Oracle Data Provider .NET

Developers Guide 12c Release 2 (12.2) Beta 3 for Microsoft .NET Core

July 2018

Introduction

Oracle Data Provider for .NET (ODP.NET) Core is an ADO.NET driver that provides fast data access from Microsoft .NET Core clients to Oracle databases. ODP.NET Core enables .NET applications to take advantage of Oracle advanced features, such as Oracle Real Application Clusters (Oracle RAC) and Oracle Data Guard. It is accessible through any .NET language, including C#, Visual Basic .NET, and C++ .NET.

ODP.NET Core consists of a single 100% managed code dynamic-link library, Oracle.ManagedDataAccess.dll, available via xcopy deployment. It uses the following namespaces:

- Oracle.ManagedDataAccess.Client
    - Namespace for ODP.NET classes and enumerations
- Oracle.ManagedDataAccess.Types
    - namespace for ODP.NET-specific data types

ODP.NET Core employs the same namespaces and application programming interfaces (APIs) as ODP.NET, Managed Driver. This parallel eases migration and developer learning curve from managed ODP.NET to ODP.NET Core. It does not support all managed ODP.NET functionality. ODP.NET Core supports a subset of managed ODP.NET APIs. These differences are listed later on in this documentation.

Changes since Last Release

This release introduces support for Configuration API.

Transport Layer Security/Secure Sockets Layer (TLS/SSL) is now supported on Linux in addition to Windows.

This release removes support for environment variables. Environment variables will not be supported in the ODP.NET Core production.

Some sqlnet.ora and tnsnames.ora parameters have been removed that have no or limited usage capabilities in .NET Core or with ODP.NET Core.

Kerberos is not supported in this beta.

System Requirements

ODP.NET Core 12.2 requires the following:

- Operating System
    - Windows x64

        - Windows 7 SP1+ (Professional, Enterprise, and Ultimate Editions)
        - Windows 8.1 (Pro and Enterprise Editions)
        - Windows 10 x64 (Pro, Enterprise, and Education Editions)
        - Windows Server 2012 x64 (Standard, Datacenter, Essentials, and Foundation Editions)
        - Windows Server 2012 R2 x64 (Standard, Datacenter, Essentials, and Foundation Editions)
        - Windows Server 2016 x64 (Standard and Datacenter Editions)
    - Linux x64
        - Oracle Linux 7
- .NET
    - .NET Core 2.1 or higher
- Access to Oracle Database 11g Release 2 (11.2) or higher

ODP.NET Core is compatible with ASP.NET Core 2.0 and ASP.NET.

ODP.NET Core is built with AnyCPU. It supports 64-bit .NET and will support 32-bit .NET in a subsequent release.

Installation

Follow these steps to install ODP.NET Core:

1. Download ODP.NET Core from Oracle Technology Network.
2. Unzip the contents of the downloaded package and the DLL to your application directory.

You can also install ODP.NET Core from nuget.org.

ODP.NET Core will be available via Oracle Universal Installer in a future release.

Configuration

ODP.NET Core developers can assign application settings in .NET Configuration API, sqlnet.ora file, and tnsnames.ora file.

.NET Configuration API

.NET Core does not support application configuration via .NET configuration files (i.e. web.config). Instead, it uses .NET Configuration API in lieu of a configuration file. ODP.NET Core supports Configuration API via the static class, OracleConfiguration, for application level provider settings. The OracleDataSourceCollection class supports adding and deleting net services names (i.e. TNS entries).

Please note that all configurations settings through OracleConfiguration should be done before opening any connection in the application. Once a connection is opened, any updates to configuration properties will result in InvalidOperationException; with only exception of trace settings that are still allowed to change during application runtime.

Oracle.ManagedDataAccess.Client.OracleConfiguration Class

OracleConfiguration is a static class for setting ODP.NET Core configuration data using a single programming interface.

```
public static class OracleConfiguration
```

OracleConfiguration Connection Properties

OracleConfiguration.DatabaseEditionName

Specifies the Oracle edition name for the connection object. This property is used with the Oracle Edition-Based Redefinition feature.

