponding to the DB Server that you just installed.

   > **Note:** Do not select the `JDBC Connection` check box when you configure Database Access Points.

4. On the `DB Info` tab, specify information about the database in the following fields:

- `DBMS Name`—The name or alias identifying the DBMS that handles the database. The value of this option is communicated to DB Server so that it connects to the correct DBMS:
  - For DB2, set this value to the name or alias-name of the database specified in the db2 client configuration.
  - For Informix, this value is the name of SQL server, specified in the `sqlhosts` file.
  - For Microsoft SQL, set this value to the SQL server name (usually the same as the host name of the computer on which Microsoft SQL runs).
  - For Oracle, the value is the name of the Listener service.
  - For PostgreSQL, set this value to the SQL server name (usually the same as the host name of the computer on which PostgreSQL runs).
  - For Sybase, this value is the server name stored in the Sybase interface file.
- `DBMS Type`—The type of DBMS that handles the database. You must set a value for this property.
- `Database Name`—The name of the database to be accessed, as it is specified in the DBMS that handles this database. You must set a value for this property unless `oracle` or `db2` is specified as the `DBMS Type`. For Sybase, Informix, and Microsoft SQL, this value is the name of the database to which the client will connect.
- `User Name`—The user name established in the SQL server to access the database. You must set a value for this property.
- `Password`—The password established in the SQL server to access the database.
- `Re-enter Password`—Confirmation for the value entered for `Password`.
- `Case Conversion`—Case conversion method for key names of key-value lists coming from DB Server. This value specifies whether and how a client application converts the field names of a database table when receiving data from DB Server. If you select `upper`, field names are converted into uppercase; if you select `lower`, field names are converted into lowercase; and if you select `any`, field names are not converted. This setting does not affect the values of key-value lists coming from DB Server. That is, actual data is being presented exactly as in the database tables.

Note: For the Case Conversion option, use the default value (`any`) unless directed to do otherwise by Genesys Ewuqo gt'Ectg.

> **Note:** Do not configure any properties on the `JDBC Info` tab when configuring a DAP application for the Management Layer.

5. On the `Server Info` tab:
- Click the `Browse` button next to the `Host` drop-down list, and select the host on which this DB Server will run.
- Specify the listening port(s) and select whether each is secure or not secure. Refer to the *Genesys Security Deployment Guide* for detailed information on specifying ports and securing connections to them.

End of procedure

To interface an application with a database through a certain DAP, add this DAP to the `Application` object's `Connections` list.

Installing DB Server

This section describes how to install DB Server on UNIX and Windows operating systems after you've configured it using Genesys Administrator or Configuration Manager.

Procedure:
Installing Client DB Server on UNIX

Prerequisites

- An `Application` object exists for this DB Server. See "Configuring DB Server" on .

Start of procedure

1. On the Management Framework 8.1 product CD in the `services_layer/dbserver/[operating_system]` directory, locate a shell script called `install.sh`.

2. Run this script from the command prompt by typing the `sh` command and the file name. For example:
`sh install.sh`

3. When prompted, specify the `Host Name` of the computer on which DB Server is to be installed.

4. Type `n` when asked whether this DB Server will provide access to the Configuration Database.

5. When prompted, specify the:
- `Host Name` of the computer on which Configuration Server is running.

- • `Port` used by client applications to connect to Configuration Server.
- • `User Name` used to log in to the Configuration Layer.
- • `Password` used to log in to the Configuration Layer.

6. The installation displays the list of `Application` objects of `DB Server` type configured for this `Host`. Type the number of the DB Server Application that you want installed.

7. Specify the destination directory into which DB Server is to be installed, with the full path to it.

8. The installation displays the names of the DB client processes for different types of SQL servers. Type the number of the DB client process name that should be configured.

As soon as the installation process is finished, a message appears announcing that installation was successful.

End of procedure

Procedure:
Installing Client DB Server on Windows

Prerequisites

- • An `Application` object exists for this DB Server. See "Configuring DB Server" on .

Start of procedure

1. From the Management Framework 8.1 product CD, open the `services_layer\dbserver\windows` directory.

2. Locate and double-click `Setup.exe` to start installation.

3. When prompted, select the option of running DB Server as a Configuration Server client.

4. When prompted, specify the:
   - • `Host Name` of the computer on which Configuration Server is running.
   - • `Port` used by client applications to connect to Configuration Server.
   - • `User Name` used to log in to the Configuration Layer.
   - • `Password` used to log in to the Configuration Layer.

5. From the list of `Application` objects of type `DB Server` configured for this Host, select the DB Server Application to install.

6. Specify the destination directory in which you want DB Server installed.

7. When prompted, click `Finish` to complete the installation.

End of procedure

Starting DB Server

For complete instructions on starting and stopping DB Server, refer to the *Framework 8.1 Deployment Guide.*

Note that when you start the DB Server through the Management Layer, the server inherits environment variables from the Local Control Agent (LCA) component, which executes the startup command. That means you must also set the environment variables required for DB Server for the account that runs LCA. See Chapter 2, "DBMS Environment Settings," on for more information.

Warning! Always restart DB Server when restarting your DBMS server.

Genesys

2 DBMS Environment Settings

This chapter provides recommendations on setting the configuration of DBMS clients for supported types of databases. The information in this chapter is divided among the following topics:

To work with a DBMS client, DB Server requires particular environment settings. This chapter lists those settings for each database type.

In addition to the recommendations in this chapter, refer to the DBMS vendor's documentation.

General Recommendations

Genesys DB clients for Solaris, AIX, and Tru64 UNIX operating systems might require the environment variable `LD_LIBRARY_PATH` (for Solaris and Tru64 UNIX) or `LIBPATH` (on AIX) environment variable for startup. Set this variable and specify a path to the DBMS client libraries.

The 64-bit DB client for the 64-bit version of the Solaris operating system might require the `LD_LIBRARY_PATH_64` environment variable for startup. Set this variable and specify a path to the DBMS client libraries.

The DB client for the HP-UX operating system might require the `SHLIB_PATH` environment variable for startup. Set this variable and specify a path to the DBMS client libraries.

Additional recommendations for your database system are given throughout this chapter.

Genesys DB Client for Sybase

You must set the environment variables `SYBASE` and `DSQUERY` for DB client for Sybase.

When you create a call to the stored procedure in Interaction Router Designer, you can omit the `@` prefix normally required by Sybase when specifying parameter names.

For more information, refer to the Sybase documentation.

Table 1 lists additional configuration requirements specific to various operating systems; that is, a name and a value of an environment variable that you must define.

Table 1: Additional Configuration for DB Client for Sybase

Operating System	Environment Variable Name	Environment Variable Value
AIX	No additional requirements	
HP-UX	No additional requirements	
Tru64 UNIX	LD_LIBRARY_PATH	—
Solaris	LD_LIBRARY_PATH	—
Windows	PATH	%PATH%;%SYBASE%\DLL

If you are using IMAGE data on Sybase for reading or writing, Genesys also recommends the following:

- To optimize the performance of DB Server, limit the size of the IMAGE to 1 MB or less.

- Set the following configuration options to the specified values on your Sybase server:

```
abort tran on log full = true
trunc log on chkpt = true
```

Genesys DB Client for Oracle

You must set the following environment variables for DB client for Oracle:

    ORACLE_HOME

    ORACLE_SID

In addition, you must specify the full path to the `bin` of the Oracle home directory in the `PATH` variable.

If the DBMS client for Oracle runs on any host other than the Oracle server, you must configure the SQLNet on both hosts. Note that a TCP/IP Adapter is also required.

For more information, refer to the Oracle documentation.

Table 2 lists additional configuration requirements specific to various operating systems; that is, a name and a value of an environment variable that you must define.

Table 2: Additional Configuration for DB Client for Oracle

Operating System		Environment Variable Name	Environment Variable Value
AIX	DB client	LIBPATH	Oracle DBMS library directory
HP-UX	32-bit version of DB client 8i	SHLIB_PATH	$ORACLE_HOME/lib
	32-bit version of DB client 9i	SHLIB_PATH	$ORACLE_HOME/lib32
	64-bit version of DB client 8i	SHLIB_PATH	$ORACLE_HOME/lib64
	64-bit version of DB client 9i	SHLIB_PATH	$ORACLE_HOME/lib
Tru64 UNIX	DB client	LD_LIBRARY_PATH	$ORACLE_HOME/lib
Solaris	32-bit version of DB client 8i	LD_LIBRARY_PATH	$ORACLE_HOME/lib
	32-bit version of DB client 9i	LD_LIBRARY_PATH	$ORACLE_HOME/lib32
	64-bit version of DB client 8i	LD_LIBRARY_PATH_64	$ORACLE_HOME/lib64
	64-bit version of DB client 9i	LD_LIBRARY_PATH_64	$ORACLE_HOME/lib

Table 2: Additional Configuration for DB Client for Oracle (Continued)

Operating System		Environment Variable Name	Environment Variable Value
Linux	DB Client	LD_LIBRARY_PATH	$ORACLE_HOME/lib

Genesys DB Client for MS SQL

When you create a call to the stored procedure in Interaction Router Designer, you can now omit the @ prefix normally required by MS SQL when specifying parameter names.

