GENESYS™

Framework Database Connectivity Guide

Management Framework 8.5.0

12/29/2021

Table of Contents

Framework Database Connectivity Reference Guide

This guide describes the concepts and procedures relevant to how Genesys software connects to databases.

Overview

Databases in Genesys

Database connectivity prior to release 8.5

Database Access Points

Databases in multi-language environments

Failure of database access functionality

Environment Variable Settings

General recommendations

Oracle databases

PostgreSQL databases

IBM DB2 databases

Microsoft SQL Server databases

Database Access Points

Configuring a DAP

Using DAPs

FAQs and Troubleshooting

Frequently Asked Questions

Troubleshooting

Overview

In Genesys software, data is stored in databases. Server applications, such as Configuration Server, manage this data by connecting to the database.

Databases in Genesys

In any Genesys environment, there is only one database (or a database cluster, if configured on the DBMS level) that is intended for use by Configuration Server – this is the Configuration Database.

In addition to the Configuration Database, there can be a number of other databases that Genesys servers and GUI applications may need to access. Applications access these databases directly, using information from Database Access Points (DAPs) to locate the database and obtain access credentials. In Management Framework, for example, Message Server accesses the Log Database through a DAP.

Connectivity Prior to Release 8.5

Prior to release 8.5, a DB Server Application object was used to access one or more databases. Release 8.5 effectively streamlines database access by removing DB Server from the access path.

However, if you still want to use DB Server in your configuration, or if you have legacy applications that cannot access database directly, do the following:

1. For new applications only: Use the configuration option that controls how a database is accessed by the application, either using DB Server or direct database access capability, and set it to the appropriate value. In Configuration Server, for example, this option is called **dbthread**. Refer to application-specific documentation for the name and description of the option, if there is one, used by the particular application.

2. Follow the instructions and information in Framework 8.1 documentation to deploy and use DB Server.

Important

If you choose to use DB Server, be aware that you will be unable to access any of the new database-related features and functionality introduced in release 8.5. In addition, you will be able to use DB Server with only those Database Management Systems supported in 8.5; you will not be able to use it with any DBMS that is no longer supported.

Database Access Points

To provide an interface between applications in the Genesys installation and databases, the Configuration Layer uses the concept of a Database Access Point (DAP). If, according to your configuration, a database can be accessed by multiple servers simultaneously, register one DAP for each potential connection.

See Database Access Points for detailed information about installing DAPs.

Databases in Multi-language Environments

To be used in a multi-language environment, a database must be able to support data that can be encoding in different, or a common, format. Normally, this is done by encoding data using UTF-8.

The DBMS-specific sections of Environment Settings contain information about using the DBMS in multi-language environments; refer to these sections for more information and instructions.

Failure of Database Access Functionality in Genesys Servers

The Management Layer can detect internal failure of the database access module within Genesys applications if you configure the unresponsive process detection feature, and specify that it should detect application thread failures. See the *Framework Management Layer User's Guide* and the related option descriptions in the *Framework Configuration Options Reference Manual* for more information about setting up this feature for the particular application for which you want to wanted the database thread.

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.

Environment Settings

To work with a particular DBMS, an application (the Genesys DB Client application) requires particular environment settings, as listed in the following sections:

- General recommendations
- Oracle databases
- Postgre databases
- IBM DB2 databases
- Microsoft SQL Server databases

General Recommendations

The recommendations in this section apply regardless of the type of DBMS that you are using.

DBMS Versions

32-bit or 64-bit

Make sure that you are installing DBMS vendor client software that matches the 32-bit or 64-bit Genesys software that you want to enable for database access.

For example, if your Genesys application is 64-bit, then make sure that it can access 64-bit DBMS client software, as provided by your database vendor.

Client Software Version

Make sure you are using the correct version of DBMS client software, as given in the following sections, for each type of database. All Genesys applications of a particular release use the same version of DBMS client software. Even if you are accessing a database of different versions, you might still need to have another version of DBMS client software on the host where the Genesys application is installed.

Genesys supports multiple versions of a DBMS using the same version of DBMS client software. For example, you need Oracle 11g client software to access both Oracle 11 and Oracle 12 databases.

DBMS Encoding

Make sure encoding on DBMS is set correctly to match the encoding being used in the Genesys environment. If you are using single-language (in addition to English-US), you must create your database with the respective encoding, as described in the following DBMS-specific pages of this section. If you are using a multi-language environment, UTF-8 encoding should be used on the DBMS, also as described in the DBMS-specific sections.

