



Oracle Driver Installation

Introduction

Enterprise Tester is a .Net Framework application and makes use of ODP.Net to connect to Oracle databases.

ODP.Net provides a ADO.Net Provider for Oracle, which communicates with Oracle using the native drivers.

The native drivers can be either 32bit or 64bit, and you must install the ODAC/ODP.Net release specific for your server and IIS application pool settings.

ODP.Net comes as part of ODAC (Oracle Data Access Components) - which can be downloaded from the Oracle website. In most organisations it's normally advised that you have your Oracle DBA install these drivers and configure the settings for Enterprise Tester to connect to Oracle, as they will be better equipped to troubleshoot connectivity issues.

Preparation

Prior to installation you need to determine if you are going to use the 64bit or 32bit version of ODAC - here is how you can determine which you should use.

Use 64bit (x64) ODAC	Use 32bit (x86) ODAC
<ul style="list-style-type: none"> • Operating system is 64bit (Note: Windows Server 2008 r2 is only 64bit) • Application pool used by Enterprise Tester has "Enable 32-Bit applications" set to False. • You do not need to use Enterprise Architect integration in Enterprise Tester with .EAP files (which use the Jet driver, which is 32bit only) • You need the Enterprise Tester application to be able to reference more than 4gb of data. 	<ul style="list-style-type: none"> • Operating system is 32bit or 64bit • Application pool used by Enterprise Tester has "Enable 32-Bit applications" set to False, or server is 32bit. • You need to use Enterprise Architect integration in Enterprise Tester with .EAP files (which use the Jet driver, which is 32bit only).

If after reading the above information you determine that you have a 64-bit operating system, but need to use the 32-bit drivers, then you will need to Enable 32-bit application support in IIS, to do so:

- Go to Control Panel -> Administrative Tools -> Internet Information Services (IIS) Manager.
- On the left hand side select Application Pools.
- Click on the Application Pool being used by Enterprise Tester (this will normally be "DefaultAppPool" or "EnterpriseTester").
- Select on the right-hand side "Actions" panel "Advanced Settings...".
- In the general section of "Advanced Settings" change "Enable 32-Bit Applications" from "False" to "True".

Downloading

For ODAC you can either download and install it, or, download a zip file of the binaries and then copy the specific files required by Enterprise Tester into the bin folder for the Enterprise Tester installation directory.

If installing ODAC, this will place the ODP.Net files (required by Enterprise Tester) into the "Global Assembly Cache" (GAC) - this can impact on other .Net applications using ODP.Net as well, so if you have a server with a lot of applications, care may need to be taken.

The other option (also called "XCOPY" deployment of ODAC) - means that we take just the files required by ODP.Net (including the native Oracle drivers), and copy them into the bin folder for Enterprise Tester (normally c:\Program Files\Catch Limited\Enterprise Tester\Web\bin). This option involves less risk, but a little more effort.

Please visit the Oracle website to download the latest ODAC release - for convenience I have included links to both below:

- [ODAC XCopy x64 \(64bit\)](#)
- [ODAC XCopy x86 \(32bit\)](#)
- [ODAC list of files](#)

Installing

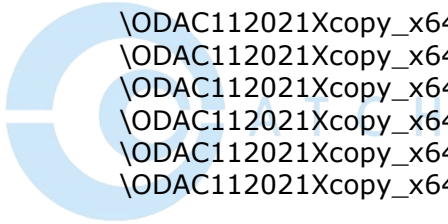
If you have downloaded an ODAC release with an installer, you can now install it (as per instructions accompanying the download) - each release changes the installer slightly, so I won't document this process here. [More details are here on the Oracle site.](#)

If you have downloaded the XCOPY version, then unzip the file to a directory, then copy the following files from the download to the destination directory:

32-bit

```
\ODAC112021Xcopy_32bit\instantclient_11_2\oci.dll
\ODAC112021Xcopy_32bit\instantclient_11_2\orannzsb11.dll
\ODAC112021Xcopy_32bit\instantclient_11_2\oraocci11.dll
\ODAC112021Xcopy_32bit\instantclient_11_2\oraociei11.dll
\ODAC112021Xcopy_32bit\odp.net20\bin\OraOps11w.dll
\ODAC112021Xcopy_32bit\odp.net20\odp.net\bin\2.x\Oracle.DataAccess.dll
```

64-bit



```
\ODAC112021Xcopy_x64\instantclient_11_2\oci.dll  
\ODAC112021Xcopy_x64\instantclient_11_2\orannzsbb11.dll  
\ODAC112021Xcopy_x64\instantclient_11_2\oraocci11.dll  
\ODAC112021Xcopy_x64\instantclient_11_2\oraociei11.dll  
\ODAC112021Xcopy_x64\odp.net20\bin\OraOps11w.dll  
\ODAC112021Xcopy_x64\odp.net20\odp.net\bin\2.x\Oracle.DataAccess.dll
```

Driver/dialect selection

For Oracle to work with Enterprise Tester you must change the web.config file to let Enterprise Tester know that you are connecting to an Oracle database.

To do so, open the web.config file in a text editor and then change the following values from the defaults:

Original Value	New Value
<code><add key="migration.providerName" value="Migrator.Providers.SqlServer.SqlServer2005Dialect" /></code>	<code><add key="migration.providerName" value="Migrator.Providers.Oracle.OracleDialect" /></code>
<code><add key="connection.driver_class" value="NHibernate.Driver.SqlClientDriver" /></code>	<code><add key="connection.driver_class" value="NHibernate.Driver.OracleDataClientDriver" /></code>
<code><add key="dialect" value="NHibernate.Dialect.MsSql2005Dialect" /></code>	<code><add key="dialect" value="NHibernate.Dialect.Oracle10gDialect" /></code>

Connection string

Next you will need to setup the connection string (also in the web.config file)

The connection string shipped "out of the box" with Enterprise Tester will look like this:

```
<connectionStrings>
  <add name="Default" connectionString="Data Source=.;Initial
Catalog=EnterpriseTester;Integrated Security=True;MultipleActiveResultSets=true;" />
```

For ODP.Net there are a number of possibilities as to how the connection string can be formatted, here are options/examples (pulled from connectionstrings.com).

