Oracle TimesTen In-Memory Database (TimesTen) is a relational database that is memory-optimized for fast response and throughput. The database resides entirely in memory at runtime and is persisted to disk storage.

- Oracle TimesTen In-Memory Database in classic mode, or TimesTen Classic, refers to single-instance and replicated databases (as in previous releases).

- Oracle TimesTen In-Memory Database in grid mode, or TimesTen Scaleout, refers to multiple-instance distributed databases. TimesTen Scaleout is a grid of interconnected hosts running instances that work together to provide fast access, fault tolerance, and high availability for in-memory data. A grid contains one or more databases and each database is distributed across all instances of the grid.

- TimesTen alone refers to both classic and grid modes (such as in references to TimesTen utilities, releases, distributions, installations, actions taken by the database, and functionality within the database).

- TimesTen Application-Tier Database Cache, or TimesTen Cache, is an Oracle Database Enterprise Edition option. TimesTen Cache is ideal for caching performance-critical subsets of an Oracle database into cache tables within TimesTen databases for improved response time in the application tier. Cache tables can be read-only or updatable. Applications read and update the cache tables using standard Structured Query Language (SQL) while data synchronization between the TimesTen database and the Oracle database is performed automatically. TimesTen Cache offers all of the functionality and performance of TimesTen Classic, plus the additional functionality for caching Oracle Database tables.

- TimesTen Replication features, available with TimesTen Classic or TimesTen Cache, enable high availability.

TimesTen supports standard application interfaces JDBC, ODBC, and ODP.NET; Oracle interfaces PL/SQL, OCI, and Pro*C/C++; and the TimesTen TTClasses library for C++.

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.

This document covers the following topics:

- Initial considerations for ODP.NET in a TimesTen environment
- Getting started with ODP.NET
Configuring TimesTen connections for an ODP.NET application
Testing your ODP.NET installation with TimesTen
Development considerations for ODP.NET with TimesTen
ODP.NET namespace and class support with TimesTen

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 Classic, covering the following topics:

- Environments and TimesTen releases supported by ODP.NET
- Support for .NET-related features
- Requirements and prerequisites for using ODP.NET with TimesTen
- 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:

- See "ODP.NET namespace and class support with TimesTen" on page 14 for details of supported namespaces and APIs. ODP.NET 12.1 is available in corresponding Oracle Database or Oracle Data Access Components (ODAC) releases.
- 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 platforms that support TimesTen. Use the 64-bit version of ODP.NET 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, which is always the case in TimesTen 18.1.
- 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 6.

Notes:

- For reference, the OCI version provided with TimesTen is under the `tt_installation_dir\ttoracle_home` directory, where `tt_installation_dir` is the TimesTen installation root directory. Do not use this version for ODP.NET applications.
- 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 TimesTen and Oracle Database documentation libraries.

TimesTen documentation is available at https://docs.oracle.com/database/timesten-18.1.

For information about ODP.NET in Oracle 12c Release 1 (12.1), see ".NET and Windows Application Development" under https://docs.oracle.com/database/121/nav/portal_5.htm

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 installations:

■ 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 “Installation and Management of TimesTen on Windows” in Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade 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

An Oracle Database installation on Windows includes ODP.NET. No special steps are necessary.

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 Select Product Languages dialog, choose your language.
2. In the Specify Oracle Home User dialog, select the appropriate user account.
3. In the Specify Installation Location dialog, specify the desired Oracle base and software location.
4. In the Available Product Components dialog, select Oracle Data Provider for .NET and any other desired components.
In the ODP.NET (Oracle Data Provider for .NET) dialog, check the box to configure ODP.NET.

In the Perform Prerequisites Checks dialog, you can optionally select desired items under "Checks" and refer to the results at the bottom of the dialog.

The Summary dialog summarizes the installation settings.

In the Install Product dialog, perform the installation.

In the Finish dialog, follow any relevant additional instructions in the dialog.

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.

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 \timesten_home\bin, where timesten_home is the TimesTen instance home directory. This should follow any other Oracle directories in the path.

   Note that on Windows, there is only one TimesTen instance per installation, and timesten_home (the instance home) refers to \tt_installation_dir\instance.