> ### Important
> You will not be able to use some Genesys applications in multi-language mode.

Database Failures

Starting in release 8.0, a database client process can detect a connection failure with the corresponding database and attempt to reconnect. To detect the failure, database clients monitor the responses they receive from the DBMS. If a response is not received within the interval specified by the configuration option **db-request-timeout**, the client process stops executing. This is understood to be a failure of the DBMS.

The option **db-request-timeout** is configured in the **Query Timeout** field of Database Access Point (DAP) Application objects and stored in the DAP's annex. The timeout set in the DAP overrides the timeout set in the database client application, but applies only to client processes that connect to the database through this DAP.

See Creating a DAP for more information about how to configure this timeout for Log Database access.

Oracle Databases

You must have Oracle Client software accessible in the environment where the Genesys application is running. Genesys uses Oracle 11g client software to access all supported versions of Oracle.

Using Full Installation of Oracle Client

Connectivity to an Oracle database relies on TCP/IP to work between an Oracle server and its client.

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, and either **LIBPATH** for AIX or **LD_LIBRARY_PATH** for Linux and Solaris, depending on the platform you are using. If the DBMS client for Oracle runs on a different host other than the Oracle server, you must also configure the **SQLNet** file on both hosts. Note that a TCP/IP Adapter is also required. For more information, refer to the Oracle documentation for your platform.

If you are using the UNIX/Linux platform, you might want to create a link to the DB client so that the server accessing the database can find the library using the default name. Enter the following on the operating system command line:

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

You must use the **tnsnames.ora** file to specify database access as defined in Oracle documentation. Genesys DB Client will load the **.tns** file.

The following is an example of TNS name content:

```
CMES =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST =<your oracle host>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = cmes)
    )
  )
```

With this definition in the **.tns** file, the parameters of the Genesys Database Access Point can be set as follows:

```
dbengine=oracle
dbserver=<your oracle host>:1521/cmes
dbname=
username=<oracle schema user>
password=<oracle schema user's password>
```

Using Oracle Instant Client

Instead of a full Oracle client installation, you can use the Instant Client package from Oracle, downloadable from here.

For core Genesys servers to work, you must have the Basic package. If you have to perform conversion of character set encodings for languages other than English, make sure that the Instant Client package contains all necessary encoding tables. If you cannot find the proper package of Instant Client, you may want to use a full Oracle installation.

Connectivity to an Oracle database relies on TCP/IP to work between the Oracle server and client. You do not need to set up the **ORACLE_HOME** or **ORACLE_SID** environment variables. However, you must specify the full path to the folder where you put the Oracle Instant Client in the appropriate environment variable—**PATH** for Windows, **LIBPATH** for AIX, or **LD_LIBRARY_PATH** for Linux and Solaris.

For example, the configuration of Genesys Database Access Points without using TNS should be as follows:

```
dbengine=oracle
dbserver=<oracle host>:1521/<oracle service name>
dbname=
username=<oracle schema user>
password=<oracle schema user's password>
```

You can still use TNS-based connection information with Instant Client if you set up the **TNS_ADMIN** environment variable, or you can use the SQL Connect URL string <host>[:port]/<service name>. Both are described in Oracle Instant Client documentation.

Service Name

Instant Client requires a Service Name, rather than the SID required by the full Oracle installation package. The Service Name can be found in the **tnsnames.ora** file, which is found in the **<oracle home>/network/admin** folder on the server. For example, in the following excerpt from the **tnsnames.ora** file, the SID appears below ORCL, and the Service Name is orcl.us.int.genesyslab.com.

```
ORCL =
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = <IP of a host>)(PORT = 1521))
  (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = orcl.us.int.genesyslab.com)
  )
 )
```

Using Oracle Database with National Languages

Single Language Deployment

You must create all Oracle databases using the same character set, such as WE8MSWIN1252. You must

select an encoding that matches Microsoft Windows Operating System default encoding for a selected language, so that applications, like Interaction Routing Designer, display data correctly.

On every host that has a Genesys application accessing an Oracle database, make sure that the **NLS_LANG** environment variable is set to match the language and character encoding of data in the Oracle database, as defined in the following table . For example, **NLS_LANG**=`AMERICAN_AMERICA.WE8MSWIN1252`.