For more information, refer to the MS SQL documentation.

No additional configuration is required for the DB client for MS SQL.

Genesys DB Client for Informix

For DB client for Informix you must set the following environment variables with the full path to the Informix home directory and with the name of the Informix server:

    INFORMIXDIR

    INFORMIXSERVER

For Informix version 10.0, you must install Informix Client SDK and set the INFORMIXDIR environment variable with the full path to the Client SDK home directory.

For more information refer to Informix's *Online Dynamic Server User's Guide.*

Table 3 on lists additional configuration requirements specific to various operating systems: that is, a name and a value of an environment variable that you must define.
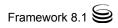
Table 3: Additional Configuration for DB Client for Informix

Operating System	Environment Variable Name	Environment Variable Value
AIX	No additional requirements	
HP-UX	SHLIB_PATH	$INFOMIXDIR/lib:$INFORMIXDIR/lib/esql
Tru64 UNIX	No additional requirements	
Solaris	No additional requirements	

Table 3: Additional Configuration for DB Client for Informix (Continued)

Operating System	Environment Variable Name	Environment Variable Value
Windows	PATH	%INFORMIXDIR%\BIN

Genesys DB Client for DB2

You must set the `DB2INSTANCE` and `INSTHOME` variables for DB client for DB2. In addition, you must specify the full path to the `bin`, `adm`, and `misc` of the DB2 home directory in the `PATH` variable. For more information, refer to DB2 documentation.

Table 4 lists additional configuration requirements specific to various operating systems: that is, a name and a value of an environment variable that you must define.

Table 4: Additional Configuration for DB Client for DB2

Operating System	Environment Variable Name	Environment Variable Value
AIX	LIBPATH	DB2 DBMS library directory
HP-UX	SHLIB_PATH	
Linux	LD_LIBRARY_PATH	—
Solaris	LD_LIBRARY_PATH	

Genesys DB Client for PostgreSQL

Table 5 lists additional configuration requirements specific to various operating systems: that is, a name and a value of an environment variable that you must define.

Table 5: Additional Configuration for DB Client for PostgreSQL

Operating System	Environment Variable Name	Environment Variable Value
Linux	LD_LIBRARY_PATH	<PostgreSQL folder>/lib:/lib:/usr/lib:/usr/local/lib
Windows	PATH	<PostgreSQL folder>\bin

![Genesys]

3

DB Server Configuration Options

This chapter describes configuration options and a configuration file for DB Server.

This chapter contains the following sections:

DB Server also supports the options described in Chapter 5 on page 45.

Setting Configuration Options

Unless specified otherwise, set DB Server configuration options in the options of the DB Server `Application` object, using one of the following navigation paths:

- In Genesys Administrator—DB Server `Application` object > `Options` tab > `Advanced View (Options)`
- In Configuration Manager—DB Server `Application` object > `Properties` dialog box > `Options` tab

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Genesys Administrator or Configuration Manager exactly as they are documented in this chapter.

Mandatory Options

Table 6 lists the DB Server options for which you must provide values; otherwise, DB Server will not start. The options are listed by section.

Table 6: Mandatory Options

Option Name	Default Value	Details
DB Server Section		
dbprocess_name	No default value	See the description on page 35.
[a DB client process name option]	No default value	The option name depends on the DBMS type: db2_name, informix_name, msql_name, oracle_name, postgre_name, or sybase_name. See the descriptions beginning on page 35.

dbserver Section

This section must be called dbserver.

client_stop_timeout

Default Value: 30
Valid Values: Any positive integer from 0–65535
Changes Take Effect: After restart

Specifies the interval, in seconds, that DB Server waits for a client to stop before DB Server terminates the DB client process.

connect_break_time

Default Value: 1200
Valid Values: Any positive integer
Changes Take Effect: After restart

Specifies a timeout, in seconds, after which DB Server closes a connection to a DB client if DB Server could not send a request to the client. Do not set this option too small; if a value of 1 to 10 seconds is set, for example, network delay might prevent a request delivery. Genesys recommends that you set this option to a value equal to or greater than 60.

db-request-timeout

Default Value: 0
Valid Values: 0–604800 (in seconds, equivalent to 0 seconds–7 days)

Changes Take Effect: After DB Server reconnects to the database; no restart is required.

Specifies the period of time, in seconds, that it should take one DBMS request to be completed. If a request to the DBMS takes longer than this period of time, the database client process stops executing, and DB Server interprets this as a DBMS failure.

DB Server uses this option for all started database client processes, unless overwritten by the value of the `db-request-timeout` option in the annex of a Database Access Point (DAP) object.

If this option is set to the default value of `0` (zero), no timeout is used.

Note: This option applies only to DB Servers that provide access to databases other than the Configuration Database. In other words, do not use this option for the Configuration DB Server.

---

db2_name

Default Value: `./dbclient_db2`
Valid Values:

`./dbclient_db2`	Strongly recommended.
`./dbclient_db2_32`	Use this value only if it is clearly indicated that you use the 32-bit DB Server client.
`./dbclient_db2_64`	Use this value only if it is clearly indicated that you use the 64-bit DB Server client.

Changes Take Effect: After restart

Specifies the name of the DB Server client process for the DB2 server. **This option is required for DB2 databases.** Also see `dbprocess_name`.

dbprocess_name

Default Value: No default value
Valid Values: Use one of the following, based on your DBMS.

`./dbclient_db2`	For DB2
`./dbclient_informix`	For Informix
`./dbclient_msql`	For Microsoft SQL
`./dbclient_oracle`	For Oracle
`./dbclient_postgre`	For PostgreSQL
`./dbclient_sybase`	For Sybase

Changes Take Effect: After restart

Specifies the type of DB client process, based on the DBMS being used. This option works with `dbprocesses_per_client` and the corresponding `<DBMS>_name` option (that is, `db2_name`, `informix_name`, `msql_name`, `oracle_name`, `postgre_name`, or `sybase_name`).

> **Note:** Enable this option only for compatibility with previous releases of client applications (5.1, 6.0, or 6.1).

dbprocess_number

Default Value: `255`
Valid Values:

`0`	Does not restrict the number of running DB Client processes
`1` and above	Sets the maximum number of simultaneously running DB Client processes

Changes Take Effect: After restart

Sets the maximum limit for the number of simultaneously running DB Client processes.

dbprocesses_per_client

Default Value: `1`
Valid Values: Any positive integer from `1`–`255`
Changes Take Effect: After restart

Specifies the number of database client processes that DB Server's main process creates for each client if a user client does not make an explicit request. This option prioritizes client access to the database. For example, if multiple processes per client are set, DB Server spawns another child process if needed. This effectively gives the client application more of the database's processing time. See documentation for a particular client application to verify whether that application supports the Multiple Processes mode. If unsure of the appropriate number, set this option to `1`. Increasing the value up to `4` increases performance; more than `4` does not increase performance.

> **Note:** Genesys recommends using the default value (`1`) for this option unless instructed otherwise by Eww•qo gt'Ectg'or by the user's guide of the applicable Genesys solution. Changing the default value (`1`) of this option may cause data loss.

informix_name

Default Value: `./dbclient_informix`
Valid Values: `./dbclient_informix`
Changes Take Effect: After restart

Specifies the name of the DB Server client process for the Informix server if present. **This option is required for Informix databases.** Also see `dbprocess_name`.

management-port

Default Value: `4051`
Valid Values: Any valid TCP/IP port

Changes Take Effect: After restart

Specifies the TCP/IP port DB Server reserves for connections established by its SNMP (Simple Network Management Protocol) Option Management Client.

msql_name

Default Value: `./dbclient_msql`
Valid Values: `./dbclient_msql`
Changes Take Effect: After restart

Specifies the name of the DB Server client process for the Microsoft SQL server. **This option is required for MSSQL databases.** Also see `dbprocess_name`.

oracle_name

Default Value: `./dbclient_oracle`
Valid Values:

`./dbclient_oracle`	Strongly recommended.
`./dbclient_oracle_32`	Use this value only if it is clearly indicated that you use the 32-bit DB Server client.
`./dbclient_oracle_64`	Use this value only if it is clearly indicated that you use the 64-bit DB Server client.

