GENESYS®
AN ALCATEL-LUCENT COMPANY

Framework 7.6

DB Server

User’s Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

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

Notice

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

Your Responsibility for Your System's Security

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

Trademarks

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

Technical Support from VARs

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

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-118-974-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-5649-6871	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the *Genesys Technical Support 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 7 Licensing Guide***.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 76fr_us-dbs_10-2007_v7.6.001.00

Table of Contents

Preface

Welcome to the *Framework 7.6 DB Server User's Guide*. This guide 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 guide is valid only for the 7.6 release of this product.

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

This chapter contains these sections:

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

Intended Audience

This guide, primarily intended for system administrators and database architects, assumes 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 Configuration Manager.

Chapter Summaries

In addition to this opening chapter, this guide contains these chapters and appendixes:

- Chapter 1, “Deploying DB Server,” on page 11, describes how to configure and install DB Server so that it runs as a client of Configuration Server. This chapter also helps you configure Database Access Points for DB Server.
- Chapter 2, “DBMS Environment Settings,” on page 19, provides recommendations on setting up DBMS clients for supported types of databases.
- Chapter 3, “DB Server Configuration Options,” on page 25, describes configuration options for DB Server.
- Chapter 4, “Common Configuration Options,” on page 31 describes configuration options, including log options, that are common to all Genesys server applications and applicable to any Framework server component.
- Appendix A, “DB Server Directory Content on UNIX” on page 51, lists the files, including those for various DB clients, that are copied to DB Server installation directories on UNIX operating systems.
- Appendix B, “Troubleshooting DB Server” on page 53, provides answers to frequently asked questions.

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:

76fr_ref_10-2007_v7.6.000.01

You will need this number when you are talking with Genesys Technical Support about this product.

Type Styles

Italic

In this document, italic is used for emphasis, for documents’ titles, for definitions of (or first references to) unfamiliar terms, and for mathematical variables.

Examples:

- Please consult the *Genesys 7 Migration Guide* for more information.
- *A customary and usual practice* is one that is widely accepted and used within a particular industry or profession.
- Do *not* use this value for this option.
- The formula, x +1 = 7 where x stands for . . .

Monospace Font

A monospace font, which looks like `teletype or typewriter text`, is used for 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; and code samples.

Examples:

- Select the `Show variables on screen` check box.
- Click the `Summation` button.
- In the `Properties` dialog box, enter the value for the host server in your environment.
- In the `Operand` text box, enter your formula.
- Click `OK` to exit the `Properties` dialog box.
- The following table presents the complete set of error messages T-Server® distributes 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.

Monospace is also used for any text that users must manually enter during a configuration or installation procedure, or on a command line:

Example:

- Enter `exit` on the command line.

Screen Captures Used in This Document

Screen captures from the product GUI (graphical user interface), as used in this document, may sometimes contain a minor spelling, capitalization, or grammatical error. 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.

Square Brackets

Square brackets indicate that a particular parameter or value is optional within a logical argument, a command, or some programming syntax. That is, the parameter's or value's presence is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information. Here is a sample:

```
smcp_server -host [/flags]
```

Angle Brackets

Angle brackets indicate a placeholder for a value that the user must specify. This might be a DN or port number specific to your enterprise. Here is a sample:

```
smcp_server -host <confighost>
```

Related Resources

Consult these additional resources as necessary:

- *Framework 7.6 Architecture online help,* which will help you identify DB Server's place in the Genesys software architecture.
- *Framework 7.6 Deployment Guide*, which will help you set up DB Server for the Configuration Database.
- The *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library CD and which provides a comprehensive list of the Genesys and CTI terminology and acronyms used in this document.
- The *Genesys 7 Migration Guide*, also on the Genesys Documentation Library CD, which contains a documented migration strategy for Genesys product releases 6.x and later. Contact Genesys Technical Support for additional information.
- The Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at `http://genesyslab.com/support.`

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

- *Genesys 7 Supported Operating Systems and Databases*
- *Genesys 7 Supported Media Interfaces*

Genesys product documentation is available on the:

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

Making Comments on This Document

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

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

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

Chapter

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 machine. Refer to the DBMS vendor's documentation for details. This document also provides some specific recommendations in Chapter 2 on page 19.

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 7.6 Deployment Guide*.

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

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.

Configuring DB Server

Unless DB Server is installed as a part of Configuration Layer, providing access to the Configuration Database, it must be a client of Configuration Server. To achieve this, configure an `Application` object of type `DB Server` using Configuration Manager.

For other applications to access the Database, you must configure both DB Server and the DAP as `Application` objects using Configuration Manager. For configuration instructions for DAPs, see “Configuring Database Access Points” on page 16.