[+] Show table

Operating System Locale	NLS_LANG Environment Variable Value
Arabic (U.A.E.)	ARABIC_UNITED ARAB EMIRATES.AR8MSWIN1256
Bulgarian	BULGARIAN_BULGARIA.CL8MSWIN1251
Catalan	CATALAN_CATALONIA.WE8MSWIN1252
Chinese (PRC)	SIMPLIFIED CHINESE_CHINA.ZHS16GBK
Chinese (Taiwan)	TRADITIONAL CHINESE_TAIWAN.ZHT16MSWIN950
Croatian	CROATIAN_CROATIA.EE8MSWIN1250
Czech	CZECH_CZECH REPUBLIC.EE8MSWIN1250
Danish	DANISH_DENMARK.WE8MSWIN1252
Dutch (Netherlands)	DUTCH_THE NETHERLANDS.WE8MSWIN1252
English (United Kingdom)	ENGLISH_UNITED KINGDOM.WE8MSWIN1252
English (United States)	AMERICAN_AMERICA.WE8MSWIN1252
Estonian	ESTONIAN_ESTONIA.BLT8MSWIN1257
Finnish	FINNISH_FINLAND.WE8MSWIN1252
French (Canada)	CANADIAN FRENCH_CANADA.WE8MSWIN1252
French (France)	FRENCH_FRANCE.WE8MSWIN1252
German (Germany)	GERMAN_GERMANY.WE8MSWIN1252
Greek	GREEK_GREECE.EL8MSWIN1253
Hebrew	HEBREW_ISRAEL.IW8MSWIN1255
Hungarian	HUNGARIAN_HUNGARY.EE8MSWIN1250
Icelandic	ICELANDIC_ICELAND.WE8MSWIN1252
Indonesian	INDONESIAN_INDONESIA.WE8MSWIN1252
Italian (Italy)	ITALIAN_ITALY.WE8MSWIN1252
Japanese	JAPANESE_JAPAN.JA16SJIS
Korean	KOREAN_KOREA.KO16MSWIN949
Latvian	LATVIAN_LATVIA.BLT8MSWIN1257
Lithuanian	LITHUANIAN_LITHUANIA.BLT8MSWIN1257
Norwegian	NORWEGIAN_NORWAY.WE8MSWIN1252
Polish	POLISH_POLAND.EE8MSWIN1250
Portuguese (Brazil)	BRAZILIAN PORTUGUESE_BRAZIL.WE8MSWIN1252
Portuguese (Portugal)	PORTUGUESE_PORTUGAL.WE8MSWIN1252
Romanian	ROMANIAN_ROMANIA.EE8MSWIN1250

Operating System Locale	NLS_LANG Environment Variable Value
Russian	RUSSIAN_CIS.CL8MSWIN1251
Slovak	SLOVAK_SLOVAKIA.EE8MSWIN1250
Spanish (Spain)	SPANISH_SPAIN.WE8MSWIN1252
Swedish	SWEDISH_SWEDEN.WE8MSWIN1252
Thai	THAI_THAILAND.TH8TISASCII
Spanish (Mexico)	MEXICAN SPANISH_MEXICO.WE8MSWIN1252
Spanish (Venezuela)	LATIN AMERICAN SPANISH_VENEZUELA.WE8MSWIN1252
Turkish	TURKISH_TURKEY.TR8MSWIN1254
Ukrainian	UKRAINIAN_UKRAINE.CL8MSWIN1251
Vietnamese	VIETNAMESE_VIETNAM.VN8MSWIN1258

For more information, see the Oracle documentation here.

If you are unable to setup MS Windows compatible character encoding when creating the Oracle database, make sure that the Oracle client software on all hosts with Genesys applications has been set to use character encoding that matches the target as close as possible, by following these steps:

1. Set up **NLS_LANG** to use the closest compatible encoding. For example, WE8IS08858P1 to match WE8MSWIN1252 if you are using Linux to host Genesys applications that should access Oracle Databases.

2. Make sure that Oracle client software contains NLS tables allowing conversion between character encoding of the database and the host. Refer to Oracle documentation for more information about supported character conversions.

With your environment set up this way, you can use character data in a single language (such as French) for all information stored and transmitted between Genesys applications.

Multi-Language Deployment

To enable storage and processing of data in multiple languages using Oracle Databases, you must create all your database instances using the AL32UTF8 character set. For example:

```
CREATE DATABASE orclutf8
...
  CHARACTER SET AL32UTF8
  NATIONAL CHARACTER SET AL16UTF16
```

On every host that has a Genesys application accessing an Oracle database, make sure the **NLS_LANG** environment variable is set to match the character encoding of data in the Oracle database; for example, **NLS_LANG=**.UTF8.