Changes Take Effect: After restart

Specifies the name of the DB Server client process for the Oracle server if present. **This option is required for Oracle databases.** Also see `dbprocess_name`.

postgre_name

Default Value: `./dbclient_postgre`
Valid Values: `./dbclient_postgre`
Changes Take Effect: After restart

Specifies the name of the DB Server client process for the PostgreSQL server if present. **This option is required for PostgreSQL databases.** Also see `dbprocess_name`.

stored_proc_result_table

Default Value: No default value
Valid Values: Any valid table name
Changes Take Effect: After restart

Used by earlier versions of DB Server that do not directly retrieve output data from stored procedures. This option specifies the name of a table that you design, to which a stored procedure that you have created writes output data (the maximum allowed size of an output parameter from a stored procedure is 2000 B). DB Server then retrieves the data stored in the specified table and

sends it to the user application. Using a result table can slow down DB Server, because each stored procedure call causes an additional select statement.

sybase_name

Default Value: `./dbclient_sybase`
Valid Values: `./dbclient_sybase`
Changes Take Effect: After restart

Specifies the name of the DB Server client process for the Sybase server if present. **This option is required for Sybase databases.** Also see `dbprocess_name`.

tran_batch_mode

Default Value: `off`
Valid Values: `on`, `off`
Changes Take Effect: After restart

Valid only for Sybase databases. If set to `on`, DB Server executes all transactions as SQL batches, which increases performance for insert and update statements.

Note: Genesys recommends using the default value (`off`) for this option unless instructed otherwise by Ewⱳqo gt'Ectg or by the user's guide of the applicable Genesys solution.

verbose

Default Value: `3`
Valid Values:

`0`	DB Server writes no debug messages.
`1`	DB Server writes errors and SQL statements.
`2`	DB Server writes information about all messages it has received and sent.
`3`	DB Server writes debug messages at the most detailed level.

Changes Take Effect: After restart

Sets the level of detail with which DB Server writes the debug messages. The option is configured in the `dbserver` section and is enabled only when the `verbose` option in the `log` section is set to either `all` or `debug`. DB Server writes the debug messages to a log output specified for the `all` and/or `debug` log output options.

Note: Although named the same, the `verbose` options in the `log` and `dbserver` sections are responsible for different types of log settings.

Transport Parameter Options

Set options in this section in the `Transport Parameters` of the port's properties, using one of the following navigation paths:

- In Genesys Administrator—DB Server `Application` object > `Configuration` tab > `General` section > `Connections` ⟩ `Connection Info` > `Advanced` tab > `Transport Parameters`

- In Configuration Manager—DB Server `Application` object > `Properties` dialog box > `Connections` tab > `Connection Properties` dialog box > `Advanced` tab > `Transport Protocol Parameters`

Transport Parameter options are not associated with a configuration option section, and do not appear in the options or annex of a DB Server `Application` object.

transport Option

In a configuration file, these options appear in the following format:

`transport = <option name>=<value>;<option name>=<value>; ...`

Collectively, the options make up the parameters of the `transport` option. When entering the options in Genesys Administrator or Configuration Manager, only the options are required; `transport =` is prefixed automatically to the list of option/value pairs.

> **Note:** Valid values for these options must have no spaces before or after the delimiter characters ";" (semi-colin) and "=".

tls

Default Value: tls=`0`
Valid Values:

`tls=0`	Regular (unsecured) connections will be used.
For Windows: tls=1; certificate=<value>	Secure connections will be used, where `certificate` = certificate value.
For UNIX: tls=1; certificate=<path>; [certificate-key=<path>]; trusted-ca=<path>	Secure connections will be used, where: • `certificate`—full path to the `<serial_#>_<host_name>_cert.pem` file • `certificate-key`—full path to the `<serial_#>_<host_name>_priv_key.pem` file (unless the private key is stored together with a certificate) • `trusted-ca`—full path to the `ca_cert.pem` file

You specify the `transport` option in any section of the DB Server configuration file that contains port configuration, and in the Configuration Database section of the Configuration Server configuration file. Refer to the chapter "Configuring Secure Configuration Server and DB Server

Connections" in the *Genesys Security Deployment Guide* for information about how to use this option.

Changes from 8.0 to 8.1

Table 7 lists all changes to DB Server options between release 8.0 and the latest 8.1 release.

Table 7: DB Server Configuration Option Changes from 8.0 to 8.1

Option Name	Option Values	Type of Change	Details
dbserver Section			
tran_batch_mode	on, off	Changed	No longer valid for MS SQL databases. Change occurred in release 7.5 and not documented. See description on page 38.
Transport Parameters			
tls	0, 1	New in 7.5	See description on page 39. Previously documented in the *Genesys 8.1 Security Deployment Guide* as the transport option.

4

Database Access Point Configuration Options

This chapter describes configuration options for a Database Access Point.

This chapter contains the following sections:

Setting Configuration Options

Refer to the description of the particular option for information about where to set its value. Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Genesys Administrator or Configuration Manager exactly as they are documented in this chapter.

Mandatory Options

You do not have to configure any options for a Database Access Point.

default Section

This section must be called `default`.

db-request-timeout

Default Value: `0`
Valid Values: `0–604800` (in seconds, equivalent to 0 seconds–7 days)
Changes Take Effect: After DB Server reconnects to the database; no restart is required.

Specifies the period of time, in seconds, that it should take one DBMS request to be completed. If a request to the DBMS takes longer than this period of time, the database client process stops executing, and DB Server interprets this as a DBMS failure.

This option overrides the value of the `db-request-timeout` option specified by DB Server, for this Database Access Point only.

If this option is set to the default value of `0` (zero), the value of the `db-request-timeout` option configured in DB Server is used.

Setting this Option

You can configure this option in any of the following locations:

- In Genesys Administrator
  - Database Access Point `Application` object > `Options` tab > `Advanced View (Annex)`
  - Database Access Point `Application` object > `Configuration` tab > `DB Info` section > `Query Timeout` field
- In Configuration Manager
  - Database Access Point `Application` object > `Properties` dialog box > `DB Info` tab > `Query Timeout` field

dbclient Section

This section must be called `dbclient`.

utf8-ucs2

Default Value: false
Valid Values: true, false
Changes Take Effect: At startup

This option applies only if you are working with an MS SQL Log Database that has been initialized as a multi-language database. MS SQL uses UCS-2 encoding instead of UTF-8. Setting this option to `true` forces the transcoding of UTF-8 to UCS-2 encoding before writing to the MS SQL database, and the transcoding of UCS-2 to UTF-8 encoding after reading from the database. Therefore, the MS SQL database is able to work with other components encoded using UTF-8.

Setting this Option

Set this option only in Genesys Administrator, as the following location:

- Database Access Point `Application` object > `Options` tab > `Advanced View (Annex)`

You cannot configure this option in Configuration Manager. Multi-language functionality is available only in Genesys Administrator.

Changes from 8.0 to 8.1

Table 8 lists all changes to Database Access Point options between release 8.0 and the latest 8.1 release.

Table 8: Database Access Point Option Changes from 8.0 to 8.1

Option Name	Option Values	Type of Change	Details
dbclient Section (new section)			
utf8-ucs2	`true`, `false`	New	See description on page 42.

Genesys

5

Common Configuration Options

Unless otherwise noted, the common configuration options that this chapter describes are common to all Framework server components. They may also be used by other Genesys server applications; refer to product-specific documentation to determine if these options apply to your product.

This chapter includes the following sections:

Note: Some server applications also support log options that are unique to them. For descriptions of a particular application's unique log options, refer to the chapter/document about that application.

Setting Configuration Options

Unless specified otherwise, set common configuration options in the options of the `Application` object, using one of the following navigation paths:

- In Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Options)`

- In Configuration Manager—`Application` object > `Properties` dialog box > `Options` tab

Warning! Configuration section names, configuration option names, and predefined option values are case-sensitive. Type them in Genesys Administrator or Configuration Manager exactly as they are documented in this chapter.

Mandatory Options

You do not have to configure any common options to start Server applications.

Common Log Options

This section contains all options relating to creating, viewing, and otherwise using the Centralized Log facility in Genesys software.

log Section

This section must be called `log`.

Warning! For applications configured via a configuration file, changes to log options take effect after the application is restarted.

buffering

Default Value: `true`
Valid Values:

`true`   Enables buffering.
`false`   Disables buffering.

Changes Take Effect: Immediately

Turns on/off operating system file buffering. The option is applicable only to the `stderr` and `stdout` output (see page 52). Setting this option to `true` increases the output performance.

Note: When buffering is enabled, there might be a delay before log messages appear at the console.

check-point

Default Value: `1`
Valid Values: `0–24`
Changes Take Effect: Immediately

Specifies, in hours, how often the application generates a `check point` log event, to divide the log into sections of equal time. By default, the application generates this log event every hour. Setting the option to `0` prevents the generation of `check-point` events.

compatible-output-priority

Default Value: `false`
Valid Values:

`true`	The log of the level specified by "Log Output Options" is sent to the specified output.
`false`	The log of the level specified by "Log Output Options" and higher levels is sent to the specified output.

Changes Take Effect: Immediately

Specifies whether the application uses 6.x output logic. 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 log file content of a 6.x application is as follows:

- `file1` contains Debug messages only.
- `file2` contains Standard messages only.

The log file content of a 7.x application 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, its log file content will be the same as for the 6.x application.