Before creating a DB Server `Application` object, make sure that you have created the `Host` object upon which the DB Server `Application` object will reside. In addition, make sure that you have the DB Server Application Template that corresponds to the DB Server version you are using. This template provides a majority of the DB Server configuration options and their default values.

To make sure you have a DB Server Application Template:

1. In Configuration Manager, select Environment > Application Templates and locate the DB Server template.
2. If the template is absent, import the file DB_Server_<current-version>.apd from the Templates folder on the Management Framework 7.6 product CD, or create one by selecting Application from the tree view of the Configuration Manager main window, then choosing File > New > Application Template. Refer to the *Framework 7.6 Deployment Guide* for detailed instructions for importing or creating new templates.

To configure a DB Server Application object:

1. Open the Configuration Manager main window.
2. Right-click the Environment > Applications folder and select New > Application from the popup menu, which opens the Browse dialog box with the available Application Templates.
3. Select the DB Server template file, and click OK.
4. On the General tab, enter a descriptive name in the Name text box. This name must be unique.
5. On the Server Info tab:
   a. Click the Browse button next to the Host drop-down list, and select the Host object.
   b. Specify the listening port(s) and select whether each is secure or not secure. Refer to the *Genesys 7.6 Transport Layer Security Deployment Guide* for detailed information on specifying ports and securing connections to them.
   c. Leave the rest of the fields at their default values.
6. On the Start Info tab, in the Working Directory, Command Line, and Command Line Arguments text boxes, do one of the following:
   - Enter the appropriate information in the text boxes. For information about command-line parameters, see the *Framework 7.6 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 DB Server, if the Installation Package can connect to Configuration Server.

7. On the Options tab, in the dbserver section:
   a. Change the value of the option dbprocess_name to the value corresponding to the option <DBMS you are using>_name. For example, if you are using Microsoft SQL DBMS, set the value dbprocess_name to ./dblient_msql.
   b. Change the value of the option management-port 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 page 25 and Chapter 4 on page 31.
8. Click OK.

Multiple Ports Configuration

Starting with release 7.5, DB Server can communicate with its clients via multiple ports. To configure additional listening ports:

1. Open the DB Server Application object.
2. Click the Server Info tab of the Properties window.
3. In the Ports section, click Add Port. The Port's Properties dialog box opens.
4. Specify the Port ID and the Communication Port that applications will use to connect to this server.
5. Click OK.

Installing DB Server

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

Installing on UNIX

1. On the Management Framework 7.6 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. The process places DB Server in the directory specified during the installation.

Installing on Windows

1. From the Management Framework 7.6 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 where you want DB Server installed.
7. When prompted, click Finish to complete the installation.

Starting DB Server

For complete instructions on starting and stopping DB Server, refer to the *Framework 7.6 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 page 19 for more information.

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

Configuring Database Access Points

Before creating a DAP Application object, verify that you have the required DAP Application Template.

To make sure you have a DAP Application Template:

1. From the Configuration Manager main window, select Environment > Application Templates and locate the DAP template.
2. If the template is absent, import the file Database_Access_Point_<current-version>.apd from the Templates folder on the Management Framework product CD, or create one by selecting Application from the tree view of the Configuration Manager main window, then choosing File > New > Application Template. Refer to the *Framework 7.6 Deployment Guide* for detailed instructions for importing or creating new templates.

This document describes how to configure DAPs when you use DB Server to access a database. When you configure a DAP for DB Server, do not configure the fields that are not applicable to a DB Server connection.

To configure a DAP associated with DB Server to access a particular database:

1. Open the Configuration Manager main window.
2. Right-click on the Environment > Applications folder and select New > Application from the popup menu, which opens the Browse dialog box with the available Application Templates.
3. Select the DAP template file, and click OK.
4. On the General tab:
   a. In the Name field, enter a descriptive name.

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

   b. In the DB Server field, use the Browse button to select the Application object corresponding to the DB Server that you just installed.

Notes: Do not select the JDBC Connection check box when you configure DAPs.

Do not alter the properties of the JDBC Info when you use DB Server for database access unless instructed otherwise by

Genesys Technical Support or by the *User's Guide* of the applicable Genesys solution.

5. On the DB Info tab, specify information about the database as follows:
   - DBMS Name—The name or alias identifying the SQL server DBMS that handles the database. Clients use the value of this option in their requests to DB Server to open the database.
     - For Sybase, this value is the name of SQL server, stored in the Sybase interface file.
     - For Oracle, this value is the name of the Listener service.
     - 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 where Microsoft SQL runs).
     - For DB2, set this value to the name or alias-name of the database specified in the db2 client configuration.
   - DBMS Type—The type of DBMS that handles the database. You must specify a value for this parameter.
   - Database Name—The name of the database as it is specified in your DBMS. You must select a value for this parameter from the list unless oracle or db2 is specified as the DBMS Type. For Sybase, Informix, and Microsoft SQL, this value is the name of the database where the client will connect.
   - User Name—The user name established to access the database. You must specify a value for this parameter.
   - Password—The password used to access the database.
   - Reenter Password—Confirmation of 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. Note that 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.