// C#

```
public static string DatabaseEditionName { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.DisableOOB

Specifies whether to enable Oracle Net Services to send or receive out-of-band break messages using urgent data provided by the underlying protocol. Default is false.

// C#

```
public static bool DisableOOB { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.DrcpConnectionClass

Specifies a logical name that identifies the DRCP connection pool that the ODP.NET connection will use. It will be used as a default if the DRCPConnectionClass property on the OracleConnection object is not set. It will be ignored for non-DRCP connections. Default value is null.

// C#

```
public static string DrcpConnectionClass { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.HostnameDefaultServiceIsHost

Specifies whether to default the service name to the hostname in Oracle Easy Connect Naming so that the service name does not need to be specified.

// C#

```
public static bool HostnameDefaultServiceIsHost { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.NamesDirectoryPath

Specifies the naming method order used for client name resolution lookups.

// C#

```
public static string[] NamesDirectoryPath { get; set; }
```

Property Value:

Type: System.String[]

OracleConfiguration.OracleDataSources

Returns a collection that can be used to add and delete net service name entries (i.e. TNS names).

// C#

```
public static OracleDataSourceCollection OracleDataSources { get; }
```

Property Value:

Type: Oracle.ManagedDataAccess.Client.OracleDataSourceCollection

Returns a static OracleDataSourceCollection object.

OracleConfiguration.TcpConnectTimeout

Specifies the time, in seconds, for a client to establish a TCP connection (PROTOCOL=tcp in the TNS connect address) to the database server before it can time out. Default value is 60 seconds.

// C#

```
public static string TcpConnectTimeout { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.TnsAdmin

Specifies the tnsnames.ora and/or sqlnet.ora directory location.

// C#

```
public static string TnsAdmin { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration Secure Connection Properties

OracleConfiguration.SqlNetAuthenticationServices

Enables one or more authentication services, such as Kerberos or TCP/IP with SSL. If authentication has been installed, then this parameter should be set to either none or to one of the listed authentication methods.

// C#

```
public static string[] SqlNetAuthenticationServices { get; set; }
```

Property Value:

Type: System.String[]

Valid Values:

Kerberos5, NTS, TCPS, or NONE.

OracleConfiguration.SqlNetEncryptionClient

Specifies the encryption client behavior. Default value is "accepted."

// C#

```
public static string SqlNetEncryptionClient { get; set; }
```

Property Value:

Type: System.String

Valid Values:

- accepted - to enable the security service if required or requested by the database.
- rejected - to disable the security service, even if required by the database.
- requested - to enable the security service if the database allows it.
- required - to enable the security service and disallow the connection if the database is not enabled for the security service.

OracleConfiguration.SqlNetEncryptionTypesClient

Specifies encryption algorithms that the client can use.

// C#

```
public static string[] SqlNetEncryptionTypesClient { get; set; }
```

Property Value:

Type: System.String[]

Valid Values:

- AES128
- AES192
- AES256
- RC4_128
- RC4_256
- 3DES112
- 3DES168

OracleConfiguration.SqlNetCryptoChecksumClient

Specifies the checksum client behavior. Default value is “accepted.”

// C#

```
public static string SqlNetCryptoChecksumClient { get; set; }
```

Property Value:

Type: System.String

Valid Values:

- accepted - to enable the security service if required or requested by the database.
- rejected - to disable the security service, even if required by the database.
- requested - to enable the security service if the database allows it.
- required - to enable the security service and disallow the connection if the database is not enabled for the security

OracleConfiguration.SqlNetCryptoChecksumTypesClient

Specifies the crypto-checksum algorithms the client can use.

// C#

```
public static string[] SqlNetCryptoChecksumTypesClient { get; set; }
```

Property Value:

Type: System.String[]

Valid Values:

- SHA1
- SHA256
- SHA384
- SHA512

OracleConfiguration.SqlNetKerberos5Conf

Specifies the Kerberos configuration file complete path name, which contains the realm for the default Key Distribution Center (KDC) and maps realms to KDC hosts.