> **Warning!** Genesys does not recommend changing the default value of this option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to `true` affects log consistency.

expire

Default Value: `false`
Valid Values:

`false`	No expiration; all generated segments are stored.
`<number> file` or `<number>`	Sets the maximum number of log files to store. Specify a number from 1–`1000`.
`<number> day`	Sets the maximum number of days before log files are deleted. Specify a number from 1–`100`.

Changes Take Effect: Immediately

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

Note: If an option's value is set incorrectly—out of the range of valid values— it will be automatically reset to `10`.

keep-startup-file

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

Specifies whether a startup segment of the log, containing the initial configuration options, is to be kept. If it is, this option can be set to `true` or to a specific size. If set to `true`, the size of the initial segment will be equal to the size of the regular log segment defined by the `segment` option. The value of this option will be ignored if segmentation is turned off (that is, if the `segment` option is set to `false`).

memory

Default Value: No default value
Valid Values: `<string>` (memory file name)
Changes Take Effect: Immediately

Specifies the name of the file to which the application regularly prints a snapshot of the memory output, if it is configured to do this (see "Log Output Options" on). The new snapshot overwrites the previously written data. If the application terminates abnormally, this file will contain 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, an application does not create a snapshot file (with the extension `*.memory.Log`). Log output to a file at a network location is not recommended and could cause performance degradation.

memory-storage-size

Default Value: `2 MB`

Valid Values:

`<number>KB` or `<number>`	The size of the memory output, in kilobytes. The minimum value is `128 KB`.
`<number> MB`	The size of the memory output, in megabytes. The maximum value is `64 MB`.

Changes Take Effect: When memory output is created

Specifies the buffer size for log output to the memory, if configured. See also "Log Output Options" on .

message_format

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

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.

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), whereas 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.

*   The message ID does not contain the prefix `GCTI` or the application type `ID`.

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
```

A log record in the short format looks like this:

```
2002-05-07T18:15:33.952 Std 05060 Application started
```

Note: Whether the full or short format is used, time is printed in the format specified by the `time_format` option.

messagefile

Default Value: As specified by a particular application
Valid Values: `<string>.lms` (message file name)
Changes Take Effect: Immediately, if an application cannot find its `*.lms` file at startup

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, an application looks for the file in its working directory.

Warning! An application that does not find its `*.lms` file at startup cannot generate application-specific log events and send them to Message Server.

---

print-attributes

Default Value: `false`
Valid Values:

`true`	Attaches extended attributes, if any exist, to a log event sent to log output.
`false`	Does not attach extended attributes to a log event sent to log output.

Changes Take Effect: Immediately

Specifies whether the application attaches extended attributes, if any exist, to a log event that it sends to log output. Typically, log events of the Interaction log level and Audit-related log events contain extended attributes. Setting this option to `true` enables audit capabilities, but negatively affects performance. Genesys recommends enabling this option for Solution Control Server and Configuration Server when using audit tracking. For other applications, refer to *Genesys Combined Log Events Help* to find out whether an application generates Interaction-level and Audit-related log events; if it does, enable the option only when testing new interaction scenarios.

segment

Default Value: `false`
Valid Values:

`false`	No segmentation is allowed.
`<number> KB` or `<number>`	Sets the maximum segment size, in kilobytes. The minimum segment size is `100 KB`.
`<number> MB`	Sets the maximum segment size, in megabytes.
`<number> hr`	Sets the number of hours for the segment to stay open. The minimum number is 1 hour.

Changes Take Effect: Immediately

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. This option is ignored if log output is not configured to be sent to a log file.

spool

Default Value: The application's working directory

Valid Values: `<path>` (the folder, with the full path to it)
Changes Take Effect: Immediately

Specifies the folder, including full path to it, in which an application creates temporary files related to network log output. If you change the option value while the application is running, the change does not affect the currently open network output.

time_convert

Default Value: `Local`
Valid Values:

`local`	The time of log record generation is 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`	The time of log record generation is expressed as Coordinated Universal Time (UTC).

Changes Take Effect: Immediately

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).

time_format

Default Value: `time`
Valid Values:

`time`	The time string is formatted according to the `HH:MM:SS.sss` (hours, minutes, seconds, and milliseconds) format.
`locale`	The time string is formatted according to the system's locale.
`ISO8601`	The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds.

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when an application generates log records.

A log record's time field in the ISO 8601 format looks like this:
`2001-07-24T04:58:10.123`

verbose

Default Value: `all`
Valid Values:

`all`	All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
`debug`	The same as `all`.
`trace`	Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.

interaction   Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated.

standard   Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.

none   No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug. See also "Log Output Options" on page 52.

> **Note:** For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework Management Layer User's Guide*, *Framework Genesys Administrator Help,* or to *Framework Solution Control Interface Help*.

Log Output Options

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

You can use:

- One log level option to specify different log outputs.

- One log output type for different log levels.

- Several log output types simultaneously, to log 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 56.

> **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) in case it terminates abnormally.
>
> • Directing log output to the console (by using the stdout or stderr settings) can affect application performance. Avoid using these log output settings in a production environment.

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

all

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.

Setting the `all` log level option to the `network` output enables an application to send log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. `Debug`-level log events 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 is the safest output in terms of the application performance.

[filename]   Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example:

```
all = stdout, logfile
```

Note: To ease the troubleshooting process, consider using unique names for log files that different applications generate.

alarm

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 resides anywhere on the network, and 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 a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Alarm level. The log output types must be separated by a comma when more than one output is configured. For example:

```
alarm = stderr, network
```

standard

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 a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

interaction

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 a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels). The log outputs must be separated by a comma when more than one output is configured. For example:

```
interaction = stderr, network
```

trace

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 a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Trace` level and higher (that is, log events of the `Standard`, `Interaction`, and `Trace` levels). The log outputs must be separated by a comma when more than one output is configured. For example:

`trace = stderr, network`

debug

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 is the safest output in terms of the application performance.
`[filename]`	Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Debug` level and higher (that is, log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels). The log output types must be separated by a comma when more than one output is configured—for example:

`debug = stderr, /usr/local/genesys/logfile`

Note: `Debug`-level log events are never sent to Message Server or stored in the Log Database.

Log File Extensions

You can use the following file extensions to identify log files that an application 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 = confservlog` for Configuration Server, it prints log messages into a text file called `confservlog.<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` for Configuration Server, it prints log messages into a file called `confserv.<time_stamp>.qsp` during the time the network is not available.

- `*.snapshot.log`—Assigned to files that contain the output snapshot when you configure output to a log file. The file contains the last log messages that an application generates before it terminates abnormally. For example, if you set `standard = confservlog` for Configuration Server, it prints the last log message into a file called `confserv.<time_stamp>.snapshot.log` in case of failure.

> **Note:** Provide `*.snapshot.log` files to Genesys Ewuvqo gt'Ectg 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 = confserv` for Configuration Server, it prints the latest memory output to a file called `confserv.<time_stamp>.memory.log`.

Examples

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

Production Mode Log Section

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

With this configuration, an application only generates the log events of the `Standard` level and sends them to Message Server, and to a file named `logfile`, which the application 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/logfile
trace = network
```

With this configuration, an application generates log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels, and sends them to the standard output and to a file named `logfile`, which the application creates in the `/usr/local/genesys/` directory. In addition, the application sends log events of the `Standard`, `Interaction`, and `Trace` levels to Message Server. Use this configuration to test new interaction scenarios in a lab environment.

Failure-Troubleshooting Log Section

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

With this configuration, an application 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 `logfile`, which the application creates in its working directory. Increased memory storage allows an application to save more of the log information generated before a failure...

Note: If you are running an application on UNIX, and you do not specify any files in which to store the memory output snapshot, a core file that the application produces before terminating contains the most current application log. Provide the application's core file to Genesys Ewuxqo gt" Ectg'when reporting a problem.

Debug Log Options

The options in this section enable you to generate `Debug` logs containing information about specific operations of an application.

x-conn-debug-all

Default Value: `0`
Valid Values:

`0`   Log records are not generated.
`1`   Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about open connection, socket select, timer creation and deletion, write, security-related, and DNS operations, and connection library function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<op type>` options.

Warning! Use this option only when requested by Genesys Ewuxqo gt Ectg.

x-conn-debug-api

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
1	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about connection library function calls.

> **Warning!** Use this option only when requested by Genesys Ewuvqo gt
> Ectg.

x-conn-debug-dns

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
1	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about DNS operations.

> **Warning!** Use this option only when requested by Genesys Ewuvqo gt
> Ectg.

x-conn-debug-open

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
1	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "open connection" operations of the
application.

> **Warning!** Use this option only when requested by Genesys Ewuvqo gt
> Ectg.

x-conn-debug-security

Default Value: `0`
Valid Values:

`0`	Log records are not generated.
1	Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about security-related operations, such as
Transport Layer Security and security certificates.

Warning! Use this option only when requested by Genesys Ewuxqo gt
Ectg.

x-conn-debug-select

Default Value: `0`
Valid Values:

`0`   Log records are not generated.
1   Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "socket select" operations of the
application.