Using Oracle TAF

Genesys supports using the Oracle Real Application Cluster to provide redundant access to database

storage. You must use the TNS file to define cluster access, as specified in Oracle documentation.

For example:

```
LORAC =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST =<oracle RAC listener> )(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME =<oracle service name>)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD=BASIC)
        (RETRIES = 2)
        (DELAY = 1)
      )
    )
  )
```

The **RETRIES** and **DELAY** parameters might affect how long the Genesys application will wait for Oracle to respond before it attempts to reconnect. Refer to the Oracle TAF guide for more details.

Failure of an Oracle 12g RAC Database

The Oracle 12g Real Application Cluster (RAC) DBMS includes a client-side feature called Transparent Application Failover (TAF). If an instance of a database fails, TAF automatically reconnects to a surviving database instance (node).

However, TAF only restores the connection; it is the responsibility of the application to restart on the new node any operations that were in process on the failed node. These operations could be any of the following:

- Individual Data Manipulation Language (DML) statements, such as INSERT, UPDATE, and DELETE.

- Active transaction involving DML statements, issuing ROLLBACK instructions to these transactions first.

- Active binding packages.

To support Oracle 12g RAC in TAF mode, Configuration Server can optionally resubmit DML statements (DML transactions or binding package execution) when the appropriate error messages are received from the DBMS. This is implemented using the configuration option **dml-retry**. Refer to the *Framework Configuration Options Reference Manual* for the full description of this option.

PostgreSQL Databases

You must make PostgreSQL client software accessible in the environment where the Genesys application is running. Genesys uses PostgreSQL 9.0 client software to access all supported versions of PostgreSQL.

Using PostgreSQL Client Software

The vendor client software must be in the folder specified in the environment variable **PATH** (for Windows), **LIBPATH** (for AIX), or **LD_LIBRARY_PATH** (for Linux and Solaris). Refer to PostgreSQL documentation for more information.

The following is an example of the configuration of Genesys Database Access Point parameters for PostgreSQL:

```
dbengine=postgre
dbserver=<postgresql server host>
dbname=<database name>
username=<user name>
password=<password>
```

Connectivity to PostgreSQL relies on TCP/IP between server and client. PostgreSQL client software uses Operating System settings for the TCP/IP stack to determine how long to wait for a response form the PostgreSQL server after submitting a request to it. For example, on Linux it may take up to 2 hours to detect a disconnection, unless the Operating System parameter **tcp_keepalive_time** is adjusted. Refer to documentation for your operating system for more information.

When using PostgreSQL 9.X server, make sure the following parameters are set in the **postgresql.conf** file, as follows:

```
bytea_output ='escape'
standard_conforming_strings='off'
```

PostgreSQL Strings

Prior to PostgreSQL version 9.1, backslash characters in strings were treated as escape characters, and Configuration Server does the same, by default. In PostgreSQL 9.1 and later, backslashes are treated as ordinary characters. How PostgreSQL handles backslashes in strings is specified by the PostgreSQL configuration option **standard_conforming_strings** in the **postgresql.conf** file, in which a value of `off` indicates the backslashes are treated as escape characters, to provide backward-compatible behavior.

Normally, the PostgreSQL option is set to `false`, and no action is required. However, if the PostgreSQL option is set to `true`, meaning that backslashes in strings are treated as ordinary characters, enable the Configuration Server option **postgre-standard-conforming-strings** to force Configuration Server to do the same.

> **Important**
>
> The value of the Configuration Server option **postgre-standard-conforming-strings** must be always equal to the value of the PostgreSQL configuration option **standard_conforming_strings** in the **postgresql.conf** file.

For more information about the Configuration Server option **postgre-standard-conforming-strings**, refer to the *Framework 8.5 Configuration Options Reference Manual*.

Using PostgreSQL Databases with National Languages

Single Language Deployment

You must create all PostgreSQL databases using the same character set, for example WIN1252. You must select encoding that matches Microsoft Windows Operating System default encoding for the selected language so applications, like Interaction Routing Designer, display data correctly.