// C#

```
public static string SqlNetKerberos5Conf { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.SqlNetKerberos5CCName

Specifies the Kerberos credentials cache file complete path name.

// C#

```
public static string SqlNetKerberos5CCName { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.SqlNetWalletOverride

Specifies whether the client overrides the strong authentication credential with the stored wallet password credential for the database connection.

// C#

```
public static bool SqlNetWalletOverride { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.SSLServerDNMatch

Enforces the database server distinguished name (DN) matches its service name. Default value is false.

// C#

```
public static bool SSLServerDNMatch { get; set; }
```

Property Value:

Type: System.Boolean

Valid Values:

- yes | on | true - to enforce a match. If the DN matches the service name, then the connection succeeds. If the DN does not match the service name, then the connection fails.
- no | off | false - to not enforce a match. If the DN does not match the service name, then the connection is successful, but an error is logged to the sqlnet.log file.

OracleConfiguration.SSLVersion

Forces a version of the SSL/TLS connection. Default value is undetermined.

// C#

```
public static string SSLVersion { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.WalletLocation

Specifies the location of wallets. Wallets are certificates, keys, and trustpoints processed by SSL/TLS.

// C#

```
public static string WalletLocation { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration Command Properties

OracleConfiguration.BindByName

Specifies whether the binding method used for the parameter collection is by name or by position. Default value (false) is bind by position.

// C#

```
public static bool BindByName { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.CommandTimeout

Specifies the number of seconds the command is allowed to execute before terminating the execution with an exception. Default value is 0 seconds, which results in no time limit.

// C#

```
public static int CommandTimeout { get; set; }
```

Property Value:

Type: System.Int32.

OracleConfiguration High Availability Properties

OracleConfiguration.DbNotificationPort

Specifies the port number that the provider listens to for all notifications sent by the database for continuous query notification, Fast Connection Failover, and/or Runtime Connection Load Balancing features. A value of -1 allows directs the provider to use a random port.

// C#

```
public static int DbNotificationPort { get; set; }
```

Property Value:

Type: System.Int32.

OracleConfiguration.HAEvents

Enables the application to receive Fast Connection Failover events for maintaining application high availability. Default is true.

// C#

```
public static bool HAEvents { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.OnsConfigFile

Specifies the configuration file to define Oracle Notification Service (ONS) behavior. The file specified should contain the same local port and remote port values as specified in the ons.config file used by the local ONS daemon. This will enable the application to receive events from the local ONS daemon.

// C#

```
public static string OnsConfigFile { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.OnsMode

Specifies the ONS daemon mode as either "local" or "remote".

For local configuration, configure and make available ONS on the node where ODP.NET is running so that ODP.NET can receive events directly from the local ONS daemon.

Remote configuration is used when the application directly receives ONS events from the ONS daemons running on remote machines. One of the advantages of this configuration is that no ONS daemon is needed on the client end and; therefore, there is no need to manage this process.

// C#

```
public static OnsConfigMode OnsMode { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.OnsDatabase

Specifies a dictionary of all potential databases with their corresponding list of nodes where the ONS daemons are talking to their remote clients.

In case of remote configuration, the application has to specify the <host>:<port> values for every potential database that it can connect to. The <host>:<port> value pairs represent the ports on the different Oracle RAC nodes.

// C#

```
public static ConcurrentDictionary<string, ConcurrentDictionary<string, string>>
OnsDatabase { get; set; }
```

Property Value:

Type: `ConcurrentDictionary<string, ConcurrentDictionary<string, string>>`

OracleConfiguration.ServiceRelocationConnectionTimeout

Specifies the time to wait before retrying connecting to a service that becomes unavailable. Default value is 90 seconds. Whenever a database service becomes unavailable, such as due to a service being relocated, an application can encounter numerous connectivity errors during this time. To avoid unnecessary connection attempts to an unavailable service which will result in an error, the driver will block any connection attempts until the service is up or until this property's specified time limit expires from the time when the service DOWN event was received, whichever comes first.

// C#

```
public static string ServiceRelocationConnectionTimeout { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration Performance Properties

OracleConfiguration.FetchSize

Specifies the total memory size, in bytes, that the provider allocates to cache data fetched in one database round-trip. Default value is 131072.

