Framework Database Connectivity Guide

Oracle Databases

5/10/2025

Oracle Databases

Contents

You must have Oracle Client software accessible in the environment where the Genesys application is running.

Genesys Framework supports Oracle 18c and 19c for both server and clients. Framework as an application requires static library files of Oracle 12.x, which is linked as part of the application itself, and can work with Oracle 18c or Oracle 19c deployment. You can choose to use the alternative version as described in Using an Alternative Version of Oracle Client Software.

See the *Genesys Supported Operating Environment Reference Guide* for a list of all supported databases.

> Warning
>
> You must be using Oracle Full Client if you want to configure secure connections. Do not configure TLS/SSL support if you are using Oracle Instant Client.

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 for Windows, 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** (i.e., **sqlnet.ora**) 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 need to create a symbolic link in the **$ORACLE_HOME\lib** directory and this directory must be included in your **LD_LIBRARY_PATH** or **LIBPATH**, 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.<xx.x> libclntsh.so
```

Note: Where <xx.x> is the version number of the Oracle client software you are using, either `11.1`, `12.1`, or `12.2`.

You must use the **tnsnames.ora** file to specify database access as defined in Oracle documentation. Genesys DB Client will subsequently use the content in the **tnsnames.ora** file to process the DB connection request.

The following is an example of the **tnsnames.ora** file 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=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 UNIX/Linux platform, the **libclntsh.so** symbolic link should be created in the same directory as your Oracle Instant Client libraries and this directory should be included in **LD_LIBRARY_PATH** or **LIBPATH**, depending on the platform.

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 `myserver.example.com`.

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

Using an Alternative Version of Oracle Client Software

Genesys provides newer versions of some dbclients in the **dbclient_next** folder of the component's Installation Package (IP). Use these processes to enable support of new databases and features. Only 64-bit versions of alternate clients are provided.

To use the latest functionality with a component, you must do the following:

1. Replace the dbclients in the root installation folder with the dbclients from the **dbclient_next** folder.
2. On the hosts that will be using this component, install the database client software and make it available for the component.

Secure Communications with Oracle DBMS

You can use Transport Layer Security (TLS) to secure connections with an Oracle database.

> Warning
>
> You must be using Oracle Full Client if you want to configure secure connections. Do not configure TLS/SSL support if you are using Oracle Instant Client.

Can Secure Connections be Set?

Before you can configure secure connections, you have to determine if the Oracle **tnsnames.ora** file can be set to enforce TLS/SSL. To do this, open the **tnsnames.ora** file and look for the **SECURITY** section and its accompanying parameter **SSL_SERVER_CERT_DN** under DESCRIPTION. It will look something like this:

```
<net_service_name>=
 (DESCRIPTION=
   (ADDRESS=...)
   (ADDRESS=...)
   (CONNECT_DATA=
    (SERVICE_NAME=...))
   (SECURITY=
    (SSL_SERVER_CERT_DN=...)))
```

If this section and its parameter is present, you can configure secure connections.

For more information about the **tnsnames.ora** file and its parameters, refer to the tnsnames.ora section of the *Oracle Database Net Services Reference*.

Configure the Secure Connections

Secure connections using TLS with the Oracle database are configured in the **tnsnames.ora** file, not in or by the DAP or Configuration Server configuration files. However, these files contain the **net_service_name** used to locate the TNS listener definition in the client configuration.

To configure the secure connections, do the following:

1. On the Oracle server, open the **listener.ora** file and configure the following parameters:

   - **SID_LIST_LISTENER**
   - **WALLET_LOCATION**
   - **LISTENER**

   For example:

```
# listener.ora Network Configuration File: /opt/oracle/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = db01)
      (ORACLE_HOME = /opt/oracle/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME = db01)
    )
  )

SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/app/oracle/product/11.2.0/dbhome_1/owm/wallets/oracle)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = myserver.example.com)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = myserver.example.com)(PORT = 2484))
    )
  )
```

```
ADR_BASE_LISTENER = /opt/oracle/app/oracle
```

For more information about the **listener.ora** file and its parameters, refer to the Listener Control Utility section of the *Oracle Database Net Services Reference*.

2. On the Oracle client, open the **tnsnames.ora** file and configure the following:
   - The Oracle secure port **2484** (default) to use the secure protocol **TCPS**.
   - The **SECURITY** section.

   For example:

```
DB_GENCHNGC3031_SSL=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (ADDRESS= (PROTOCOL = TCPS)(HOST = GEN-CHN-GC3-031)(PORT = 2484))
    (CONNECT_DATA=
      (SERVER= DEDICATED)
      (SERVICE_NAME= GENCHNGC3031.genesys.lab))
    (SECURITY =
      (SSL_SERVER_CERT_DN="CN=GENCHNGC3031.genesys.lab,C=US")))
```

Note: For SSL_SERVER_CERT_DN, specify the value within double-quotes (") on Windows platform and use single-quotes (') on UNIX platform.

For more information about the **tnsnames.ora** file and its parameters, refer to the tnsnames.ora section of the *Oracle Database Net Services Reference*.

3. On the Oracle client, open the **sqlnet.ora** file and configure the following parameters:
   - **SQLNET.AUTHENTICATION_SERVICES**
   - **SSL_SERVER_DN_MATCH**
   - **WALLET_LOCATION**

   For example:

```
SQLNET.AUTHENTICATION_SERVICES= (BEQ, TCPS)
```

```
SSL_SERVER_DN_MATCH= ON
```

```
WALLET_LOCATION=
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = C:\app\wallet)))
```

For more information about the **sqlnet.ora** file and its parameters, refer to the sqlnet.ora section of the *Oracle Database Net Services Reference*.

Additional Information

For additional information about using TLS with Oracle databases, refer to the *Oracle Database Advanced Security Administrator's Guide*, and *Configuring Secure Sockets Layer Authentication* in particular.

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`.

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

Operating System Locale	NLS_LANG Environment Variable Value
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
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 Database

The Oracle 12c and above 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 12c and above 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.