On every host that has Genesys applications accessing PostgreSQL databases, make sure that the language and encoding environment variable (or the settings for non-Unicode applications, if you are using Windows) is set to match character encoding of data in the PostgreSQL database. If there is a discrepancy between the encoding that the database and the local client are using, set the environment variable **PGCLIENTENCODING** on the host where the client software is running to match the database (for example, **PGCLIENTENCODING=**Win1252), based on the following table:

[+] Show table

Name	Description	Language	Server?	Bytes/Char	Aliases
GBK	Extended National Standard	Simplified Chinese	No	1-2	WIN936, Windows936
ISO_8859_5	ISO 8859-5, ECMA 113	Latin/Cyrillic	Yes	1	
ISO_8859_6	ISO 8859-6, ECMA 114	Latin/Arabic	Yes	1	
ISO_8859_7	ISO 8859-7, ECMA 118	Latin/Greek	Yes	1	
ISO_8859_8	ISO 8859-8, ECMA 121	Latin/Hebrew	Yes	1	
JOHAB	JOHAB	Korean (Hangul)	Yes	1-3	
LATIN1	ISO 8859-1, ECMA 94	Western European	Yes	1	ISO88591
LATIN2	ISO 8859-2, ECMA 94	Central European	Yes	1	ISO88592
LATIN3	ISO 8859-3,	South	Yes	1	ISO88593

Name	Description	Language	Server?	Bytes/Char	Aliases
	ECMA 94	European			
LATIN4	ISO 8859-4, ECMA 94	North European	Yes	1	ISO88594
LATIN5	ISO 8859-9, ECMA 128	Turkish	Yes	1	ISO88599
LATIN6	ISO 8859-10, ECMA 144	Nordic	Yes	1	ISO885910
LATIN7	ISO 8859-13	Baltic	Yes	1	ISO885913
LATIN8	ISO 8859-14	Celtic	Yes	1	ISO885914
LATIN9	ISO 8859-15	LATIN1 with Euro and accents	Yes	1	ISO885915
LATIN10	ISO 8859-16, ASRO SR 14111	Romanian	Yes	1	ISO885916
SJIS	Shift JIS	Japanese	No	1-2	Mskanji, ShiftJIS, WIN932, Windows932
SQL_ASCII	unspecified	any	Yes	1	
UHC	Unified Hangul Code	Korean	No	1-2	WIN949, Windows949
UTF8	Unicode, 8-bit	all	Yes	1-4	Unicode
WIN866	Windows CP866	Cyrillic	Yes	1	ALT
WIN874	Windows CP874	Thai	Yes	1	
WIN1250	Windows CP1250	Central European	Yes	1	
WIN1251	Windows CP1251	Cyrillic	Yes	1	WIN
WIN1252	Windows CP1252	Western European	Yes	1	
WIN1256	Windows CP1256	Arabic	Yes	1	
WIN1258	Windows CP1258	Vietnamese	Yes	1	ABC, TCVN, TCVN5712, VSCII

For more information, refer to PostgreSQL documentation here.

With the environment set up this way, you can use character data in a single language (such as French) for all information stored and transmitted between Genesys applications.

Multiple Languages Deployment

To enable storage and processing of data in multiple languages using a PostgreSQL database, you must create all your database instances using the UTF8 character set.

IBM DB2 Databases

You must make DB2 client software accessible in the environment where the Genesys application is running. Genesys uses IBM DB2 9.7 client software to access all supported versions of DB2.

Using IBM DB2 Client Software

Vendor client software must be in the folder specified in the environment variable **PATH** (for Windows), **LIBPATH** (for AIX), or **LD_LIBRARY_PATH** (for Linux and Solaris).

You must set the **DB2INSTANCE** and **INSTHOME** environment variables for the DB client for DB2. For more information, refer to DB2 documentation.

Using DB2 Databases with National Languages

Single Language Deployment

You must create all DB2 databases using the same character set, such as IS0-8859-1, as provided in the following table. You must select an encoding that matches Microsoft Windows Operating System default encoding for the selected language so applications, like Interaction Routing Designer, display data correctly.

[+] Show table

Category	Encoding	Codepage
ASCII	iso-8859-1	819
	ibm-1252	1252
	iso-8859-2	912
	iso-8859-5	915
	iso-8859-6	1089
	iso-8859-7	813
	iso-8859-8	916
	iso-8859-9	920
MBCS	gb2312	1386
	ibm-932, shift_jis78	932
	Shift_JIS	943
	IBM-eucCN	1383
	ibm-1388	1388
	IBM-eucJP, EUC-JP	954, 33722

Category	Encoding	Codepage
	ibm-930	930
	ibm-939	939
	ibm-1390	1390
	ibm-1399	1399
	ibm-5026	5026
	ibm-5035	5035
	euc-tw, IBM-eucTW	964
	ibm-937	937
	euc-kr, IBM-eucKR	970
	big5	950

For more information about encoding and DB2 codepages, see IBM documentation here.