// C#

```
public static int FetchSize { get; set; }
```

Property Value:

Type: System.Int32

OracleConfiguration.LoadBalancing

Enables the application to receive runtime connection load balancing information. Default is true.

// C#

```
public static bool LoadBalancing { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.MaxStatementCacheSize

Specifies the maximum number of statements that can be cached when self-tuning is enabled. Default value is 100.

// C#

```
public static int MaxStatementCacheSize { get; set; }
```

Property Value:

Type: System.Int32

OracleConfiguration.ReceiveBufferSize

Specifies the buffer space limit for receive operations of sessions.

```
// C#

public static int ReceiveBufferSize{ get; set; }
```

Property Value:

Type: System.Int32

OracleConfiguration.SelfTuning

Specifies whether self-tuning is enabled for an ODP.NET application. By default, self-tuning (true) is enabled.

```
// C#

public static bool SelfTuning { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration.SendBufferSize

Specifies the buffer space limit for send operations of sessions.

```
// C#

public static int SendBufferSize { get; set; }
```

Property Value:

Type: System.Int32

OracleConfiguration.StatementCacheSize

Specifies the number of cursors or statements to be cached for each database connection. This setting corresponds to the connection string Statement Cache Size attribute. A value greater than zero enables statement caching. Default value is zero.

// C#

```
public static int StatementCacheSize { get; set; }
```

Property Value:

Type: System.Int32

OracleConfiguration.TcpNoDelay

Preempts delays in buffer flushing within the TCP/IP protocol stack. Default value is true.

// C#

```
public static bool TcpNoDelay { get; set; }
```

Property Value:

Type: System.Boolean

OracleConfiguration Debug Tracing Properties

OracleConfiguration.TraceFileLocation

Specifies the destination directory to output provider traces.

// C#

```
public static string TraceFileLocation { get; set; }
```

Property Value:

Type: System.String

OracleConfiguration.TraceLevel

Specifies the generated trace level to trace ODP.NET calls and diagnose provider issues. Errors will always be traced. Default value is 0 indicating tracing is disabled.

// C#

```
public static int TraceLevel { get; set; }
```

Property Value:

Type: System.Int32

Valid Values:

- 1 = public APIs
- 2 = private APIs
- 4 = network APIs/data

These values can be ORed. To enable all traces, set TraceLevel to 7.

OracleConfiguration.TraceOption

Specifies whether to generate a single trace file or multiple trace files for multithreaded applications. Default value is 0 indicating single trace file for all application threads.

// C#

```
public static int TraceOption { get; set; }
```

Property Value:

Type: System.Int32

Oracle.ManagedDataAccess.Client.OracleDataSourceCollection Class

The OracleDataSourceCollection class supports adding and deleting network service name (i.e. TNS) entries in the OracleDataSourceCollection.

OracleDataSourceCollection.Add

Add unique network service name (TNS) entries in the OracleDataSourceCollection using net service names and complete connect descriptors.

// C#

```
public void Add(string tnsName, string tnsDescriptor)
```

Parameters:

- tnsName
  - Type: System.String
  - Network service name used as the ODP.NET connection string Data Source name.
- tnsDescriptor
  - Type: System.String
  - The complete connect descriptor information for this specific net service name.

OracleDataSourceCollection.Count

Read-only property that returns the number of OracleDataSourceCollection elements

// C#

```
public int Count { get; }
```

Property Value:

Type: System.Int32

OracleDataSourceCollection.Remove

Remove network service name entries from an OracleDataSourceCollection.

// C#

```
public void Remove(string tnsName)
```

Parameters:

- tnsName
    - Type: System.String
    - Unique network service name to be removed from the collection.

OracleDataSourceCollection.this

Returns or sets the network service name entry with the specified network service name.