Warning! Use this option only when requested by Genesys Ewuxqo gt
Ectg.

x-conn-debug-timers

Default Value: `0`
Valid Values:

`0`   Log records are not generated.
1   Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about the timer creation and deletion operations
of the application.

Warning! Use this option only when requested by Genesys Ewuxqo gt
Ectg.

x-conn-debug-write

Default Value: `0`
Valid Values:

`0`   Log records are not generated.
1   Log records are generated.

Changes Take Effect: After restart

Generates `Debug` log records about "write" operations of the application.

Warning! Use this option only when requested by Genesys Ewuxqo gt
Care.

log-extended Section

This section must be called `log-extended`.

level-reassign-disable

Default Value: `false`
Valid Values: `true, false`
Changes Take Effect: Immediately

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, but not delete the customization statements.

level-reassign-<eventID>

Default Value: Default value of log event `<eventID>`
Valid Values:

`alarm`	The log level of log event `<eventID>` is set to `Alarm`.
`standard`	The log level of log event `<eventID>` is set to `Standard`.
`interaction`	The log level of log event `<eventID>` is set to `Interaction`.
`trace`	The log level of log event `<eventID>` is set to `Trace`.
`debug`	The log level of log event `<eventID>` is set to `Debug`.
`none`	Log event `<eventID>` is not recorded in a log.

Changes Take Effect: Immediately

Specifies a log level for log event `<eventID>` that is different than its default level, or disables log event `<eventID>` completely. If no value is specified, the log event retains its default level. This option is useful when you want to customize the log level for selected log events.

These options can be deactivated with option `level-reassign-disable` option.

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

Depending on the log configuration, changing the log level to a higher priority may 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 may cause the log event to be not logged at all, or to be not logged to specific outputs, thereby losing important information. The same applies to any alarms associated with that log event.

In addition to the preceding warning, take note of the following:

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

• 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 set to `none`, it is not logged and is therefore not subject to any log configuration.

• Using this feature to change the log level of a log 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 a High Availability (HA) pair can define its own unique set of log customizations, 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 (URS) release 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 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 using customized log level settings, subject to the following log configuration:

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

Before the log levels of the log are changed:

- Log event 1020, with default level `standard,` is output to `stderr` and `log_file,` and sent to Message Server.

- Log event 2020, with default level `standard,` is output to `stderr` and `log_file,` and sent to Message Server.

- Log event 3020, with default level `trace,` is output to `stderr`.

- Log event 4020, with default level `debug,` is output to `stderr`.

Extended log configuration section:

```
[log-extended]
level-reassign-1020=none
level-reassign-2020=interaction
level-reassign-3020=interaction
level-reassign-4020=standard
```

After the log levels are changed:

- Log event 1020 is disabled and not logged.

- Log event 2020 is output to `stderr` and `log_file`.

- Log event 3020 is output to `stderr` and `log_file`.

- Log event 4020 is output to `stderr` and `log_file`, and sent to Message Server.

Common Security Options

Common security options are used to implement some security features in Genesys software. These options are configured on supporting `Application` objects.

Filtering and/or Tagging Data in Logs

log-filter Section

The `log-filter` section contains configuration options used to define the default treatment of filtering data in log output. Specifically, it defines the treatment of all KV pairs in the `AttributeUserData` section of the log.

This section must be called `log-filter`.

default-filter-type

Default Value: `copy`
Valid Values: One of the following:

`copy`	The keys and values of the KVList pairs in the `AttributeUserData` section are copied to the log.
`hide`	The keys of the KVList pairs in the `AttributeUserData` section are copied to the log; the values are replaced with asterisks.
`hide-first,<n>`	The keys of the KVList pairs in the `AttributeUserData` section are copied to the log; the first <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value.
`hide-last,<n>`	The keys of the KVList pairs in the `AttributeUserData` section are copied to the log; the last <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value.
`skip`	The KVList pairs in the `AttributeUserData` section are not copied to the log.

tag[(<tag-prefix>, <tag-postfix>)]	The KVList pairs in the `AttributeUserData` section are tagged with the prefix specified by <tag-prefix> and the postfix specified by <tag-postfix>. If the two parameters are not specified, the default tags <# and #> are used as prefix and postfix, respectively.
	To use the default tags, you can use any of the following values:
	• `tag`
	• `tag()`
	• `tag(,)`
	To define your own tags, replace the two parameters in the value with your tags. Your own tag can be any string up to 16 characters in length; any string longer than that will be truncated. If the string is includes a blank space or any of the characters , (comma), (, or) as start and stop characters, they will not be counted as part of the length of the string.
unhide-first,<n>	The keys of the KVList pairs in the `AttributeUserData` section are copied to the log; all but the first <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the value of the key appears, with no asterisks.
unhide-last,<n>	The keys of the KVList pairs in the `AttributeUserData` section are copied to the log; all but the last <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the key, the value of the key appears, with no asterisks.

Changes Take Effect: Immediately

Specifies the default way of presenting KVList information (including `UserData`, `Extensions`, and `Reasons`) in the log. This setting will be applied to the attributes of all KVList pairs in the `AttributeUserData` section except those that are explicitly defined in the `log-filter-data` section.

Refer to the "Hide Selected Data in Logs" chapter in the *Genesys Security Deployment Guide* for information about how to use this option.

filtering

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately, if application is subscribed to notifications that this option has been changed.

Enables (`true`) or disables (`false`) log filtering at the Application level.

log-filter-data Section

The `log-filter-data` section defines the treatment of specific KV pairs in the `AttributeUserData` section of the log. It overrides the general settings in the `log-filter` section.

This section must be called `log-filter-data`.

<key-name>

Default Value: No default value
Valid Values: One of the following:

`copy`	The key and value of the given KVList pair in the `AttributeUserData` section is copied to the log.
`hide`	The key of the given KVList pair in the `AttributeUserData` section is copied to the log; the value is replaced with a string of asterisks.
`hide-first,<n>`	The key of the given KVList pair in the `AttributeUserData` section is copied to the log; the first <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value.
`hide-last,<n>`	The key of the given KVList pair in the `AttributeUserData` section is copied to the log; the last <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the number of asterisks will be equal to the number of characters in the value.
`skip`	The KVList pair in the `AttributeUserData` section is not copied to the log.
`tag[(<tag-prefix>, <tag-postfix>)]`	The KVList pair in the `AttributeUserData` section is tagged with the prefix specified by <tag-prefix> and the postfix specified by <tag-postfix>. If the two parameters are not specified, the default tags `<#` and `#>` are used as prefix and postfix, respectively.

To use the default tags, you can use any of the following values:

· `tag`
· `tag()`
· `tag(,)`

To define your own tags, replace the two parameters in the value with your tags. Your own tag can be any string up to 16 characters in length, and cannot include a blank space or any of the characters `,` (comma), `(`, or `)`. If the string is longer than 16 characters, it will be truncated.

`unhide-first,<n>`	The key of the given KVList pair in the AttributeUserData section is copied to the log; all but the first <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the value of the key appears, with no asterisks.
`unhide-last,<n>`	The key of the given KVList pair in the AttributeUserData section is copied to the log; all but the last <n> characters of the value are replaced with asterisks. If <n> exceeds the number of characters in the value, the value of the key appears, with no asterisks.

Changes Take Effect: Immediately

Specifies the way of presenting the KVList pair defined by the key name in the log. This setting supersedes the default way of KVList presentation as defined in the `log-filter` section for the given KVList pair.

If no value is specified for this option, no additional processing of this data element is performed.

Refer to the chapter "Hide Selected Data in Logs" in the *Genesys Security Deployment Guide* for complete information about how to use this option.

TLS Options

security Section

The `security` section contains configuration options used to specify security elements for your system. This section must be called `security`.

client-auth

Default Value: `1`
Valid Values: `0`, `1`
Changes Take Effect: After application restart

Specifies whether authentication of the security certificate in the client TLS socket is to be disabled. When set to `1` (default), authentication is enabled. When set to `0`, the client socket does not authenticate the server when connected over TLS.

This option must be set at the same level where the certificate itself is configured. Use this option if the security certificate is configured at the application level. If it is configured at another level, do one of the following:

• If the security certificate is configured at the connection level, see page 72.

• If the security certificate is configured at the host level, set this option at the host level.

Note: If this option is configured at multiple levels (connection, application, host), the value set at the lowest level takes precedence. That is:

- The value set at the connection level takes precedence over the value set at the application and host levels.
- The value set at the application level takes precedence over the value set at the host level.

tls-target-name-check

Default Value: `no`
Valid Values: `no`, `host`
Changes Take Effect: After component restart

Specifies whether the subject field in the server's certificate will be compared to the target host name (option value `host`). If they are not identical, the connection fails. If the option is set to `no`, such a comparison is not made, and the connection is allowed.

Secure User Authentication

security-authentication-rules Section

The `security-authentication-rules` section contains configuration options that relate to user accounts and user passwords. Refer to the chapter "User Passwords" in the *Genesys Security Deployment Guide* for full information about how to use these options.

