Oracle® Data Provider for .NET

Oracle TimesTen In-Memory Database Support User's Guide

12*c* Release 1 (12.1) for Windows

E38358-02

August 2013

Oracle Data Provider for .NET (ODP.NET) is an implementation of the Microsoft ADO.NET interface. ODP.NET support for Oracle TimesTen In-Memory Database (TimesTen) provides fast and efficient ADO.NET data access from .NET client applications to TimesTen databases.

This document covers only those aspects of ODP.NET that are specific to its use in a TimesTen environment and uses the term *ODP.NET for TimesTen* to refer to ODP.NET support for TimesTen.

For general ODP.NET and related API information, refer to *Oracle Data Provider for .NET Developer's Guide*.

You can use ODP.NET with any of the following TimesTen installations:

- TimesTen Data Manager only (for direct connections)
- TimesTen Client only (for client/server connections, assuming a TimesTen Data Manager instance and TimesTen Server instance are accessible elsewhere)
- TimesTen Data Manager with TimesTen Server and TimesTen Client (for either direct connections or client/server connections)

This document covers the following topics:

There is also a Documentation Accessibility section at the end of this document.

Initial considerations for ODP.NET in a TimesTen environment

This section discusses points you should be aware of before starting to use ODP.NET with TimesTen, covering the following topics:

- Related documents

Environments and TimesTen releases supported by ODP.NET

This revision of the document is for TimesTen support of the ODP.NET 12.1 release.

Note the following:

- You must have a TimesTen 11.2.2 release to use ODP.NET 12.1 with TimesTen. See "ODP.NET namespace and class support with TimesTen" on page 16 for details of supported namespaces and APIs. ODP.NET 12.1 is available in corresponding Oracle Database or Oracle Data Access Components (ODAC) releases.

  Note: TimesTen release 11.2.2.4.0 or higher is recommended.

- As of this release, ODP.NET for TimesTen can be used in the following environments:
  - ODP.NET for .NET Framework 2.0 with Microsoft .NET Framework 3.5 SP 1 or higher
  - ODP.NET for .NET Framework 4 with Microsoft .NET Framework 4 or 4.5
- ODP.NET for TimesTen can be used on all Microsoft Windows 32-bit and 64-bit platforms that support TimesTen. The 32-bit version of ODP.NET must be used with a 32-bit instance of the TimesTen database or TimesTen client. Likewise, the 64-bit version of ODP.NET must be used with a 64-bit instance of the TimesTen database or TimesTen client.

Support for .NET-related features

ODP.NET for TimesTen supports a subset of features currently available in ODP.NET for Oracle Database. In particular, as of this release, it supports the following features:

- ODP.NET connection pooling
- ODP.NET tracing

ODP.NET for TimesTen does not currently support these features:

- ADO.NET Entity Framework object relational mapper
- LINQ (Language-Integrated Query)

ODP.NET for TimesTen does not currently support interoperability with the following Oracle Database client components:

- Oracle Developer Tools for Visual Studio
- Oracle Database Extensions for .NET
- Oracle Providers for ASP.NET

Requirements and prerequisites for using ODP.NET with TimesTen

Note the following requirements to use ODP.NET for TimesTen:

- You must install TimesTen Data Manager or TimesTen Client or both on your system. TimesTen is not provided with ODP.NET or OCI.

- PL/SQL must be installed and enabled. It is installed by default during TimesTen installation. It is also enabled by default, or can be enabled through the first connection attribute setting `PLSQL=1` or the ODBC Data Source Administrator.
- ODP.NET 12.1 for TimesTen depends on Oracle Call Interface (OCI) support for TimesTen and requires the version of OCI that is provided with ODP.NET 12.1 releases, not the version provided with TimesTen.

  Also see "Post-installation path considerations" on page 7.

  Notes:

  - For reference, the OCI version provided with TimesTen is under the *`tt_install_dir`*`\ttoracle_home` directory, where *`tt_install_dir`* is the TimesTen installation root directory. Do *not* use this version for ODP.NET applications.
  - Of course there is no issue in using the TimesTen version of OCI for OCI or Pro*C/C++ programs that do not use ODP.NET.

- Requirements for the execution environment to use ODP.NET with Oracle Database apply to using ODP.NET with TimesTen as well. Refer to *Oracle Data Provider for .NET Developer's Guide* for information.

Related documents

Some of the preceding discussion refers to documents in the Oracle TimesTen In-Memory Database and Oracle Database documentation libraries.

TimesTen documentation is available on the product distribution media, and on the Oracle Technology Network at the following location.

`http://www.oracle.com/technetwork/products/timesten/documentation/`

Oracle documentation is also available on the Oracle Technology network, at the following location.

`http://www.oracle.com/pls/db121/homepage`

What's new

With this release, ODP.NET for TimesTen now supports the following features:

- Access to TimesTen `CLOB`, `NCLOB` and `BLOB` SQL column types: These SQL types can be accessed using the `GetOracleClob` and `GetOracleBlob` methods of `OracleDataReader` objects. For important additional information, refer to "Support for LOBs" on page 14.
- Associative arrays (formerly known as index-by tables or PL/SQL tables): These are supported as `IN`, `OUT`, or `IN OUT` bind parameters in TimesTen PL/SQL and can be used from an ODP.NET application in a TimesTen environment (as previously supported from OCI, Pro*C/C++, and JDBC applications in TimesTen). This enables arrays of data to be passed efficiently between an application and the database.
- .NET Framework 4.5 environment: Also refer to the "Environments and TimesTen releases supported by ODP.NET" on page 2.

Note: New features listed in the "What's New" section of *Oracle Data Provider for .NET Developer's Guide* apply to Oracle Database, not to TimesTen. (Regarding ODP.NET 12.1 new features for character data type support, however, refer to "Support for VARCHAR2, NVARCHAR2 and VARBINARY data types" on page 14.)

Getting started with ODP.NET

This section discusses the following topics to help you start using ODP.NET. Note that installation steps are not TimesTen-specific.