// C#

```
public string this[string tnsName] { get; set; }
```

Parameters:

- tnsName
    - Type: System.String
    - Unique network service name to be added or updated.

Oracle Configuration Files

ODP.NET Core supports the sqlnet.ora and tnsnames.ora parameters below, which is a subset of managed ODP.NET `<settings>` section settings. These settings can be used in conjunction with .NET Configuration API.

- `BindByName`
- `DbNotificationPort`
- `Disable_Oob - sqlnet.ora`
- `DRCPConnectionClass`
- `FetchSize`
- `MaxStatementCacheSize`
- `NAMES.DIRECTORY_PATH - sqlnet.ora`
- `NODELAY - sqlnet.ora`
- `RETRY_COUNT`
- `RETRY_DELAY`
- `RECEIVE_BUF_SIZE - sqlnet.ora or tnsnames.ora`
- `SelfTuning`
- `SEND_BUF_SIZE - sqlnet.ora or tnsnames.ora`
- `ServiceRelocationConnectionTimeout`
- `SQLNET.AUTHENTICATION_SERVICES - sqlnet.ora`
- `SQLNET.CRYPTO_CHECKSUM_CLIENT - sqlnet.ora`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT - sqlnet.ora`
- `StatementCacheSize`

- `SSL_SERVER_DN_MATCH - sqlnet.ora`
- `SSL_VERSION - sqlnet.ora`
- `TNS_ADMIN`
- `TraceFileLocation`
- `TraceLevel`
- `TraceOption`
- `TCP.CONNECT_TIMEOUT - sqlnet.ora`
- `SQLNET.ENCRYPTION_CLIENT - sqlnet.ora`
- `SQLNET.ENCRYPTION_TYPES_CLIENT - sqlnet.ora`

ODP.NET Core will look for sqlnet.ora and tnsnames.ora files in the following precedence order:

1. Directory set in OracleConfiguration.TnsAdmin property
2. Directory of the running ODP.NET Core assembly
3. Current working directory

Application Programming Interfaces and Configuration Setting Differences with Managed ODP.NET

ODP.NET Core supports all the same APIs as ODP.NET, Managed Driver with some exceptions. The most common reason for non-support is due to .NET Standard non- support the underlying APIs and may not have a .NET Core-specific implementation.

ODP.NET Core includes OracleConfiguration class, which is not currently available in managed ODP.NET.

Table: Features Not Supported by ODP.NET Core

Feature	Class or APIs	.NET Standard Support	Notes
Entity Framework	System.Data.Metadata.Edm	N	Planned for future release
ADO.NET Provider Configuration File	System.Configuration.ConfigurationManager	N	
Registry	Microsoft.Win32.RegistryKey	N	
.NET Configuration File	N/A	N/A	
Factory Classes	DbProviderFactories	N	
Event Log	System.Diagnostics.EventLog is not supported by .NET Standard	N	
Performance Counters	System.Diagnostics.PerformanceCounter	N	

Distributed Transactions	System.EnterpriseServices and OracleConnection.EnlistDistributedTransaction(ITransction)	N	Only local transactions are supported with TransactionScope class and EnlistTransaction method. See Note below.
Code Access Security	Examples: OraclePermission, OraclePermissionAttribute, ConfigurationPermissionAttribute, FileIOPermissionAttribute, DnsPermissionAttribute, SocketPermissionAttribute, and OracleClientFactory.CreatePermission	N	
Lightweight Directory Access Protocol	System.DirectoryServices	N	

Note: ODP.NET Core does not support distributed transactions. If you attempt to enlist ODP.NET Core in a distributed transaction, such as promoting a local transaction to distributed, you will receive an error, such as a platform not supported exception. This error also occurs if a connection attempt is made within a TransactionScope after the initial connection that was opened in the same TransactionScope is severed due to an unplanned outage.

ODP.NET Core Windows Only Features

The following features are available on Windows platforms only:

- Secure External Password Store (SEPS)
- Windows Native Authentication