This section must be called `security-authentication-rules`.

no-change-password-at-first-login

Default Value: `false`
Valid Values: `false`, `true`
Changes Take Effect: At the next attempt to log in to this application

Specifies whether this application supports password change when a user first logs in. If set to `true`, this application can override of the policy of changing passwords at first login. If set to `false` (the default), this application supports password change at first login.

This option does not apply if the `force-password-reset` option is set to `true` at the Tenant level, enforcing the current policy of changing passwords at first login.

Note: This option is set in the options of the `Application` object.

sml Section

This section must be called `sml`.

Options in this section are defined in the annex of the `Application` object, as follows:

- in Genesys Administrator—`Application` object > `Options` tab > `Advanced View (Annex)`

- in Configuration Manager—`Application` object > `Properties` dialog box > `Annex` tab

---

Warning!   Use the options in this section with great care, and only with those applications for which support for this functionality has been announced. Failure to use these options properly could result in unexpected behavior, from ignoring the options to an unexpected restart of the application.

---

hangup-restart

Default Value: `true`
Valid Values: `true`, `false`
Changes Take Effect: Immediately

If set to `true` (the default), specifies that LCA is to restart the unresponsive application immediately without any user intervention.

If set to `false`, specifies that LCA is only to generate a notification that the application has stopped responding; the application is not automatically restarted.

Note: This option is set to `true` automatically in Solution Control Server; any other value is ignored.

---

heartbeat-period

Default Value: None
Valid Values:

`0`	This method of detecting an unresponsive application is not used by this application.
`<min value>-604800`	Length of timeout, in seconds, where `min value` is:

- `40` seconds for Configuration Server and Solution Control Server.

- `10` seconds for applications that support hangup detection if you are using Solution Control Server 8.1.1 (or later).

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from an application. If Local Control Agent (LCA)

does not receive a heartbeat message from the application within this period, it assumes the application is not responding and carries out corrective action.

This option can also be used to specify the maximum heartbeat interval for threads registered with class zero (0). This thread class is reserved for use by the Management Layer only.

Note: Genesys does not recommend that you set the heartbeat period option for Configuration Server and Solution Control Server if you are using Solution Control Server 8.1.0.(or earlier).

If this option is not configured or is set to zero (`0`), heartbeat detection is not used by this application.

heartbeat-period-thread-class-<n>

Default Value: None
Valid Values:

`0`   Value specified by `heartbeat-period` in application is used.
`3-604800`   Length of timeout, in seconds; equivalent to 3 seconds–7 days.

Changes Take Effect: Immediately

Specifies the maximum amount of time, in seconds, in which heartbeat messages are expected from a thread of class <n> registered by an application. If a heartbeat message from the thread is not received within this period, the thread is assumed to be not responding, and therefore, the application is unable to provide service.

Note: Do not set this option to a value less than the `heartbeat-period` option.

If this option is not configured or is set to zero (`0`), but the application has registered one or more threads of class <n>, the value specified by the value of `heartbeat-period` for the application will also be applied to these threads.

Refer to application-specific documentation to determine what thread classes, if any, are used.

dbserver Section

In addition to options specific to an application that may exist in this section, this section also contains an option to support Oracle 11g RAC in TAF mode. When this option is set, Configuration Server can resubmit DML statements (DML transactions or binding package execution) when the appropriate error messages are received from the DBMS. For more information, refer to the *Framework Deployment Guide*.

This section must be called `dbserver`.

dml-retry

Default Value: 1
Valid Values: Integer values in the range of 0 to 32766
Changes Take Effect: After restart of Configuration Server

Specifies the number of retries for issuing a DML statement or transaction to DB Server after receiving TAF range error from Oracle DBMS. When the number of retries has been attempted with no success, Configuration Server considers the database operation to have failed and reports the error to the database client. A value of zero (0) specifies that no retry is to be attempted, in which the error is reported to the client immediately, without any retries.

Note: The node failover procedure can be time- and resource-consuming, so take care to set this option to a reasonable value to avoid overloading Configuration Server and the DBMS.

common Section

This section must be called common.

enable-async-dns

Default Value: 0
Valid Values:

0	Disables asynchronous processing of DNS requests.
1	Enables asynchronous processing of DNS requests.

Changes Take Effect: Immediately

Enables the asynchronous processing of DNS requests such as, for example, host-name resolution.

Warnings! • Use this option only when requested by Genesys Customer Care.
               • Use this option only with T-Servers.

enable-ipv6

Default Value: 0
Valid Values:

0	Off (default), IPv6 support is disabled.
1	On, IPv6 support is enabled.

Changes Take Effect: Immediately

When set to 1, specifies that this application supports IPv6. It is set to 0 by default to ensure backward compatibility. Refer to component-specific documentation and the *Framework Deployment Guide* for more information

about IPv6 and any specific considerations for deploying IPv6 in your situation.

rebind-delay

Default Value: `10`
Valid Values: `0–600`
Changes Take Effect: After restart

Specifies the delay, in seconds, between socket-bind operations that are being executed by the server. Use this option if the server has not been able to successfully occupy a configured port.

Warning!   Use this option only when requested by Genesys Customer Care.

Transport Parameter Options

Set options in this section in the `Transport Parameters` of the connection's properties. Transport Parameter options are not associated with a configuration option section, and do not appear in the options or annex of an `Application` object.

transport
Option

In a configuration file, these options appear in the following format:

```
transport = <option name>=<value>;<option name>=<value>; ...
```

Collectively, the options make up the parameters of the `transport` option. When entering the options in Genesys Administrator or Configuration Manager, only the options are required; `transport =` is prefixed automatically to the list of option/value pairs.

Note: Valid values for these options must have no spaces before or after the delimiter characters ";" (semi-colon) and "=".

Configuring Client-side Port Definition

This Transport Parameter options in this section are used to configure client-side port definition, Refer to the chapter "Client-Side Port Definition" in the *Genesys Security Deployment Guide* for information about how to use these options.

Set these options in one of the following navigation paths:

*   In Genesys Administrator—`Application` object > `Configuration` tab > General section > Connections > `<Connection>` > `Connection Info` > Advanced tab > `Transport Parameters`

- In Configuration Manager—`Application` object > `Properties` dialog box > `Connections` tab > `Connection Properties` dialog box > `Advanced tab >` `Transport Protocol Parameters`

port

Default Value: No default value
Valid Values: A valid port number
Changes Take Effect: After client application restart

The port that the client application uses for its TCP/IP connection to the server.

address

Default Value: No default value
Valid Values: A valid IP address

Optional. Specifies the IP address or host name that a client uses for its TCP/IP connection to the server.

backup-port

Default Value: No default value
Valid Values: A valid port number
Changes Take Effect: After client application restart

In an HA pair, the port on the backup server that the client application will use for its TCP/IP connection to the server.

Note: If the client application servers are in an HA pair, the `port` and `backup-port` values will be propagated from the primary server to the backup. As a result, after switchover, these ports will be in use by another server, so the new primary client application will be unable to open and use them.

To prevent this, Genesys recommends that you do one of the following:

- Locate the backup pair on different hosts.
- Manually change the `port` and `backup-port` settings for the backup server.

Configuring Secure Connections Using Security Certificates

The options in this section are used to define the use of security certificates and configuring secure connections between components. Refer to the chapter "Genesys TLS Configuration" in the *Genesys Security Deployment Guide* for more information about security certificates, configuring secure connections, and TLS security.

Unless otherwise stated, set options in this section in the `Transport Parameters` of the port's properties, using one of the following navigation paths:

- In Genesys Administrator—`Application` object > `Configuration` tab > `General` section > `Connections` > `Connection Info` > `Advanced` tab > `Transport Parameters`

- In Configuration Manager—`Application` object > `Properties` dialog box > `Connection` tab > `Connection Properties` dialog box > `Advanced` tab > `Transport Protocol Parameters`

cipher-list

Default Value: No default value
Valid Values: The list of ciphers
Changes Take Effect: After component restart
Related Option: crl

Specifies the defined list of ciphers. The cipher list must be in a valid format. See the chapter "Genesys TLS Configuration" in the *Genesys Security Deployment Guide* for information about cipher formatting rules and examples of valid cipher strings.

client-auth

Default Value: `1`
Valid Values: `0`, `1`
Changes Take Effect: After application restart

Specifies whether authentication of the security certificate in the client TLS socket is to be disabled. When set to `1` (default), authentication is enabled. When set to `0`, the client socket does not authenticate the server when connected over TLS.

This option must be set at the same level where the certificate itself is configured. Use this option if the security certificate is configured at the port level. If it is configured at another level, do one of the following:

- If the security certificate is configured at the application level, see page 65.
- If the security certificate is configured at the host level, set this option at the host level.