- Installing ODP.NET
- Post-installation path considerations
- Uninstalling ODP.NET
- Building an application for ODP.NET

Installing ODP.NET

This section covers the following installation:

- Install ODP.NET as part of Oracle Database 12.1
- Install ODP.NET as part of ODAC 12.1 for Windows, OUI version
- Install ODP.NET as part of ODAC 12.1 for Windows, XCopy version

The installation process for ODP.NET is independent of the TimesTen environment. Nothing is installed into the TimesTen installation directories.

Refer to *Oracle Data Provider for .NET Developer's Guide* for additional information about ODP.NET installation, including associated Windows registry entries.

Important:

- To use ODP.NET for TimesTen, ODP.NET should be installed on the same system as TimesTen Data Manager or TimesTen Client. See "TimesTen Installation" in *Oracle TimesTen In-Memory Database Installation Guide* for installation information. In addition, a TimesTen DSN must be configured. Refer to "Specifying Data Source Names to identify TimesTen databases" in *Oracle TimesTen In-Memory Database Operations Guide* for information about setting up a DSN.
- It is recommended, but not required (unless otherwise noted), to remove any previous versions of ODP.NET before installing a new version.
- After you have completed the installation steps, the location of ODP.NET binaries varies depending on your type of Oracle product installation and version of .NET. Consult *Oracle Data Provider for .NET Developer's Guide* and the ODP.NET README file for information.

Install ODP.NET as part of Oracle Database 12.1

Use these instructions to install ODP.NET as part of an Oracle Database 12.1 installation.

This installation uses the Oracle Universal Installer (OUI), as would be the case in any Oracle Database environment. Run it by executing its `setup.exe` file (in the `Disk1` location), then complete the following steps:

1. In the Select Installation Type dialog, typically choose the "Runtime" option, then choose **Next**.

   (There are four installation options: Instant Client, Runtime, Custom, and Administrator. The Runtime option provides the Oracle Client and ODP.NET. The Instant Client option does not provide ODP.NET. If you have reason to use a Custom installation, there is a check box for ODP.NET.)

2. In the resulting Download Software Updates dialog, you may choose the "Skip software updates" option at your discretion, as appropriate for your situation. Choose **Next** when you have finished.

3. In the resulting Select Product Languages dialog, use the default English language setting (or choose a desired language), then choose **Next**.

4. In the resulting Specify Oracle Home User dialog, choose from among "Use Existing User", "Create New User", or "Use Built-in Account" according to what best fits your needs.

5. In the resulting Specify Installation Location dialog, you can typically use the default values for the Oracle base path and software location. Alternatively:

   a. Enter the path (or browse) to your Oracle home location for the Oracle base path.

   b. Enter the path (or browse) to a desired directory location under the Oracle home location where the software files will be installed.

   Important: Paths cannot have spaces.

6. The installer performs prerequisite checks and display a Summary dialog. Choose **Install**.

Install ODP.NET as part of ODAC 12.1 for Windows, OUI version

Use these instructions to install ODP.NET as part of the Oracle Universal Installer (OUI) version of Oracle Data Access Components (ODAC) 12.1 releases.

Run OUI by executing its `setup.exe` file, which you can access in the staging directory into which you unzip the ODAC ZIP file for OUI installation, then complete the following steps:

1. In the OUI welcome page, choose **Next**.

2. In the page to select a product to install, choose "Oracle Data Access Components for Oracle Client 12.1.*x.x.x*" (the appropriate point release number is indicated). Then choose **Next**.

3. In the page for the installation location, you can either use the default value or specify a desired alternative location. Then choose **Next**.

4. In the page to choose available product components, "Oracle Data Access Components for Oracle Client 12.1.0.1.0" is selected by default and cannot be deselected (given the selection of that product in step 2). Underneath that, confirm that Oracle Data Provider for .NET is selected (default). Also confirm that Oracle Instant Client, required for ODP.NET, is selected (default). All other components are optional for ODP.NET. Then choose **Next**.

   Note: *Oracle Providers for ASP.NET is not relevant to TimesTen*. You can deselect it unless you need it for other purposes. Note that if it is selected, you will see a page (not covered in these instructions) for running a set of SQL scripts.

5. In the page for database connection configuration, you are prompted to specify a database connection entry (alias) that will be entered in the `tnsnames.ora` file, and the location of that file is indicated. Leave this page as is and choose **Next**. (Define a database connection later, as discussed in "Configuring TimesTen connections for an ODP.NET application" on page 8.)
6. Check the summary page, then choose **Install**. The installation page will indicate progress.
7. After installation, there is an end page. You can choose **Exit**.

Install ODP.NET as part of ODAC 12.1 for Windows, XCopy version

Use these instructions to install ODP.NET as part of the Oracle XCopy version of Oracle Data Access Components (ODAC) 12.1 releases. The installation process is the same for the 64-bit version as for the 32-bit version.

XCopy provides system administrators with an ODP.NET client that is smaller than the standard ODP.NET client and can be configured more easily, with finer-grained control than OUI offers. This makes it more convenient for production deployments to large numbers of computers, and simplifies the embedding of ODP.NET in customized deployment packages.

This installation does *not* use the Oracle Universal Installer. Instead, run the installation by executing the `install.bat` batch file, which you can access in the installation directory into which you unzip the ODAC ZIP file for XCopy installation.

This is a summary of the installation instructions. For further details, refer to `readme.htm`, which is also located in the installation directory.

Important: The `readme.htm` file emphasizes the following points.

- Do not install XCopy over an existing OUI-based Oracle home installation.
- If you do multiple ODAC product installations to the same directory, specify the same Oracle home name each time.
- By default, ODAC products and dependencies are installed without a check to see if there are newer product versions already installed.