Using TNS

```
Data Source=TORCL;User Id=myUsername;Password=myPassword;
```

Using Integrated Security

```
Data Source=TORCL;Integrated Security=SSPI;
```

Using ODP.NET without tnsnames.ora

```
Data
Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=MyHost)(PORT
=MyPort)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));User
Id=myUsername;Password=myPassword;
```

Using the Easy Connect Naming Method (aka EZ Connect)

The easy connect naming method enables clients to connect to a database without any configuration.

```
Data Source=username/password@//myserver:1521/my.service.com;
```

Port 1521 is used if no port number is specified in the connection string.

Make sure that EZCONNECT is enabled in the sqlnet.ora file. NAMES.DIRECTORY_PATH=(TNSNAMES, EZCONNECT)

'//' in data source is optional and is there to enable URL style hostname values

- **Easy Connect Naming Method to connect to an Instance**

This one does not specify a service or a port.

```
Data Source=username/password@myserver//instancename;
```

- **Easy Connect Naming Method to connect to a dedicated server instance**

This one does not specify a service or a port.

```
Data Source=username/password@myserver/mysevice:dedicated/instancename;
```

Other server options: SHARED, POOLED (to use instead of DEDICATED). Dedicated is the default.



Appendixes

Appendix A - Application unable to load Oracle.DataAccess.dll

The above changes are normally enough to get connectivity to Oracle working, especially if you use the "EZ Connect" or "ODP.Net with TNSNames.ORA" methods.

But in some cases this may not be enough, and you will need to help Oracle client locate the unmanaged (native) .dll files for the application - this is normally required if when starting "Enterprise Tester you can an error indicating that "Oracle.DataAccess.dll" or one of it's dependencies could not be loaded".

To do so edit the web.config file and near the top, locate the end of the config sections element:

```
<configSections>
...
</configSections>
```

Once found, add a new section group just before the end and a new section afterwards called "oracle.dataaccess.client":

```
<sectionGroup name="oracle.dataaccess.client"
type="Oracle.DataAccess.Client.OracleSectionGroup, Oracle.DataAccess, Version=2.112.2.0">
  <section name="settings" type="Oracle.DataAccess.Client.SettingsSection,
Oracle.DataAccess, Version=2.112.2.0, Culture=neutral" />
  <section name="udtCustomTypeMappings"
type="Oracle.DataAccess.Client.UdtCustomTypeMappingsSection, Oracle.DataAccess,
Version=2.112.2.0" />
</sectionGroup>
</configSections>
<oracle.dataaccess.client>
  <settings>
    <add name="DllPath" value="C:\Program Files (x86)\Catch Limited\Enterprise
Tester\Web\bin\ "></add>
    <add name="FetchSize" value="65536"></add>
    <add name="PromotableTransaction" value="promotable"></add>
    <add name="StatementCacheSize" value="10"></add>
    <add name="TraceFileName" value="c:\temp\odpnet2.log"></add>
    <add name="TraceLevel" value="0"></add> <!-- 63 -->
    <add name="TraceOption" value="0"></add> <!-- 1 -->
  </settings>
</oracle.dataaccess.client>
</configuration>
```

Where the "DllPath" points to your native drivers i.e. oraoci11.dll, oci.dll etc.

Appendix B - Fixing tnsnames.ora resolution issues

One issue you can have after making these changes (depending on how your machine is configured) is that ODP.Net client cannot locate the tnsnames.ora file and it's entries - you can normally establish this is the problem if connecting via the method "ODP.Net without tnsnames.ora" works, but if using a connection string that requires a tnsnames.ora entry, then it can't connect.

This can be resolved by implementing one of the below fixes:

- By configuring ORA_HOME environment variable

The preferred fix is to set the Environment variable ORA_HOME with the location of the Oracle HOME directory which contains the correct tnsnames.ora file - normally you would have an Oracle home directory such as:

```
C:\app\admin\product\11.1.0\client_1
```

And your tnsnames.ora file would be located here:

```
C:\app\admin\product\11.1.0\client_1\network\admin\
```

So you would then go to Start -> Control Panel -> System -> Advanced System Settings -> Environment Variables.

Once the Environment Variables dialog is displayed, you would then add a new "System Variables" entry by clicking "New", and then enter "ORA_HOME" as the Variable name, and the Oracle home path as the Variable value i.e. C:\app\admin\product\11.1.0\client_1 from the example above.

After making these changes you may need to restart IIS for them to be applied.

- Using an ifile redirect

As an alternative to configuring ORA_HOME, you can use an ifile to direct Enterprise Tester to the location of the correct tnsnames.ora file:

You can do this by creating a tnsnames.ora file in the Bin folder of Enterprise Tester (in some cases you may actually need to create a couple of folders underneath the bin folder i.e. bin/Network/Admin) and then configuring that file with an ifile reference, pointing to the correct tnsnames.ora file:

```
ifile=c:\path_to_tnsnames.ora_file\tnsnames.ora
```

[More info on ifile is here](http://oracleprofessional.blogspot.com/2007/05/using-ifile-to-centralise-tnsnamesora.html) (http://oracleprofessional.blogspot.com/2007/05/using-ifile-to-centralise-tnsnamesora.html)