Note: If this option is configured at multiple levels (connection, application, host), the value set at the lowest level takes precedence. That is:
- The value set at the connection level takes precedence over the value set at the application and host levels.
- The value set at the application level takes precedence over the value set at the host level.

crl

Default Value: No default value
Valid Values: Valid file name

Changes Take Effect: After a server restart
Related Option: cipher-list

Specifies the name of the file that contains one or more certificates in `PEM` format, defining the Certificate Revocation List. As part of the authentication process, the system checks whether a presented certificate is included in this list of revoked certificates before completing authentication.

Note: Configuration of a CRL in SIP Server differs slightly from other Genesys components. Refer to the *Framework SIP Server Deployment Guide* for more information.

---

tls

Default Value: tls=`0`
Valid Values:

`tls=0`	Regular (unsecured) connections will be used.
For Windows: tls=1; certificate=<value>	Secure connections will be used, where `certificate` = certificate value.
For UNIX: tls=1; certificate=<path>; [certificate-key=<path>]; trusted-ca=<path>	Secure connections will be used, where: • `certificate`—full path to the `<serial_#>_<host_name>_cert.pem` file • `certificate-key`—full path to the `<serial_#>_<host_name>_priv_key.pem` file (unless the private key is stored together with a certificate) • `trusted-ca`—full path to the `ca_cert.pem` file

This option is used to enable the set up of secure connections between Genesys components. Refer to the section "Configuring Secure Configuration Server and DB Server Connections" in the *Genesys Security Deployment Guide* for information about how to use this option.

You also specify the `transport` option in any section of the DB Server configuration file that contains port configuration.

Changes from 8.0 to 8.1

Table 9 on page 74 lists all changes to common configuration options between release 8.0 and the latest 8.1 release.

Table 9: Common Configuration Option Changes from 8.0 to 8.1

Option Name	Option Values	Type of Change	Details
common Section			
enable-async-dns	`0, 1`	Changed values	Values `off` and `on` replaced with `0` and `1`, respectively.
enable-ipv6	`0, 1`	New	See description on page 69.
dbserver Section			
dml-retry	`0 to 32766`	New	See description on page 69.
log-filter Section			
default-filter-type	`copy; hide; hide-first, <n>; hide-last, <n>; skip; unhide-first, <n>; unhide-last, <n>; tag(tag-prefix, tag-postfix)`	Added two valid values in 8.1	See description on page 62.
	`copy; hide; hide-first, <n>; hide-last, <n>; skip; unhide-first, <n>; unhide-last, <n>`	New in 8.0	See description on page 62 Previously documented in Genesys Security Deployment Guide.
filtering	`true, false`	New	See description on page 63.
security Section			
client-auth	`1, 0`	New	See description on page 65.
tls-target-name-check	`no, host`	New	See description on page 66.
security-authentication-rules Section (new)			
no-change-password-at-first-login	`true, false`	New	See description on page 66.
sml Section			
heartbeat-period	`<min value> to 604800`	Changed minimum value	See description on page 67.

Table 9: Common Configuration Option Changes from 8.0 to 8.1 (Continued)

Option Name	Option Values	Type of Change	Details
Transport Parameters			
cipher-list	`List of ciphers`	New	See description on page 72.
client-auth	`1, 0`	New	See description on page 72.
crl	`Valid file name`	New	See description on page 72.
address	<valid IP address>	New in 7.6	Not documented in 7.6. See description on page 71.
backup-port	<any port number>	New in 7.6	Not documented in 7.6. See description on page 71.
port	<any port number>	New in 7.6	Not documented in 7.6. See description on page 71.

Genesys

A DB Server Directory Content on UNIX

When you are installing DB Server on UNIX, the installation script copies the following files to the installation directory:

- The `*.lms` files that enable the unified logging

- The sample configuration file `dbserver.conf.sample`

- The DB Server executable file `multiserver` (on the Tru64 UNIX and Linux operating systems) or `multiserver_32` and `multiserver_64` (on other UNIX operating systems)

  Note: The installation script makes a link to the appropriate executable file, so that a user need only specify `multiserver` as the executable file name.

- DB client files for all supported DBMS types, as specified in Table 10.

Table 10: DB Clients in the UNIX Installation Directory

File Name	Description
`dbclient_db2` (on Linux) `dbclient_db_32/dbclient_db2_64` (on AIX, HPUX, and Solaris)	DB client file for the DB2 DBMS
`dbclient_informix`	DB client file for the Informix DBMS (not present for the Linux operating system)
`dbclient_oracle_32` (`dbclient_oracle` on Tru64 UNIX and Linux)	DB client file for the 32-bit version of the Oracle DBMS releases 8i and higher

Table 10: DB Clients in the UNIX Installation Directory (Continued)

File Name	Description
dbclient_oracle_64	DB client file for the 64-bit version of the Oracle DBMS releases 8i and higher (not present for the Tru64 UNIX and Linux operating systems)
dbclient_postgre	DB client file for the PostgreSQL DBMS (on the Linux operating system only)
dbclient_sybase	DB client file for the Sybase DBMS (not present for the Linux operating system)

B

Troubleshooting DB Server

This appendix contains answers to frequently asked questions about DB Server as well as some information about troubleshooting common problems. Look through this information before contacting Genesys Customer Care with your questions and problems.

This appendix contains two sections:

Frequently Asked Questions

Question:

If I change the port of my MS SQL server, how do I connect DB Server?

Answer:

If you change the port of the MS SQL database, then you need to make the following changes in both the Client Network Utility and the Server Network Utility programs in MS SQL:

- On the General tab on the Client Network Utility, ensure that TCP/IP is enabled and click `Properties`. Enter the new port number and repeat this procedure on the Server Network Utility.
- Save the changes and restart DB Server.

Question:

What is the character limit for the stored procedure parameter?

Answer:

The maximum number of characters for processes are as follows:

- Parameter name: 64 bytes

- Number of parameters: 255

- Input/Output parameter limits: 2 KB

- The SQL statement passed from router to DB Server: 10240 bytes

Question:

What causes the error Unresolved symbol: `getrlimit64(code)` in the
DB Server log?

Answer:

Generally, this error message is caused either by a mismatch between
DB Server and the OS version (for instance, running a 32-bit DB Server on a
64-bit OS), or by using an outdated DB Server patch.

Question:

Why does DB Server seem to be losing 50% or more of all data writes?

Answer:

To date, only Universal Routing Server (Interaction Router) is capable of
handling multiple database processes simultaneously; DB Server client (for
example, Call Concentrator) can only handle one process. Therefore, you must
set the DB Server option `dbprocesses_per_client` to 1.

Question:

What shutdown sequence for DB Server should I use to avoid problems?

Answer:

The recommended sequence is to first shut down DB Server (if you are using
the Backup feature, bring the backup down first, followed by the primary
DB Server), and then shut down the database. When you restart, start the
database first, and then start DB Server.

Question:

What should I do if DB Server cannot connect to our Informix database?

Answer:

This problem occurs when the variables `INFORMIXDIR` and `INFORMIXSERVER` do
not specify the full path to the Informix home directory. The problem is

specific to Informix 7.1 and you can avoid it by upgrading to Informix 7.2. Consult your *Informix Server User's Guide* for details.

Question:

What do I do if DB Server cannot connect to the Informix database when I start the server as a Windows NT Service?

Answer:

This problem arises when the environment variables INFORMIXDIR and INFORMIXSERVER are not set in the computing environment of the person trying to start DB Server.

You can correct this by changing DB Server's service logon account from `System Account` to the user who is able to successfully start DB Server manually. To do this:

1. Open the Windows Control Panel and select `Services`.

2. Select the `Genesys DB Server` service.

3. Click `Startup`.

4. Select `This Account` in the `Logon As` section, and specify the `username` and `password` of the Windows NT user who is able to successfully start DB Server manually.

If you must start DB Server using `System Account`, contact your System or Database Administrator to correct your environment configuration.

Question:

What should I do if DB Server cannot connect to Oracle database and logs the following error:

`ORA-12154` or `TNS-12154` ?

Answer:

This error indicates that the Listener may be incorrectly configured in the `tnsnames.ora` file. Carefully check to ensure that the Listener is entered correctly in `tnsnames.ora`.

Question:

What should I do if DB Server cannot connect to an Oracle database and logs the following error:

`ORA-12560` ?

Answer:

To correct this problem, check that the `DBMS Name` parameter specified in the Database Access Point `Application` object configured in the Configuration

Layer is correct—it must match the name of the Listener configured in the `tnsnames.ora` file. Note that the `Database Name` parameter, although mandatory for all other database types, is not required for Oracle and thus can be left blank.

Question:

What should I do if DB Server cannot connect to Oracle database and records the following error:

`ORA-12203 or TNS-12203` ?

Answer:

To correct this problem, verify that the option `dbserver` in DB Server's client configuration matches the setting of the `ORACLE_SID` environment variable on the computer on which DB Server is running. It might also be helpful for you to analyze the `sqlnet.log` log file in the DB Server installation folder. This file is created by the Oracle Client in case there is a need to troubleshoot a network configuration problem.

Question:

What is DB Server's BLOB data size limit?

Answer:

DB Server can handle a maximum 10 MB.