Execute `install.bat` to specify the desired ODAC products to install. For example, assuming `C:\oracle\odac` is your installation directory and `odachome` is your Oracle

home name for ODAC, use the following command to install the client with only ODP.NET for .NET 2.0 libraries:

```
install.bat odp.net2 C:\oracle\odac odachome
```

Or use this command to install the client with only ODP.NET for .NET 4 libraries:

```
install.bat odp.net4 C:\oracle\odac odachome
```

Alternatively, use the following command to install the client with all ODAC products:

```
install.bat all C:\oracle\odac odachome
```

Post-installation path considerations

In a TimesTen environment, ODP.NET finds and uses the appropriate version of OCI; namely, the Oracle Client version and not the TimesTen Instant Client version. In addition, check the following for your path:

1. Confirm that the `PATH` setting has the location of the TimesTen shared libraries at *`tt_install_dir`*`\bin`, where *`tt_install_dir`* is the TimesTen installation root directory. This should follow any other Oracle directories in the path.
2. For an XCopy installation, add your ODAC installation directory and ODAC installation `bin` directory to the `PATH` setting, preceding any other Oracle directories, including TimesTen directories. For example, if `C:\oracle\odac` is the installation directory:

   ```
   set PATH=C:\oracle\odac;C:\oracle\odac\bin;%PATH%
   ```

> **Note:** Refer to the ODP.NET README file for any further information about setting up ODP.NET.

Uninstalling ODP.NET

For information about uninstalling Oracle Database products, including ODP.NET, refer to "Removing Oracle Database Software" in *Oracle Database Installation Guide for Microsoft Windows*.

To uninstall an OUI installation, run `setup.exe` again (refer to "Install ODP.NET as part of ODAC 12.1 for Windows, OUI version" on page 5). In the OUI welcome page, choose **Deinstall Products**. In the resulting Inventory dialog, select the product or products to uninstall, then choose **Remove**. Then choose **Yes** in the Confirmation and Warning dialogs. Close the Inventory dialog once the products have been uninstalled.

To uninstall an XCopy installation, execute the `uninstall.bat` batch file from your ODP.NET installation directory (refer to "Install ODP.NET as part of ODAC 12.1 for Windows, XCopy version" on page 6), specifying the product to uninstall (or all products) and the Oracle home name for ODAC products. For example, to uninstall a client with ODP.NET for .NET 2.0 libraries, assuming the Oracle home name is `odachome`:

```
uninstall.bat odp.net2 odachome
```

Or to uninstall all ODAC products:

```
uninstall.bat all odachome
```

Building an application for ODP.NET

You can use the Visual Studio IDE to build your application, or you can use the `csc.exe` command-line compiler executed from the Visual Studio command prompt. The following example uses `csc.exe` (where the input is all one command line):

```
C:\Temp> csc /out:myapp.exe /reference:C:\app\mydir\path\
Oracle.DataAccess.dll myapp.cs

Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.3053
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.
```

The location of the `Oracle.DataAccess.dll` assembly and dependent libraries is according to your type of Oracle product installation and version of .NET. Refer to *Oracle Data Provider for .NET Developer's Guide* and the ODP.NET README file for information.

> **Note:** Visual Studio is not a runtime requirement of ODP.NET for TimesTen, but you would need a .NET compiler, such as the C# compiler that comes with Visual Studio, to develop applications.

Configuring TimesTen connections for an ODP.NET application

ODP.NET for TimesTen supports multiple simultaneous connections to TimesTen and Oracle databases. Existing applications written for the ODP.NET interface can access TimesTen with a minimal set of changes to their application code.

In a TimesTen environment, ODP.NET uses OCI to interact with the TimesTen database. Therefore, an ODP.NET application can connect to TimesTen using either the `tnsnames` or the *easy connect* naming method, as with Oracle Database. See "Configuring Naming Methods" in *Oracle Database Net Services Administrator's Guide* for information about the `tnsnames` and easy connect naming methods beyond what is provided below.

This section covers the following topics:

- Using the tnsnames naming method to connect
- Using the easy connect naming method to connect
- Configuring whether to use tnsnames or easy connect naming method
- Setting TimesTen connection attributes in ODP.NET connection strings

Notes:

- TimesTen does not support distributed transactions through OCI. Therefore, an ODP.NET application cannot use distributed transactions in a TimesTen connection.
- ODP.NET for TimesTen does not support global runtime load balancing (a feature for Oracle RAC databases) and therefore does not support the connection string attribute setting "`Load Balancing=true`".
- Error messages associated with connections to TimesTen from an ODP.NET application are based on TimesTen OCI error message mapping. TimesTen OCI errors are propagated to the ODP.NET application as `OracleException` objects. (Also see "OCI error reporting" in *Oracle TimesTen In-Memory Database C Developer's Guide*.)

Using the tnsnames naming method to connect

TimesTen supports `tnsnames` syntax. You can use a TimesTen `tnsnames.ora` entry in the same way you would use an Oracle `tnsnames.ora` entry.

The syntax of a TimesTen entry in the `tnsnames.ora` file is as follows:

```
tns_entry = (DESCRIPTION =
               (CONNECT_DATA =
                  (SERVICE_NAME = dsn)
                  (SERVER = timesten_direct | timesten_client)))
```

Where *`tns_entry`* is an arbitrary TNS name you assign to the entry. Note the following:

- `DESCRIPTION` and `CONNECT_DATA` are required as shown.
- For `SERVICE_NAME`, *`dsn`* must be a TimesTen data source name (DSN) that is defined in the ODBC Data Source Administrator and is visible to the user running the ODP.NET application.
- For `SERVER`, `timesten_direct` specifies a direct connection to a TimesTen database, while `timesten_client` specifies a client/server connection. If you specify `timesten_direct`, then *`dsn`* must be a TimesTen Data Manager DSN. If you specify `timesten_client`, then *`dsn`* must be a TimesTen Client DSN.