Multiple Languages Deployment

For a DB2 database to store multiple languages, it must be created using the UTF-8 codeset.

On every host running a Genesys application that accesses a DB2 multi-language database, set the environment variable **DB2CODEPAGE=**1208.

Microsoft SQL Server Databases

You must install software to access the version of Microsoft SQL Server you are using. Refer to Microsoft documentation for details. You can use any edition of Microsoft SQL Server, including Express.

Using Microsoft Client Software

Genesys uses TCP/IP as a way to access Microsoft SQL Sever. When installing Microsoft SQL Server and/ or Microsoft client software, make sure that Server and Client are using TCP/IP. Dynamic ports are not supported; you must configure the server to listen on a fixed port (1433).

You can access default instances or named instance (including Express) of Microsoft SQL Server. To use a default instance, set the following parameters of the Database Access Point:

```
dbengine = mssql
dbserver = <sql server host>
dbname = <database name>
username = <user>
password = <password>
```

If a named (non-default) instance is used, the **dbserver** parameter must be specified in the format:
dbserver = <sql server host>\<named instance>

Or for the Microsoft SQL Express edition:
dbserver = <sql server host>\sqlexpress

Notes for Management Framework Components

- To use native client connectivity, you must install MS SQL 2005 Server Native Client software (the **sqlncli** package), available from here. Without this software, the connection is made using ODBC, which can be slower.

- To work with MS SQL databases, Configuration Server and Message Server require Microsoft Data Access Components (MDAC) version 2.8 or later.

- For MS SQL databases, DB Server did not correctly read international characters that were written to the database if both of the following conditions existed:

    - The records were originally written using DB Server 7.2 or earlier.

    - On the host on which DB Server was running, the option **SQL Server Client Network Utility > DB-Library Options > Automatic ANSI to OEM** conversion was turned on.

Using Microsoft SQL Server Databases with National Languages

Single Language Deployment

No special configuration or other preparations are needed to use Genesys applications in single language mode with Microsoft SQL Server databases. The databases themselves must be created with target language and default encoding, as given in the following table:

[+] Show table

Sort Order ID	SQL Server Collation Came
30	SQL_Latin1_General_Cp437_BIN
31	SQL_Latin1_General_Cp437_CS_AS
32	SQL_Latin1_General_Cp437_CI_AS
33	SQL_Latin1_General_Pref_CP437_CI_AS
34	SQL_Latin1_General_Cp437_CI_AI
40	SQL_Latin1_General_Cp850_BIN
41	SQL_Latin1_General_Cp850_CS_AS
42	SQL_Latin1_General_Cp850_CI_AS
43	SQL_Latin1_General_Pref_CP850_CI_AS
44	SQL_Latin1_General_Cp850_CI_AI
49	SQL_1Xcompat_CP850_CI_AS
50	Latin1_General_BIN
51	SQL_Latin1_General_Cp1_CS_AS
52	SQL_Latin1_General_Cp1_CI_AS
53	SQL_Latin1_General_Pref_CP1_CI_AS
54	SQL_Latin1_General_Cp1_CI_AI
55	SQL_AltDiction_Cp850_CS_AS
56	SQL_AltDiction_Pref_CP850_CI_AS
57	SQL_AltDiction_Cp850_CI_AI
58	SQL_Scandinavian_Pref_Cp850_CI_AS
59	SQL_Scandinavian_Cp850_CS_AS
60	SQL_Scandinavian_Cp850_CI_AS
61	SQL_AltDiction_Cp850_CI_AS
71	Latin1_General_CS_AS
72	Latin1_General_CI_AS
73	Danish_Norwegian_CS_AS
74	Finnish_Swedish_CS_AS
75	Icelandic_CS_AS
80	Hungarian_BIN (or Albanian_BIN, Czech_BIN, and so on)