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 4). 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 5), 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
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"
```

---

**Note:** For TimesTen Classic, you can use the `ttInstanceCreate -tnsadmin` option or the `ttInstanceModify -tns_admin` option (in addition to the TNS_ADMIN environment variable) to set the tnsnames location.

---

**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_installation_dir\network\admin\samples directory, where tt_installation_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_installation_dir\ttoracle_home\instantclient_11_2 for the Oracle Database 11.2 Instant Client shipped with TimesTen 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 TimesTen Application-Tier Database Cache (TimesTen 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.)

---

**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 and access to TimesTen Quick Start sample applications to perform these tests. These instructions also assume you have Visual Studio.

---

**Note:** The TimesTen Classic Quick Start is available from the TimesTen GitHub location. There is a complete set of tutorials, how-to instructions, and sample applications.

---

1. Run the Quick Start ttquickstartenv.bat script, a superset of the ttenv.bat script typically used for TimesTen setup, to set up your environment.
2. Run the Quick Start build_sampledb.bat script. This creates a TimesTen database, sampledb, with users and objects.
3. Copy the ODP.NET sample program DemoODP.cs to your system.

4. Create a tnsnames.ora file that contains the following:
   ```
sampledb = (DESCRIPTION = (CONNECT_DATA =
   (SERVICE_NAME = sampledb) (SERVER = timesten_direct)))
   ```

5. 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
   ```

6. 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 18.1 releases is `sampledb`.

7. Execute DemoODP as follows. (The database name, user name, and password are determined automatically during execution of build_sampledb.bat.)
   ```
   DemoODP -db sampledb -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, DLVR1, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7946, DLVR2, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7947, DLVR3, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7948, DLVR4, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7949, DLVR5, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7950, DLVR6, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7951, DLVR7, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7952, DLVR8, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7953, DLVR9, DEVELOPER, 7944, 10/08/2010 12:00:00 AM, 2000, <NULL>, 50
7954, DLVR10, 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
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.

```csharp
// 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:

```csharp
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 19.

**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.

**Troubleshooting**

This section discusses solutions for various exceptions you may encounter when using ODP.NET for TimesTen.

- 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_
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. 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.

The `ttSrcScan` utility is available on the Oracle Technology Network site. See the README file there for additional information.

**ODP.NET namespace and class support with TimesTen**

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

---

**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 `timesten_home\install\lib` 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 ttcommonxxxx.dll was not found. Re-installing the application may fix the problem"**

  This indicates that the location of the TimesTen shared libraries at `timesten_home\install\lib` is not in the `PATH` environment variable setting.

---

**Note:** Instead of "xxxx", the TimesTen release number is indicated. In TimesTen 18.1 releases the file name is `ttcommon181.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. 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.

The `ttSrcScan` utility is available on the Oracle Technology Network site. See the README file there for additional 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:

```csharp
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 supported delegates, classes, and enumerations of the `Oracle.DataAccess.Client` namespace.

#### Table 1  Oracle.DataAccess.Client namespace delegate support

<table>
<thead>
<tr>
<th>Delegate Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OracleInfoMessageEventHandler</td>
<td></td>
</tr>
<tr>
<td>OracleRowUpdatedEventHandler</td>
<td></td>
</tr>
<tr>
<td>OracleRowUpdatingEventHandler</td>
<td></td>
</tr>
</tbody>
</table>

#### Table 2  Oracle.DataAccess.Client namespace class support

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OracleClientFactory</td>
<td></td>
</tr>
<tr>
<td>OracleCommand</td>
<td>See &quot;OracleCommand class support&quot; on page 16 for information about TimesTen support for properties and public methods of this class.</td>
</tr>
<tr>
<td>OracleCommandBuilder</td>
<td></td>
</tr>
</tbody>
</table>