Question:

Is it possible to start multiple DB Servers on the same computer?

Answer:

Yes, provided that they use separate ports.

However, when running multiple DB Server instances in the same UNIX environment, you must install the instances in different directories. In addition, the setting for the `dbprocess_name` option must begin with a period and a forward slash (./) (for example, `./process_name`). This will ensure that DB Server starts the DB client from the current directory. Otherwise, the DB client may read the wrong configuration file and be unable to connect to the correct DB Server.

Question:

Is it possible to configure DB Server to automatically reconnect to a database after a database problem is corrected?

Answer:

DB Server connects to the database only when its client application sends a request to open the database. If the database is down for any reason, all such client requests fail, and DB Server notifies the client that the database is not available. The application must reconnect (and reopen the database) after receiving such an error.

Question:

How do I configure Automatic Disconnection Detection Protocol (ADDP) on DB Server?

Answer:

Genesys recommends enabling ADDP on the links between any pair of Genesys components. ADDP helps detect a connection failure on both the client and the server side. You configure ADDP in Genesys Administrator of Configuration Manager. For detailed instructions, refer to the *Framework 8.1 Deployment Guide*, *Framework 8.1 Genesys Administrator Help*, or *Framework 8.1 Configuration Manager Help*.

Troubleshooting Common Problems

This section discusses the solutions to several problems commonly encountered with DB Server.

Note: Set the value of the `verbose` option to 3 (see Chapter 3, "verbose," on page 38) when you are troubleshooting DB Server. This setting provides a detailed log that may indicate the source of a particular problem. After you resolve the problem, reset the `verbose` option to its previous level to increase performance.

Increase the Processes Allowed Limit in Windows

By default, Microsoft Windows limits each local system account to approximately 120 processes that are running as—or are started by—services. When the number of running processes reaches this limit, any attempt to start an additional process will fail. Because Windows counts DBClient processes which are started by DB Server, 120 can be too low a limit. One example of when this may happen is during Outbound deployment with a high number of calling lists.

To change the number of processes allowed, follow the procedure described in the Microsoft Knowledge Base article located here: `http://www.support.microsoft.com/?id=184802`, which describes changing the registry value:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session
Manager\SubSystems\Windows
```

In this procedure, each additional 512 kilobytes that you specify in the third `SharedSection` parameter will allow approximately 120 additional processes/services to be started. The default `SharedSection` setting is highlighted in the default registry value data:

```
%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows
SharedSection=1024,3072,512 Windows=On SubSystemType=Windows
ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3
ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Off
MaxRequestThreads=16
```

Estimate the number of processes/services and apply it to the above calculation. Consider this example:

- 65 services are running under the local system account.

- You plan to start DB Server as a service under the local system account.

- You plan to use 60 connections from the Genesys DB Server to the DBMS.

These estimates require 125 services, which exceeds the default limit of 120. In this example, Genesys recommends that you to set the third `SharedSection` parameter to 1024, which increases the limit to 240. Use the Registry Editor to change the setting to this:

```
SharedSection=1024,3072,1024
```

Warning! Within the procedure, Microsoft presents a disclaimer about the Registry Editor:

> ```
> Warning: If you use Registry Editor incorrectly, you may
> cause serious problems that may require you to reinstall
> your operating system. Microsoft cannot guarantee that you
> can solve problems that result from using Registry Editor
> incorrectly. Use Registry Editor at your own risk.
> ```

Client Cannot Connect to DB Server

When a client connects to DB Server the following message is printed in the log:

```
DB Server - new client XX connected to server
```

If you see this message in the log, you are connecting properly. If it does not appear, it means the client is unable to connect to DB Server. In this case:

- Check that the DB Server `host` and `port` parameters specified in the Database Access Point `Application` object configured in the Configuration Layer (or the client's configuration file if the client has one) are correct. They must exactly match the `host` and `port` of DB Server as specified in the DB Server `Application` object configured in the Configuration Layer (or the DB Server configuration file).

> **Note:** Using `localhost` for the server defined in Configuration Layer may cause problems if DB Server and the client are not running on the same computer.

- Make sure that DB Server is listening on the port you have specified in the Configuration Layer (configuration file). Whenever DB Server starts, it prints the port used:

```
10/21/03 21:47:32.555   Trace   KARADAG   DBServerLog
GCTI-00-04525   Port 7030 opened for listening
```

- If it is not the port you have specified in the configuration, check the command line you used for startup.
- From the host of the client that cannot connect to DB Server, try to open a Telnet session to the DB Server host and port specified in the Database Access Point `Application` object configured in the Configuration Layer (or the client's configuration file if the client has one). If the Telnet connection fails, the problem lies in your network rather than in DB Server or the client. In this case, contact your Network Administrator.

DB Server Cannot Start the DB Client Process

If you see the following message in your log, there is a problem with starting DB client:

```
21:15:28.494 Std 05020 Cannot start process 'dbclient_oracle', error
code 2
Error start dbclient process 'dbclient_oracle' type='2', copy='0'
errno='No such file or directory'
```

- Make sure that the executable file specified in the log message does exist. Note that on some UNIX systems this name must be preceded by a period and a forward slash (`./`) to make it start from the current directory (for example, `./dbclient_oracle`).
- Make sure that the user starting DB Server also has permission to start the DB client executable file.
- Try to start the DB client executable manually without the command-line argument.

DB Server Cannot Open the SQL Server Database

- Make sure the SQL Server client is properly installed and configured on the machine on which DB Server is running (see Chapter 5, "Deploying DB Server" on page 13 for more information).

- When DB Server tries to open the database, the following message appears in the log:

```
DBClient 96 – Server: username = 'user', password = '***********',
database = 'test', DB Server = 'cti'
```

Try to open the database manually (with the SQL Server client) using the values from the log message. If you can't do this, the problem lies with your SQL Server client configuration. In this case, contact your Database Administrator.

Request to Execute SQL Statement Fails

Check the log file for error messages. DB Server prints the error code and message the SQL Server returns. You may need to contact your Database Administrator for information on database-specific error codes.

Check the log for the text of the SQL command that fails. Try to execute the same command manually with SQL Server client, using the same username, password, and so on as specified in the Configuration Layer (configuration file).

C Installing and Configuring an Oracle Client

Notes: • This procedure can be used when DB Server is installed as either a standalone (it is used to access the Configuration Database) or as a client of Configuration Server.
• This procedure is specific to installing Oracle 11.2 on the Linux platform.

To install and configure an Oracle client and make it work with DB Server, do the following, referring to Oracle documentation for details about the specific steps.

1. Download and install the Oracle Instant Client Package - Basic, version 11.2.0.4.0, from the Oracle web site:

   http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html

   This installs the client into the `/usr/lib/oracle/11.2/client64/` folder.

2. Create a link to the client so that DB Server can find the library. Enter the following command on the Oracle command-line:

   `ln -s libclntsh.so.11.1 libclntsh.so`

3. Create the `tnsnames.ora` file.

   Note: If you are using a Linux user without a profile (for example, to run DB Server), place this file into the `/etc` folder.

4. In `/etc/environment`, set `TNS_ADMIN=<location of tnsnames.ora file>`.

5. Create the configuration file `/etc/ld.so.conf.d/oracle-11.2-client64.conf` containing the line `LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client64/lib`.

6. Run `ldconfig`.

7. Reboot the computer.

Related Documentation Resources

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

DB Server

* *Framework 8.1 Architecture Help*, which will help you identify DB Server's place in the Genesys software architecture.
* *Framework 8.1 Deployment Guide*, which will help you set up DB Server for the Configuration Database.
* *Genesys Administrator 8.1 Help*, which will help you set up DB Server as a client of the Configuration Database.
* Release Notes and Product Advisories for Genesys products, which are available on the Genesys Documentation website at `http://docs.genesys.com`.

Genesys

* *Genesys 8.1 Security Deployment Guide*, which helps you set up security features for DB Server and your data.
* *Genesys Technical Publications Glossary*, which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
* *Genesys Migration Guide*, which provides documented migration strategies for Genesys product releases. Contact Genesys Customer Care for more information.

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

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

Consult the following 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 Gplus Adapters Interoperability.

- *Genesys Licensing Guide,* which introduces you to the concepts, terminology, and procedures that are relevant to the Genesys licensing system.

- *Genesys Database Sizing Estimator 8.x Worksheets,* which provides a range of expected database sizes for various Genesys products.

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

Genesys product documentation is available on the:

- Genesys Customer Care website at http://genesys.com/customer-care.

- Genesys Documentation website at http://docs.genesys.com/.

- Genesys Documentation Library DVD and/or the Developer Documentation CD, which you can order by e-mail from Genesys Order Management at orderman@genesys.com.

Document Conventions

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

Document Version Number

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

80fr_ref_06-2008_v8.0.001.00

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

Screen Captures Used in This Document

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

Type Styles

Table 11 describes and illustrates the type conventions that are used in this document.

Table 11: 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 92).	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 11: 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

U

V

W

X

Framework 8.1