Sort Order ID	SQL Server Collation Came
	See Note
81	SQL_Latin1_General_Cp1250_CS_AS
82	SQL_Latin1_General_Cp1250_CI_AS
83	SQL_Czech_Cp1250_CS_AS
84	SQL_Czech_Cp1250_CI_AS
85	SQL_Hungarian_Cp1250_CS_AS
86	SQL_Hungarian_Cp1250_CI_AS
87	SQL_Polish_Cp1250_CS_AS
88	SQL_Polish_Cp1250_CI_AS
89	SQL_Romanian_Cp1250_CS_AS
90	SQL_Romanian_Cp1250_CI_AS
91	SQL_Croatian_Cp1250_CS_AS
92	SQL_Croatian_Cp1250_CI_AS
93	SQL_Slovak_Cp1250_CS_AS
94	SQL_Slovak_Cp1250_CI_AS
95	SQL_Slovenian_Cp1250_CS_AS
96	SQL_Slovenian_Cp1250_CI_AS
104	Cyrillic_General_BIN (or Ukrainian_BIN, Macedonian_FYROM_90_BIN)
105	SQL_Latin1_General_Cp1251_CS_AS
106	SQL_Latin1_General_Cp1251_CI_AS
107	SQL_Ukrainian_Cp1251_CS_AS
108	SQL_Ukrainian_Cp1251_CI_AS
112	Greek_BIN
113	SQL_Latin1_General_Cp1253_CS_AS
114	SQL_Latin1_General_Cp1253_CI_AS
120	SQL_MixDiction_Cp1253_CS_AS
121	SQL_AltDiction_Cp1253_CS_AS
124	SQL_Latin1_General_Cp1253_CI_AI
128	Turkish_BIN
129	SQL_Latin1_General_Cp1254_CS_AS
130	SQL_Latin1_General_Cp1254_CI_AS
136	Hebrew_BIN
137	SQL_Latin1_General_Cp1255_CS_AS
138	SQL_Latin1_General_Cp1255_CI_AS
144	Arabic_BIN
145	SQL_Latin1_General_Cp1256_CS_AS

Sort Order ID	SQL Server Collation Came
146	SQL_Latin1_General_Cp1256_CI_AS
153	SQL_Latin1_General_Cp1257_CS_AS
154	SQL_Latin1_General_Cp1257_CI_AS
155	SQL_Estonian_Cp1257_CS_AS
156	SQL_Estonian_Cp1257_CI_AS
157	SQL_Latvian_Cp1257_CS_AS
158	SQL_Latvian_Cp1257_CI_AS
159	SQL_Lithuanian_Cp1257_CS_AS
160	SQL_Lithuanian_Cp1257_CI_AS
183	SQL_Danish_Pref_Cp1_CI_AS
184	SQL_SwedishPhone_Pref_Cp1_CI_AS
185	SQL_SwedishStd_Pref_Cp1_CI_AS
186	SQL_Icelandic_Pref_Cp1_CI_AS
192	Japanese_BIN
193	Japanese_CI_AS
194	Korean_Wansung_BIN
195	Korean_Wansung_CI_AS
196	Chinese_Taiwan_Stroke_BIN
197	Chinese_Taiwan_Stroke_CI_AS
198	Chinese_PRC_BIN
199	Chinese_PRC_CI_AS
200	Japanese_CS_AS
201	Korean_Wansung_CS_AS
202	Chinese_Taiwan_Stroke_CS_AS
203	Chinese_PRC_CS_AS
204	Thai_BIN
205	Thai_CI_AS
206	Thai_CS_AS
210	SQL_EBCDIC037_CP1_CS_AS
211	SQL_EBCDIC273_CP1_CS_AS
212	SQL_EBCDIC277_CP1_CS_AS
213	SQL_EBCDIC278_CP1_CS_AS
214	SQL_EBCDIC280_CP1_CS_AS
215	SQL_EBCDIC284_CP1_CS_AS
216	SQL_EBCDIC285_CP1_CS_AS
217	SQL_EBCDIC297_CP1_CS_AS

> ### Important
>
> For Sort Order ID 80, use any of the Window collations with the code page of 1250, and binary order. For example: Albanian_BIN, Croatian_BIN, Czech_BIN, Romanian_BIN, Slovak_BIN, Slovenian_BIN.

For more information, refer to Microsoft SQL documentation here.

Multiple Languages Deployment

To use Microsoft SQL to store data in multiple languages, the database tables must be able to store UNICODE characters (UCS-2 encoding).

When configuring a Database Access Point to access a multi-language database, you must specify **utf8-ucs2=**true in the **[dbclient]** section of the annex of the DAP.

Failure of an MSSQL 2012 Cluster Database

There is no automatic resubmission for MSSQL. If the database fails, you must manually resubmit all failed write operations.

Database Access Points

To provide an interface between applications in the Genesys installation and databases to which the applications require access, 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 the database itself. If, according to your configuration, a database can be accessed by multiple applications simultaneously, register one DAP for each possible connection.