Warning! Use the default value (any) unless instructed otherwise by Genesys Technical Support or by the *User's Guide* of the applicable Genesys solution.

6. Click OK.

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

Chapter

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

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 on page 21 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
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 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	
Windows	PATH	%INFORMIXDIR%\BIN

Genesys DB Client for DB2

You must set the following variables for DB client for DB2:

```
DB2INSTANCE
INSTHOME
```

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 the 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	

Chapter

3 DB Server Configuration Options

This chapter describes configuration options for DB Server and includes the following sections:

DB Server also supports the options described in Chapter 4 on page 31.

Setting Configuration Options

Unless otherwise specified in this chapter or in the documentation for your application, you set DB Server configuration options in Configuration Manager, in the corresponding sections on the `Options` tab of the DB Server `Application` object.

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

Mandatory Options

Table 5 on page 26 lists the DB Server options for which you must provide values; otherwise, DB Server will not start. The options are listed by section.

Table 5: Mandatory Options

Option Name	Default Value	Details
DB Server Section		
dbprocess_name	No default value	See the description on page 26.
[a DB client process name option]	See the option specific to your DBMS type.	The option name depends on the DBMS type: `oracle_name`, `sybase_name`, `informix_name`, `db2_name`, or `msql_name`. See the descriptions beginning on page 27.

DB Server Section

This section must be called `dbserver`.

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.

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`.

dbprocess_name

Default Value: No default value
Valid Values (use these names for the appropriate application):

`dbclient_db2`	For DB2
`dbclient_informix`	For Informix
`dbclient_msql`	For Microsoft SQL

dbclient_oracle	For Oracle
dbclient_sybase	For Sybase

Changes Take Effect: After restart

Specifies the type of a DB client process. This option works with dbprocesses_per_client and with the appropriate database name option (oracle_name, for example). The name must begin with a period and a slash (./dbclient_oracle, for example).

Note: Only enable the dbprocess_name option for compatibility with previous releases of client applications (5.1, 6.0, or 6.1).

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 the dbprocesses_per_client option unless instructed otherwise by Technical Support or by the *User's Guide* of the applicable Genesys solution. Changing the default value (1) of this option may cause data loss.

db2_name

Default Value: dbclient_db2
Valid Values: Any valid name
Changes Take Effect: After restart

Specifies the name of the DB client process for the DB2 server. **This option is required for DB2 databases.** Also see "dbprocess_name" on page 26.

informix_name

Default Value: dbclient_informix
Valid Values: Any valid name
Changes Take Effect: After restart

Specifies the name of the DB client process for the Informix server if present. **This option is required for Informix databases.** Also see “dbprocess_name” on page 26.

msql_name

Default Value: `dbclient_msql`
Valid Values: Any valid name
Changes Take Effect: After restart

Specifies the name of the DB client process for the Microsoft SQL server. **This option is required for MSSQL databases.** Also see “dbprocess_name” on page 26.

oracle_name

Default Value: `dbclient_oracle`
Valid Values: Any valid name
Changes Take Effect: After restart

Specifies the name of the DB client process for the Oracle server if present. **This option is required for Oracle databases.** Also see “dbprocess_name” on page 26.

sybase_name

Default Value: `dbclient_sybase`
Valid Values: Any valid name
Changes Take Effect: After restart

Specifies the name of the DB client process for the Sybase server if present. **This option is required for Sybase databases.** Also see “dbprocess_name” on page 26.

client_stop_timeout

Default Value: `30`
Valid Values: `0` or any positive integer
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.

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.

dbprocess_number

Default Value: `255`
Valid Values:

`0`	Does not impose restrictions to the number of running DB Client processes
`1` and above	Sets 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.

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 did 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.

tran_batch_mode

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

Valid only for Microsoft SQL and 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 the `tran_batch_mode` option unless instructed otherwise by Technical Support or by the *User's Guide* of the applicable Genesys solution.

transport

Default Value: `tls=0`
Valid Values:

`tls=0`	Regular (unsecured) connections will be used.
For Windows:	Secure connections will be used.
`tls=1;certificate=<value>`	Specify the certificate value in the `certificate` option.
For UNIX:	Secure connections will be used.

`tls=1;certificate=<path>;`
`[certificate-key=<path>];`
`trusted-ca=<path>`
	Specify the full path to the certificate files.

Changes Take Effect: After restart

Specifies the parameters for secure client connections to this DB Server. Refer to the *Genesys 7.6 Security Deployment Guide* for more information.