---

15
Table 2 (Cont.) Oracle.DataAccess.Client namespace class support

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OracleConnection</td>
<td>See &quot;OracleConnection class support&quot; on page 17 for information about TimesTen support for properties and public methods of this class.</td>
</tr>
<tr>
<td>OracleConnectionStringBuilder</td>
<td></td>
</tr>
<tr>
<td>OracleDataAdapter</td>
<td>The IdentityInsert and IdentityUpdate properties are not supported.</td>
</tr>
<tr>
<td>OracleDataReader</td>
<td>See &quot;OracleDataReader class support&quot; on page 19 for information about TimesTen support for properties and public methods of this class.</td>
</tr>
<tr>
<td>OracleDataSourceEnumerator</td>
<td></td>
</tr>
<tr>
<td>OracleError</td>
<td></td>
</tr>
<tr>
<td>OracleErrorCollection</td>
<td></td>
</tr>
<tr>
<td>OracleException</td>
<td></td>
</tr>
<tr>
<td>OracleInfoMessageEventArgs</td>
<td></td>
</tr>
<tr>
<td>OracleParameter</td>
<td></td>
</tr>
<tr>
<td>OracleParameterCollection</td>
<td></td>
</tr>
<tr>
<td>OraclePermission</td>
<td></td>
</tr>
<tr>
<td>OraclePermissionAttribute</td>
<td></td>
</tr>
<tr>
<td>OracleRowUpdatedEventArgs</td>
<td></td>
</tr>
<tr>
<td>OracleRowUpdatingEventArgs</td>
<td></td>
</tr>
<tr>
<td>OracleTransaction</td>
<td>See &quot;OracleTransaction class support&quot; on page 20 for information about TimesTen support for properties and public methods of this class.</td>
</tr>
</tbody>
</table>

Table 3 Oracle.DataAccess.Client namespace enumeration support

<table>
<thead>
<tr>
<th>Enumeration Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>OracleDbType</td>
<td></td>
</tr>
<tr>
<td>OracleParameterStatus</td>
<td></td>
</tr>
</tbody>
</table>

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 supported properties and methods of the **OracleCommand** class.
Table 4  OracleCommand class property support

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddToStatementCache</td>
<td></td>
</tr>
<tr>
<td>ArrayBindCount</td>
<td></td>
</tr>
<tr>
<td>CommandText</td>
<td></td>
</tr>
<tr>
<td>CommandType</td>
<td></td>
</tr>
<tr>
<td>Connection</td>
<td></td>
</tr>
<tr>
<td>FetchSize</td>
<td></td>
</tr>
<tr>
<td>ImpliedRefCursors</td>
<td>While TimesTen supports the ImpliedRefCursors property, its use is complementary to the ADO.NET Entity Framework, which TimesTen does not support.</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>RowSize</td>
<td></td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
</tr>
<tr>
<td>UpdatedRowSource</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** ODP.NET for TimesTen does not support 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

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone</td>
<td></td>
</tr>
<tr>
<td>CreateParameter</td>
<td></td>
</tr>
<tr>
<td>ExecuteNonQuery</td>
<td></td>
</tr>
<tr>
<td>ExecuteNonQuery</td>
<td></td>
</tr>
<tr>
<td>ExecuteReader</td>
<td></td>
</tr>
<tr>
<td>ExecuteScalar</td>
<td></td>
</tr>
</tbody>
</table>

OracleConnection class support

The following tables list supported properties and methods of the OracleConnection class.