Creating a DAP

To create a DAP, you do not have to install a DAP; you only need to configure it.

[+] Show steps

Prerequisites

- You are logged in to Genesys Administrator.

- The database to which the DAP is to provide access exists.

Start of procedure

1. In Genesys Administrator, create a new Application Template for the Database Access Point. Refer to the Additional Information > Generic Deployment and Login Procedures section of the *Framework Deployment Guide* for instructions.

2. Go to **Provisioning > Environment > Applications**, click **New**, and import the DAP template you just created.

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

4. In the **General** section, enter a descriptive name in the **Name** field; for example, MyDAP.
   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.

5. In the **DB Info** section, provide the following information about the Database:

   - **Connection Type**—The type of connection to the DBMS.

   - **Query Timeout**—The period of time for which a database client process using this DAP expect a response from the DBMS. If the client process does not receive a response within this period, it stops executing. This is interpreted as a failure of the DBMS. The timeout set in this DAP overrides that set in the database client, but applies only to database client processes using this DAP. For more information about how DB Server uses this value, see Database Failure.

   - **DBMS Name**—The name or alias identifying the DBMS that handles the database.

     - For DB2, set this value to the name or alias-name of the database specified in the DB2 client

configuration.

- 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).

> ## Important
>
> For named instances of MS SQL server it must be specified in the format:
>   <computer name>\<instance name>

- For Oracle, set this value to the name of your listener service as specified in the TNS file, or, if you are using Oracle Instant Client and do not have a TNS file, use the format <oracle host>/<service name>.

- For PostgreSQL, set this value to the SQL server name (usually the same as the host name of the computer on which PostgreSQL runs).

6. **DBMS Type**—The type of DBMS that handles the database. You must set a value for this property.

7. **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 Microsoft SQL and PostgreSQL, this value is the name of the database where the client will connect.

8. **User Name**—The user name established in the SQL server to access the database. You must set a value for this property.

9. **User Password**—The password established in the SQL server to access the database.

10. **Re-enter Password**—Confirmation of the value entered for **User Password**.

11. **Case Conversion**—Case conversion method for key names of key-value lists coming from the database client process. This value specifies whether and how a client application converts the field names of a database table when receiving data from the database client process. 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 the database client process. That is, actual data is being presented exactly as in the database tables.

> ## Important
>
> For the **Case Conversion** option, use the default value (any) unless directed to do otherwise by Genesys Customer Care.

12. If the Log Database is an MS SQL database, and has been initialized for use in multi-language environments, select UTF-8 for MSSQL.

13. Click **Save** or **Apply** in the toolbar to save the new object. The new object will appear in the list of Applications.

End of procedure

> ## Warning
>
> If you are still using DB Server to access the database, you must also specify the following parameters:
>
> - **Host** and **Port**
> - **dbserver**

Using DAPs

To interface an Application object with a database through a certain DAP, add the DAP to the list of the Application's connections.

DAP Configuration Options

Configuration options for DAP are set by values that you enter on the Configuration tab when creating a DAP object in Genesys Administrator. For more information about the options, refer to the *Framework Configuration Options Reference Manual*.

FAQs and Troubleshooting

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

Frequently Asked Questions

Question

If I change the port of my MS SQL server, how do I connect to my database?

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 your DB Client.

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 Client's log?

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

Question

What should I do if the DB Client cannot connect to Oracle database and logs the 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 the DB Client cannot connect to an Oracle database and logs the error **ORA-12560**?

Answer:
To correct this problem, check that the **DBMS Name** parameter specified in the DAP 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 **DBMS 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 the DB Client cannot connect to Oracle database and records the error **ORA-12203** or **TNS-12203**?

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

Troubleshooting

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

> ### Important
> Set the value of the **verbose** option to 3 when you are troubleshooting a DB Client. 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, 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, 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. For 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 Genesys DB Client applications to the DBMS.

These estimates require 125 services, which exceeds the default limit of 120.

In this example, Genesys recommends that you set the third **SharedSection** parameter to 1024, which increases the limit to 240. Use the Registry Editor to change the setting, as follows:

SharedSection=1024,3072,1024

> ### Warning
> Within the procedure, Microsoft presents a disclaimer about the Registry Editor: 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.

DB Client Cannot Open the SQL Server Database

- Make sure the SQL Server client is properly installed and configured on the machine on which the DB Client is running (see the *Framework Deployment Guide* 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. 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 the SQL Server client, using the same username, password, and so on as specified in the Configuration Layer (configuration file).