The following is a sample `tnsnames.ora` entry for a direct connection to the TimesTen database referenced by the DSN `my_dsn`:

```
my_tnsname = (DESCRIPTION =
                (CONNECT_DATA =
                   (SERVICE_NAME = my_dsn)
                   (SERVER = timesten_direct)))
```

To connect as user `scott` with password `tiger` to the `my_dsn` TimesTen database that is referenced by the `my_tnsname` entry in the `tnsnames.ora` file, specify the following connection string in your ODP.NET application:

```
"User Id=scott;Password=tiger;Data Source=my_tnsname"
```

To connect as the current operating system user to `my_dsn` that is referenced by the `my_tnsname` entry in the `tnsnames.ora` file, specify the following connection string in

your ODP.NET application. The current operating system user must be either the TimesTen instance administrator or a defined TimesTen external user.

```
"User Id=/;Data Source=my_tnsname"
```

Using the easy connect naming method to connect

TimesTen supports easy connect syntax, which allows connections to be made without configuring a `tnsnames.ora` entry. The syntax of a TimesTen easy connect string is as follows:

host/*service_name*:*server*

Note the following:

- A host name must be specified to satisfy easy connect syntax, but is otherwise ignored by TimesTen. The name `localhost` is typically used by convention.
- For *service_name*, specify a TimesTen DSN that is defined in the ODBC Data Source Administrator and is visible to the user running the ODP.NET application.
- For *server*, `timesten_direct` specifies a direct connection to a TimesTen database, while `timesten_client` specifies a client/server connection. If you specify `timesten_direct`, then *service_name* must be a TimesTen Data Manager DSN. If you specify `timesten_client`, then *service_name* must be a TimesTen Client DSN.

To establish a direct connection as user `scott` with password `tiger` to the TimesTen database referenced by the `my_dsn` DSN, specify the following connection string in your ODP.NET application:

```
"User Id=scott;Password=tiger;Data Source=localhost/my_dsn:timesten_direct"
```

To establish a direct connection as the current operating system user to the TimesTen database referenced by `my_dsn`, specify the following connection string in your ODP.NET application. The current operating system user must be either the TimesTen instance administrator or a defined TimesTen external user.

```
"User Id=/;Data Source=localhost/my_dsn:timesten_direct"
```

Configuring whether to use tnsnames or easy connect naming method

If a `sqlnet.ora` file is present, it specifies the naming methods to be tried and the order in which to try them. ODP.NET looks for a `sqlnet.ora` file with the following precedence:

1. If the `TNS_ADMIN` environment variable has been set, ODP.NET looks in that specified location.
2. If `TNS_ADMIN` has not been set, ODP.NET looks in the Oracle Database default location, as noted in "Parameters for the sqlnet.ora File" in *Oracle Database Net Services Reference*.

If `sqlnet.ora` is found, you can use only naming methods that are indicated there. If `sqlnet.ora` is not found, you can use either the `tnsnames` or easy connect naming method.

In TimesTen, sample copies of the `tnsnames.ora` and `sqlnet.ora` files are in the *tt_install_dir*`\network\admin\samples` directory, where *tt_install_dir* is the TimesTen installation root directory. The following is the `sqlnet.ora` file that TimesTen provides, which supports both the `tnsnames` naming method and the easy connect naming method.

```
# To use ezconnect syntax or tnsnames, the following entries must be
# included in the sqlnet.ora configuration.
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

With this setting, ODP.NET first looks for `tnsnames` syntax in your connection strings. If it cannot find `tnsnames` syntax, it looks for easy connect strings.

> **Important:** Oracle Database network libraries are provided with ODP.NET. In a TimesTen environment, ODP.NET does *not* use the copy of the Oracle Database network libraries provided with the Instant Client shipped with TimesTen. (That location, for reference, is *tt_install_dir*`\ttoracle_home\instantclient_11_2` for the Oracle Database 11.2 Instant Client shipped with TimesTen 11.2.2 releases.)

Setting TimesTen connection attributes in ODP.NET connection strings

You can set TimesTen connection attributes within the `Password` setting of your ODP.NET connection string, with syntax as follows:

- Components of the `Password` setting, including the password setting itself and any TimesTen connection attribute settings, are delimited by semi-colons.
- Whenever the `Password` setting has semi-colons, the entire setting must be quoted.
- Because the ODP.NET connection string as a whole is quoted, the begin quotation mark and end quotation mark of the `Password` setting must each be preceded by the "`\`" escape character.

The following example specifies `lion` as the password for user `scott` in TimesTen. It also sets the TimesTen `OraclePWD` connection attribute, which specifies the password `tiger` for user `scott` in Oracle Database, for use of Oracle In-Memory Database Cache (IMDB Cache).

```
"Data Source=mysource;User Id=scott;Password=\"lion;OraclePwd=tiger\"";
```

The next example again specifies `lion` as the password for `scott` in TimesTen. This time, it sets the TimesTen `OracleNetServiceName` connection attribute as well as the `OraclePWD` connection attribute. `OracleNetServiceName` specifies the Oracle ID in Oracle Database, with the `OraclePWD` setting specifying the corresponding password `tiger`. Finally, this example sets the TimesTen passthrough level to 1.

```
"Data Source=mysource;User ID=scott;Password=\"lion;OraclePwd=tiger;
OracleNetServiceName=mytest-pc.example.com;passthrough=1\"";
```

(For general information about TimesTen connection attributes, refer to "Connection Attributes" in *Oracle TimesTen In-Memory Database Reference*. For information about IMDB Cache and passthrough levels, refer to *Oracle In-Memory Database Cache User's Guide*.)

> **Note:** As always, you can also set TimesTen connection attributes in your TimesTen DSN definition in ODBC Data Source Administrator, as shown in "Managing TimesTen Databases" in *Oracle TimesTen In-Memory Database Operations Guide*. This is not secure, however, so is not advisable for password settings such as the `OraclePWD` attribute.

Testing your ODP.NET installation with TimesTen

Perform the following steps to test ODP.NET with TimesTen in a .NET environment.

You must have a TimesTen installation, including the Quick Start sample programs, to perform these tests. These instructions also assume you have Visual Studio.

1. Where *tt_install_dir* is the TimesTen installation root directory, execute the `build_sampledb.bat` script from the *tt_install_dir*`\quickstart\sample_scripts\createdb\` directory. This creates a TimesTen database, `sampledb_1122`, with users and objects.
2. Copy the ODP.NET sample program `DemoODP.cs` to your system. It is located in the following TimesTen Quick Start directory:

   tt_install_dir`\quickstart\sample_code\odp.net\`

3. Create a `tnsnames.ora` file that contains the following:

   ```
   SAMPLEDB_1122 =(DESCRIPTION=(CONNECT_DATA =
      (SERVICE_NAME = SAMPLEDB_1122)(SERVER = timesten_direct)))
   ```

4. Open Visual Studio Command Prompt and set the environment variable `TNS_ADMIN` to specify the location of the `tnsnames.ora` file you created. For example:

   ```
   >set TNS_ADMIN=c:\mytnsdir\sqlnet
   ```

5. Navigate to the directory where `DemoODP.cs` was placed and compile the `DemoODP` program. For example:

   ```
   csc /out:DemoODP.exe /reference:C:\path\Oracle.DataAccess.dll DemoODP.cs
   ```

   Notes:

   - The location of the `Oracle.DataAccess.dll` assembly and dependent libraries is according to your type of Oracle product installation and version of .NET. Refer to *Oracle Data Provider for .NET Developer's Guide* and the ODP.NET README file for information.
   - The name of the TimesTen sample database in TimesTen 11.2.2 releases is `sampledb_1122`.

6. Execute `DemoODP` as follows. (The database name, user name, and password are determined automatically during execution of `build_sampledb.bat`.)

   ```
   DemoODP -db sampledb_1122 -user appuser -passwd welcome1
   ```

This should produce the following output:

```
Start Test
The employee who got the 10% pay raise was CLARK

Employees in department #50:
7944, ITMGR, MANAGER, 7839, 10/08/2010 10:34:20 AM, 2500, <NULL>, 50
7945, DVLPR1, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7946, DVLPR2, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7947, DVLPR3, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7948, DVLPR4, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
```

```
7949, DVLPR5, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7950, DVLPR6, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7951, DVLPR7, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7952, DVLPR8, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7953, DVLPR9, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7954, DVLPR10, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
Test finished
```

Development considerations for ODP.NET with TimesTen

This section discusses points to be aware of when developing applications to use ODP.NET in a TimesTen environment, covering the following topics:

- Effects of transaction commits on result sets and REF CURSORs
- Support for TimesTen built-in procedures
- Support for VARCHAR2, NVARCHAR2 and VARBINARY data types
- Support for LOBs
- Known issues and limitations
- Troubleshooting
- The ttSrcScan utility

Effects of transaction commits on result sets and REF CURSORs

When processing result sets generated from executing statements and creating REF CURSORs, the behavior when transactions in TimesTen connections are committed differs from that when transactions in Oracle Database connections are committed. When a transaction is committed in TimesTen while a result set of an `OracleDataReader` object is open, the result set is closed automatically, unlike in an Oracle database. This applies to explicit commits, autocommit, and implicit commits.

In TimesTen, an implicit commit occurs after a DDL statement. In ODP.NET, an implicit commit also occurs when an `OracleCommand` object is executed without there first being an `OracleTransaction` object instantiated from the command connection. An explicit commit occurs when the `Commit` method is called on an `OracleTransaction` object. In either case, if a commit occurs in a TimesTen connection before a result set that is open in the transaction is completely processed, the "`Function sequence error`" exception may be thrown.

This difference in behavior is likely to occur when the execution of an `OracleCommand` object is interleaved with the processing of a result set associated with another `OracleCommand` object. To avoid the "`Function sequence error`" exception, the execution and processing of a result set should be contained exclusively within the context of an `OracleTransaction` object. This prevents a commit from occurring before all rows of the result set are retrieved.

The occurrence of a "`Function sequence error`" exception may depend on the value of the `FetchSize` property of an `OracleCommand`, `OracleRefCursor` or `OracleDataReader` object. If the `FetchSize` property is not explicitly set or if it is set to a large value, then many rows may be fetched by the application before the "`Function sequence error`" exception is thrown.

Support for TimesTen built-in procedures

You can call TimesTen built-in procedures directly from TimesTen OCI only for built-ins that do not return a result set. Therefore, this restriction also applies to ODP.NET for TimesTen.

Use an `OracleCommand` instance to call a built-in, as in the following example. This assumes an `OracleConnection` instance `conn` with a connection to TimesTen has been established. Call the `Dispose` method to free resources when you have finished using the `OracleCommand` instance.

```
// switching to passthrough 1 mode using ttOptSetFlag built-in function
string switchModeStmt = "call ttOptSetFlag('passthrough', 1)";
OracleCommand switchCmd = new OracleCommand(switchModeStmt, conn);
switchCmd.CommandType = CommandType.Text;
switchCmd.ExecuteNonQuery();
switchCmd.Dispose();
```

For built-in procedures that do return a result set, the result set would not be accessible directly through ODP.NET. However, you could access it as an `OUT` parameter if you call the built-in from PL/SQL. Here is an example:

```
int passThruValue = -1;
OracleCommand cmd = conn.CreateCommand();
cmd.CommandText = "declare v_name varchar2(255); begin execute immediate
                  'call ttOptGetFlag(''passthrough'')' into v_name, :rc1; end;";
cmd.Parameters.Add("rc1", OracleDbType.Int32, -1, ParameterDirection.Output);
cmd.ExecuteNonQuery();
passThruValue = Convert.ToInt32(cmd.Parameters[0].Value.ToString());
cmd.Parameters.Clear();
cmd.Dispose();
```

Support for VARCHAR2, NVARCHAR2 and VARBINARY data types

TimesTen `VARCHAR2`, `NVARCHAR2` and `VARBINARY` types support a maximum of 4 MB of data. ODP.NET for TimesTen supports the transfer of the maximum amount of data for these types (and all other TimesTen SQL types).

Note: ODP.NET 12.1, when used outside of a TimesTen environment, has a 32 KB size limit for character data, increased from a 4 KB limit in previous releases.

Support for LOBs

TimesTen LOB support is limited to the LOB access methods associated with the default 0 (zero) setting of the `InitialLobFetchSize` property of the `OracleDataReader` object. If this property is changed to another value then TimesTen ignores such changes, assuming it retains the 0 setting.

Also refer to "OracleDataReader class support" on page 23.

Known issues and limitations

This section discusses limitations that are known as of release time.

- The self-tuning statement cache disables itself if it detects that not enough memory is available for its operations. Note that when an application uses a direct (as

opposed to client/server) connection to TimesTen, the entire database is loaded into memory, thereby reducing memory available for the statement cache and making this behavior more likely.

- Binding variables using `NCHAR` or `NVARCHAR` PL/SQL associative array types in an application using ODP.NET with TimesTen currently fails. For example, consider the following type declaration:

  ```
  TYPE MYTYPE IS TABLE OF NCHAR(20) INDEX BY BINARY_INTEGER;
  ```

  Attempting to use this type as follows will result in the error "PLS-00418: array bind type must match PL/SQL table row type":

  ```
  OracleParameter Param1 = cmd.Parameters.Add("1", OracleDbType.NChar);
  Param1.Direction = ParameterDirection.Output;
  Param1.CollectionType = OracleCollectionType.PLSQLAssociativeArray;
  ```

  The workaround is to use equivalent PL/SQL associative array types such as `CHAR` or `VARCHAR2`.

Troubleshooting

This section discusses solutions for various exceptions you may encounter.

- **Exception "ORA-44818: PL/SQL feature not installed in this TimesTen database"**

  If PL/SQL is disabled for the TimesTen database (it is enabled by default), enable it by setting the `PLSQL` connection attribute to 1 in the ODBC Data Source Administrator. ODP.NET applications cannot connect to TimesTen databases that are not PL/SQL-enabled.

- **Exception "TimesTen database version is less than the minimum required version of 11.2.2.0.0"**

  Change the service name in the `tnsname.ora` entry or easy connect string to reference a TimesTen 11.2.2 database. ODP.NET 12.1 applications cannot connect to TimesTen database versions prior to 11.2.2.

- **Exception "ORA-12154: TNS: Could not resolve the connect identifier specified"** or **"ORA-12541: TNS: No listener"**

  To connect to a TimesTen database from an ODP.NET application, the `Data Source` attribute in the ODP.NET connection string must be set either to the TNS name of a TimesTen entry in the `tnsnames.ora` file or to a TimesTen easy connect string.

  If the `tnsnames` naming method is used to connect, verify that an entry in the `tnsnames.ora` file is associated with a TimesTen DSN. Also verify that the `TNS_ADMIN` environment variable is set to the directory where the `tnsnames.ora` file is located.

  If the easy connect naming method is used to connect, verify that *`service_name`* is set to a TimesTen DSN and that *`server`* is set to either `timesten_direct` or `timesten_client`, depending on whether the DSN configures a direct connection or a client/server connection.

- **Exception "ORA-29158: Unable to open library"**

  If you are connecting to a TimesTen database, verify either that the entry in the `tnsnames.ora` file is associated with a TimesTen DSN or that *`service_name`* in the easy connect string is set to a TimesTen DSN.

This error may also occur due to a path issue, if ODP.NET cannot find the TimesTen ODBC driver, which is located in the TimesTen *tt_install_dir*\bin directory. (Also see the next troubleshooting item.)

- **Exception "ORA-29159: Unable to read library"**

  In addition to the steps for ORA-29158 above, verify that the server setting in the tnsnames.ora file entry or easy connect string is timesten_direct or timesten_client, as appropriate for the type of TimesTen DSN.

  See "Connecting to a TimesTen database from OCI" in *Oracle TimesTen In-Memory Database C Developer's Guide* for information about tnsnames.ora and easy connect.

- **Exception "The application has failed to start because ttcommon*xxxx*.dll was not found. Re-installing the application may fix the problem"**

  This indicates that the location of the TimesTen shared libraries at *tt_install_dir*\bin is not in the PATH environment variable setting.

  Note: Instead of "*xxxx*", the TimesTen release number is indicated. In TimesTen 11.2.2 releases the file name is ttcommon1122.dll.

The ttSrcScan utility

If you have an existing ODP.NET application and want to see whether it uses ODP.NET features that TimesTen does not support, you can use the ttSrcScan command line utility to scan your program for unsupported functions, types, type codes, attributes, modes, and constants. This is a standalone utility that can be run without TimesTen or Oracle Database being installed and runs on any platform supported by TimesTen. It reads source code files as input and creates HTML and text files as output. If the utility finds unsupported items, then they are logged and alternatives are suggested. You can find the ttSrcScan executable in the quickstart/sample_util directory in your TimesTen installation.

Specify an input file or directory for the program to be scanned and an output directory for the ttSrcScan reports. Other options are available as well. See the README file in the sample_util directory for information.

ODP.NET namespace and class support with TimesTen

This reference section documents support for ODP.NET namespaces and classes in a TimesTen environment.

ODP.NET implements the classes, enumerations, interfaces, delegates, and structures of the Oracle.DataAccess.Client and Oracle.DataAccess.Types namespaces. The Oracle.DataAccess.Client namespace contains implementations of core ADO.NET classes, enumerations for ODP.NET, and ODP.NET-specific classes. The Oracle.DataAccess.Types namespace provides classes, structures, and exceptions for Oracle Database native types that can be used with ODP.NET. See *Oracle Data Provider for .NET Developer's Guide* for information about these namespaces beyond what is provided below. You must have access to them in your program as follows:

```
using Oracle.DataAccess.Client;
using Oracle.DataAccess.Types;
```

The following sections list TimesTen support for the ODP.NET classes, enumerations and types of the `Oracle.DataAccess.Client` and `Oracle.DataAccess.Types` namespaces that are documented for ODP.NET 12.1 releases:

- Oracle.DataAccess.Client namespace support
- Oracle.DataAccess.Types namespace support

> **Note:** When connecting to a TimesTen database from an ODP.NET application, your application can use only ODP.NET features that correspond to features that TimesTen supports. This is reflected in what is supported for the namespaces discussed here.
>
> For example, you cannot use Oracle Streams Advanced Queueing because TimesTen does not support this feature. `OracleException` objects are thrown when you attempt to use ODP.NET features that are not supported by TimesTen. These exceptions are based on corresponding TimesTen OCI error messages.

Oracle.DataAccess.Client namespace support

The following tables list support for delegates, classes, and enumerations of the `Oracle.DataAccess.Client` namespace.

Table 1 Oracle.DataAccess.Client namespace delegate support

Delegate Name	Supported?
`OnChangeEventHandler`	No
`OracleAQMessageAvailableEventHandler`	No
`OracleFailoverEventHandler`	No
`OracleHAEventHandler`	No
`OracleInfoMessageEventHandler`	Yes
`OracleRowsCopiedEventHandler`	No
`OracleRowUpdatedEventHandler`	Yes
`OracleRowUpdatingEventHandler`	Yes

Table 2 Oracle.DataAccess.Client namespace class support

Class Name	Supported?	Notes
`OracleAQAgent`	No	
`OracleAQDequeueOptions`	No	
`OracleAQEnqueueOptions`	No	
`OracleAQMessage`	No	
`OracleAQMessageAvailableEventArgs`	No	
`OracleAQQueue`	No	
`OracleBulkCopy`	No	
`OracleBulkCopyColumnMapping`	No	
`OracleBulkCopyColumnMappingCollection`	No	

Table 2 (Cont.) Oracle.DataAccess.Client namespace class support

Class Name	Supported?	Notes
OracleClientFactory	Yes	
OracleCommand	Yes	See "OracleCommand class support" on page 20 for information about TimesTen support for properties and public methods of this class.
OracleCommandBuilder	Yes	
OracleConnection	Yes	See "OracleConnection class support" on page 21 for information about TimesTen support for properties and public methods of this class.
OracleConnectionStringBuilder	Yes	
OracleDataAdapter	Yes	The IdentityInsert and IdentityUpdate properties are not supported.
OracleDatabase	No	
OracleDataReader	Yes	See "OracleDataReader class support" on page 23 for information about TimesTen support for properties and public methods of this class.
OracleDataSourceEnumerator	Yes	
OracleDependency	No	
OracleError	Yes	
OracleErrorCollection	Yes	
OracleException	Yes	
OracleFailoverEventArgs	No	
OracleGlobalization	No	
OracleHAEventArgs	No	
OracleInfoMessageEventArgs	Yes	
OracleLogicalTransactionStatus	No	
OracleNotificationEventArgs	No	
OracleNotificationRequest	No	
OracleParameter	Yes	
OracleParameterCollection	Yes	
OraclePermission	Yes	
OraclePermissionAttribute	Yes	
OracleRowsCopiedEventArgs	No	
OracleRowUpdatedEventArgs	Yes	
OracleRowUpdatingEventArgs	Yes	

Table 2 (Cont.) Oracle.DataAccess.Client namespace class support

Class Name	Supported?	Notes
OracleTransaction	Yes	See "OracleTransaction class support" on page 25 for information about TimesTen support for properties and public methods of this class.
OracleXmlQueryProperties	No	
OracleXmlSaveProperties	No	

Table 3 Oracle.DataAccess.Client namespace enumeration support

Enumeration Name	Supported?
FailoverEvent	No
FailoverReturnCode	No
FailoverType	No
OracleAQDequeueMode	No
OracleAQMessageDeliveryMode	No
OracleAQMessageState	No
OracleAQMessageType	No
OracleAQNavigationMode	No
OracleAQNotificationGroupingType	No
OracleAQNotificationType	No
OracleAQVisibilityMode	No
OracleBulkCopyOptions	No
OracleCollectionType	No
OracleDBShutdownMode	No
OracleDBStartupMode	No
OracleDbType	Yes
OracleHAEventSource	No
OracleHAEventStatus	No
OracleIdentityType	No
OracleNotificationInfo	No
OracleNotificationSource	No
OracleNotificationType	No
OracleParameterStatus	Yes
OracleRowidInfo	No
OracleXmlCommandType	No

The rest of this section presents the following:

- OracleCommand class support

- OracleConnection class support
- OracleDataReader class support
- OracleTransaction class support

OracleCommand class support

The following tables list support for properties and methods of the `OracleCommand` class.

Table 4 OracleCommand class property support

Property Name	Supported?
`AddRowid`	No
`AddToStatementCache`	Yes
`ArrayBindCount`	Yes
`ArrayBindRowsAffected`	No
`BindByName`	No
`CommandText`	Yes
`CommandTimeout`	No
`CommandType`	Yes
`Connection`	Yes
`DesignTimeVisible`	No
`FetchSize`	Yes
`ImpliedRefCursors`	Yes (see note)
`InitialLOBFetchSize`	No (see note)
`InitialLONGFetchSize`	No
`Notification`	No
`NotificationAutoEnlist`	No
`Parameters`	Yes
`RowSize`	Yes
`Transaction`	Yes
`UpdatedRowSource`	Yes
`XmlCommandType`	No
`XmlQueryProperties`	No
`XmlSaveProperties`	No

Notes:

- While TimesTen supports the `ImpliedRefCursors` property, its use is complementary to the ADO.NET Entity Framework, which TimesTen does not support.
- ODP.NET for TimesTen does not support use of the `InitialLOBFetchSize` property. Changing its value has no effect. It is always effectively set to the default value of 0 (zero).

Table 5 OracleCommand class method support

Method Name	Supported?
`Cancel`	No
`Clone`	Yes
`CreateParameter`	Yes
`ExecuteNonQuery`	Yes
`ExecuteReader`	Yes
`ExecuteScalar`	Yes
`ExecuteStream`	No
`ExecuteToStream`	No
`ExecuteXmlReader`	No

OracleConnection class support

The following tables list support for properties and methods of the `OracleConnection` class.

Table 6 OracleConnection class property support

Property Name	Supported?
`ActionName`	No
`ClientId`	No
`ClientInfo`	No
`ConnectionString`	Yes
`ConnectionTimeout`	Yes
`Database`	No
`DatabaseDomainName`	No
`DatabaseName`	No
`DataSource`	Yes
`HostName`	No
`InstanceName`	No
`IsAvailable` (static property)	No
`LogicalTransactionID`	No
`ModuleName`	No

Table 6 (Cont.) OracleConnection class property support

Property Name	Supported?
ServerVersion	Yes
ServiceName	No
State	Yes
StatementCacheSize	Yes

Table 7 OracleConnection class event support

Method Name	Supported?
Failover	No
HAEvent	No
InfoMessage	No
StateChange	Yes

Table 8 OracleConnection class method support

Method Name	Supported?	Notes
BeginTransaction	Yes	
ChangeDatabase	No	
ClearAllPools (static method)	Yes	
ClearPool (static method)	Yes	
Clone	Yes	
Close	Yes	
CreateCommand	Yes	
EnlistDistributedTransaction	No	TimesTen does not support distributed transactions through OCI. Therefore, an ODP.NET application cannot use distributed transactions in a TimesTen connection.
EnlistTransaction	No	See EnlistDistributedTransaction.
FlushCache	No	
GetLogicalTransactionStatus	No	
GetSchema	Yes	Returns metadata collections of tables, columns, users, and other objects that allow application developers to discover and enumerate database information. This information is specific to TimesTen and may differ from corresponding metadata collections returned from Oracle Database. For example, TimesTen does not support the JavaClasses and XMLSchemas metadata collections because these object types are not supported by TimesTen.
GetSessionInfo	Yes	
Open	Yes	

Table 8 (Cont.) OracleConnection class method support

Method Name	Supported?	Notes
OpenWithNewPassword	No	
PurgeStatementCache	Yes	
SetSessionInfo	No	

OracleConnectionType, an enumeration and public OracleConnection class property, allows an ODP.NET application to determine whether a particular connection object is associated with a TimesTen connection, an Oracle Database connection, or no physical connection at all. The property has the following signature:

```
public OracleConnectionType ConnectionType
```

It returns one of the following values from the OracleConnectionType enumeration:

```
OracleConnectionType.Undefined: No connection is associated with the
OracleConnection object

OracleConnectionType.Oracle: The OracleConnection object is associated with an
Oracle database

OracleConnectionType.TimesTen: The OracleConnection object is associated with a
TimesTen database
```

OracleDataReader class support

The following tables list support for properties and methods of the OracleDataReader class.

Table 9 OracleDataReader class property support

Property Name	Supported?
Depth	Yes
FetchSize	Yes
FieldCount	Yes
HasRows	Yes
HiddenFieldCount	Yes
IsClosed	Yes
Item	Yes
InitialLOBFetchSize	No
InitialLONGFetchSize	No
RecordsAffected	No
RowSize	Yes
VisibleFieldCount	Yes

Note: ODP.NET for TimesTen does not support use of the InitialLOBFetchSize property. Changing its value has no effect. It is always effectively set to the default value of 0 (zero).

Table 10 OracleDataReader class method support

Method Name	Supported?
Close	Yes
Dispose	Yes
GetBoolean	No
GetByte	Yes
GetBytes	Yes
GetChar	Yes
GetChars	Yes
GetData	No
GetDataTypeName	Yes
GetDateTime	Yes
GetDecimal	Yes
GetDouble	Yes
GetEnumerator	No
GetFieldType	Yes
GetFloat	Yes
GetGuid	No
GetInt16	Yes
GetInt32	Yes
GetInt64	Yes
GetName	Yes
GetOracleBFile	No
GetOracleBinary	Yes
GetOracleBlob	Yes
GetOracleBlobForUpdate	Yes
GetOracleClob	Yes
GetOracleClobForUpdate	Yes
GetOracleDate	Yes
GetOracleDecimal	Yes
GetOracleIntervalDS	No
GetOracleIntervalYM	No
GetOracleRef	No
GetOracleString	Yes
GetOracleTimeStamp	Yes
GetOracleTimeStampLTZ	No
GetOracleTimeStampTZ	No
GetOracleXmlType	No

Table 10 (Cont.) OracleDataReader class method support

Method Name	Supported?
GetOracleValue	Yes
GetOracleValues	Yes
GetOrdinal	Yes
GetProviderSpecificFieldType	Yes
GetProviderSpecificValue	Yes
GetProviderSpecificValues	Yes
GetSchemaTable	Yes
GetString	Yes
GetTimeSpan	No
GetValue	Yes
GetValues	Yes
GetXmlReader	No
IsDBNull	Yes
NextResult	No
Read	Yes

OracleTransaction class support

The following tables list support for properties and methods of the `OracleTransaction` class.

Table 11 OracleTransaction class property support

Property Name	Supported?
IsolationLevel	Yes
Connection	Yes

Table 12 OracleTransaction class method support

Class Name	Supported?	Notes
Commit	Yes	
Dispose	Yes	
Rollback	Yes	
Rollback (string)	No	TimesTen does not support transaction savepoints.
Save	No	TimesTen does not support transaction savepoints.

Oracle.DataAccess.Types namespace support

The following tables list support for structures, exceptions, classes, interfaces, and enumerations of the `Oracle.DataAccess.Types` namespace.

Table 13 Oracle.DataAccess.Types namespace structure support

Structure Name	Supported?
OracleBinary	Yes
OracleBoolean	No
OracleDate	Yes
OracleDecimal	Yes
OracleIntervalDS	No
OracleIntervalYM	No
OracleString	Yes
OracleTimeStamp	Yes
OracleTimeStampLTZ	No
OracleTimeStampTZ	No

Table 14 Oracle.DataAccess.Types namespace exception support

Class Name	Supported?
OracleTypeException	Yes
OracleNullValueException	Yes
OracleTruncateException	Yes

Table 15 Oracle.DataAccess.Types namespace class support

Class Name	Supported?
OracleArrayMappingAttribute	No
OracleBFile	No
OracleBlob	Yes
OracleClob	Yes
OracleCustomTypeMappingAttribute	No
OracleObjectMappingAttribute	No
OracleRef	No
OracleRefCursor	Yes
OracleUdt	No
OracleXmlStream	No
OracleXmlType	No

Table 16 Oracle.DataAccess.Types namespace interface support

Interface Name	Supported?
IOracleArrayTypeFactory	No
IOracleCustomType	No
IOracleCustomTypeFactory	No

Table 16 (Cont.) Oracle.DataAccess.Types namespace interface support

Interface Name	Supported?
INullable	Yes

Table 17 Oracle.DataAccess.Types namespace enumeration support

Enumeration Name	Supported?
OracleUdtFetchOption	No
OracleUdtStatus	No

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle Data Provider for .NET Oracle TimesTen In-Memory Database Support User's Guide, 12*c* Release 1 (12.1) for Windows
E38358-02