Changes from 7.5 to 7.6

There are no changes to DB Server configuration options between release 7.5 and the latest 7.6 release.

Chapter

4 Common Configuration Options

Unless otherwise noted, the common configuration options that this chapter describes are common to all Genesys server applications and applicable to any Framework server component. 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 it is otherwise specified in this document or in the documentation for your application, you set common configuration options in Configuration Manager in the corresponding sections on the `Options` tab of the `Application` object.

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

Mandatory Options

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

Log Section

This section must be called `log`.

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 38.

Note: For definitions of the Standard, Interaction, Trace, and Debug log levels, refer to the *Framework 7.6 Deployment Guide* or to *Framework 7.6 Solution Control Interface Help.*

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 38). 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.

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.

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 1–100.
<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 T-Server configuration, 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 set to `false`).

Note: This option applies only to T-Servers.

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.

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.

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
```

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 7.6 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.

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.

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 page 38). 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`).

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 page 38.

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.

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 the `compatible-output-priority` option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to `true` affects log consistency.

Log Output Options

To configure log outputs, set log level options (`all`, `standard`, `interaction`, `trace`, 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 42.

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

Warnings!

- If you direct log output to a file on the network drive, an application does not create a snapshot log file (with the extension `*.snapshot.log`) 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.

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.

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 Technical Support when reporting a problem.

- `*.memory.log`—Assigned to log files that contain the memory output snapshot when you configure output to memory and redirect the most recent memory output to a file. For example, if you set `standard = memory` and `memory = 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. Use this configuration when trying to reproduce an application's failure. The memory log file will contain a snapshot of the application's log at the moment of failure; this should help you and Genesys Technical Support identify the reason for the 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 Technical Support when reporting a problem.

Debug Log Options

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

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

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 Technical Support.

Log-Extended Section

This section must be called `log-extended`.

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 the option `level-reassign-disable` (see page 48).

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 *URS 7.6 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.

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.

Log-Filter Section

This section must be called `log-filter`.

default-filter-type

Default Value: `copy`
Valid Values:

`copy` The keys and values of the KVList pairs are copied to the log.

`hide` The keys of the KVList pairs are copied to the log; the values are replaced with strings of asterisks.

`skip` The KVList pairs are not copied to the log.

Changes Take Effect: Immediately

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

Example

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

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

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

Log-Filter-Data Section

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

<key name>

Default Value: `copy`
Valid Values:

`copy`	The key and value of the given KVList pair are copied to the log.
`hide`	The key of the given KVList pair is copied to the log; the value is replaced with a string of asterisks.
`skip`	The KVList pair is not copied to the log.

Changes Take Effect: Immediately

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

Example

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

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

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

Common Section

This section must be called `common`.

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 Technical Support.

Changes from 7.5 to 7.6

Table 6 provides all the changes to common configuration options between release 7.5 and the latest 7.6 release.

Table 6: Common Log Option Changes from 7.5 to 7.6

Option Name	Option Values	Type of Change	Details
Log Section			
Use the following options only when requested by Genesys Technical Support.			
x-conn-debug-open	0, 1	New	See the description on page 43.
x-conn-debug-select	0, 1	New	See the description on page 44.
x-conn-debug-timers	0, 1	New	See the description on page 44.
x-conn-debug-write	0, 1	New	See the description on page 44.
x-conn-debug-security	0, 1	New	See the description on page 44.
x-conn-debug-api	0, 1	New	See the description on page 45.
x-conn-debug-dns	0, 1	New	See the description on page 45.
x-conn-debug-all	0, 1	New	See the description on page 45.
Extended Log Section (New Section)			
level-reassign-<eventID>	alarm, standard, interaction, trace, debug, none	New	See the description on page 46.
level-reassign-disable	true, false	New	See the description on page 48.
Common Section (New Section)			
Use the following options only when requested by Genesys Technical Support.			
rebind-delay	10–600	New	See the description on page 49.

Appendix

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 7.

Table 7: DB Clients in the UNIX Installation Directory

File Name	Description
`dbclient_db2`	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 7: 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_sybase`	DB client file for the Sybase DBMS (not present for the Linux operating system)

Appendix

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 Technical Support with your questions and problems.This appendix has two sections:

- Frequently Asked Questions, page 53
- Troubleshooting Common Problems, page 57

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 then click on `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:
  - Oracle: 2 KB
  - MS SQL: 256 bytes
  - Informix: 2 KB
  - DB2: 2 KB
  - Sybase: 256 bytes
- 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 machine where 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 machine?

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 Configuration Manager. For detailed instructions, refer to the *Framework 7.6 Deployment Guide*, or to *Framework 7.6 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 28) 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 machine.

- 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 where DB Server is running (see Chapter 1, "Deploying DB Server" on page 11 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).

Index