Table 6  OracleConnection class property support

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionString</td>
<td></td>
</tr>
<tr>
<td>ConnectionTimeout</td>
<td></td>
</tr>
<tr>
<td>DataSource</td>
<td></td>
</tr>
<tr>
<td>ServerVersion</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td></td>
</tr>
<tr>
<td>StatementCacheSize</td>
<td></td>
</tr>
</tbody>
</table>
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:

```csharp
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

### Table 7  OracleConnection class event support

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateChange</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8  OracleConnection class method support

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeginTransaction</td>
<td></td>
</tr>
<tr>
<td>ClearAllPools (static method)</td>
<td></td>
</tr>
<tr>
<td>ClearPool (static method)</td>
<td></td>
</tr>
<tr>
<td>Clone</td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td></td>
</tr>
<tr>
<td>CreateCommand</td>
<td></td>
</tr>
<tr>
<td>GetSchema</td>
<td>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.</td>
</tr>
<tr>
<td>GetSessionInfo</td>
<td></td>
</tr>
<tr>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>PurgeStatementCache</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** TimesTen does not support distributed transactions through OCI. An ODP.NET application cannot use `EnlistDistributedTransaction` or `EnlistTransaction` in a TimesTen connection.
OracleDataReader class support
The following tables list supported properties and methods of the OracleDataReader class.

### Table 9  OracleDataReader class property support

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td></td>
</tr>
<tr>
<td>FetchSize</td>
<td></td>
</tr>
<tr>
<td>FieldCount</td>
<td></td>
</tr>
<tr>
<td>HasRows</td>
<td></td>
</tr>
<tr>
<td>HiddenFieldCount</td>
<td></td>
</tr>
<tr>
<td>IsClosed</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td></td>
</tr>
<tr>
<td>RowSize</td>
<td></td>
</tr>
<tr>
<td>VisibleFieldCount</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td></td>
</tr>
<tr>
<td>Dispose</td>
<td></td>
</tr>
<tr>
<td>GetByte</td>
<td></td>
</tr>
<tr>
<td>GetBytes</td>
<td></td>
</tr>
<tr>
<td>GetChar</td>
<td></td>
</tr>
<tr>
<td>GetChars</td>
<td></td>
</tr>
<tr>
<td>GetDataTypeName</td>
<td></td>
</tr>
<tr>
<td>GetDateTime</td>
<td></td>
</tr>
<tr>
<td>GetDecimal</td>
<td></td>
</tr>
<tr>
<td>GetDouble</td>
<td></td>
</tr>
<tr>
<td>GetFieldTyp</td>
<td></td>
</tr>
<tr>
<td>GetFloat</td>
<td></td>
</tr>
<tr>
<td>GetInt16</td>
<td></td>
</tr>
<tr>
<td>GetInt32</td>
<td></td>
</tr>
<tr>
<td>GetInt64</td>
<td></td>
</tr>
<tr>
<td>GetName</td>
<td></td>
</tr>
<tr>
<td>GetOracleBinary</td>
<td></td>
</tr>
</tbody>
</table>
OracleTransaction class support
The following tables list supported properties and methods of the OracleTransaction class.

**Table 11  OracleTransaction class property support**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsolationLevel</td>
<td></td>
</tr>
<tr>
<td>Connection</td>
<td></td>
</tr>
</tbody>
</table>

**Table 12  OracleTransaction class method support**

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit</td>
<td></td>
</tr>
<tr>
<td>Dispose</td>
<td></td>
</tr>
<tr>
<td>Rollback</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** TimesTen does not support transaction savepoints.
Oracle.DataAccess.Types namespace support

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

<table>
<thead>
<tr>
<th>Table 13</th>
<th>Oracle.DataAccess.Types namespace structure support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure Name</td>
<td>Notes</td>
</tr>
<tr>
<td>OracleBinary</td>
<td></td>
</tr>
<tr>
<td>OracleDate</td>
<td></td>
</tr>
<tr>
<td>OracleDecimal</td>
<td></td>
</tr>
<tr>
<td>OracleString</td>
<td></td>
</tr>
<tr>
<td>OracleTimeStamp</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 14</th>
<th>Oracle.DataAccess.Types namespace exception support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name</td>
<td>Notes</td>
</tr>
<tr>
<td>OracleTypeException</td>
<td></td>
</tr>
<tr>
<td>OracleNullValueException</td>
<td></td>
</tr>
<tr>
<td>OracleTruncateException</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 15</th>
<th>Oracle.DataAccess.Types namespace class support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name</td>
<td>Notes</td>
</tr>
<tr>
<td>OracleBlob</td>
<td></td>
</tr>
<tr>
<td>OracleClob</td>
<td></td>
</tr>
<tr>
<td>OracleRefCursor</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 16</th>
<th>Oracle.DataAccess.Types namespace interface support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Name</td>
<td>Notes</td>
</tr>
<tr>
<td>INullable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 17</th>
<th>Oracle.DataAccess.Types namespace enumeration support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumeration Name</td>
<td>Notes</td>
</tr>
<tr>
<td>(No enumerations are supported.)</td>
<td></td>
</tr>
</tbody>
</table>

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 that have purchased support 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.