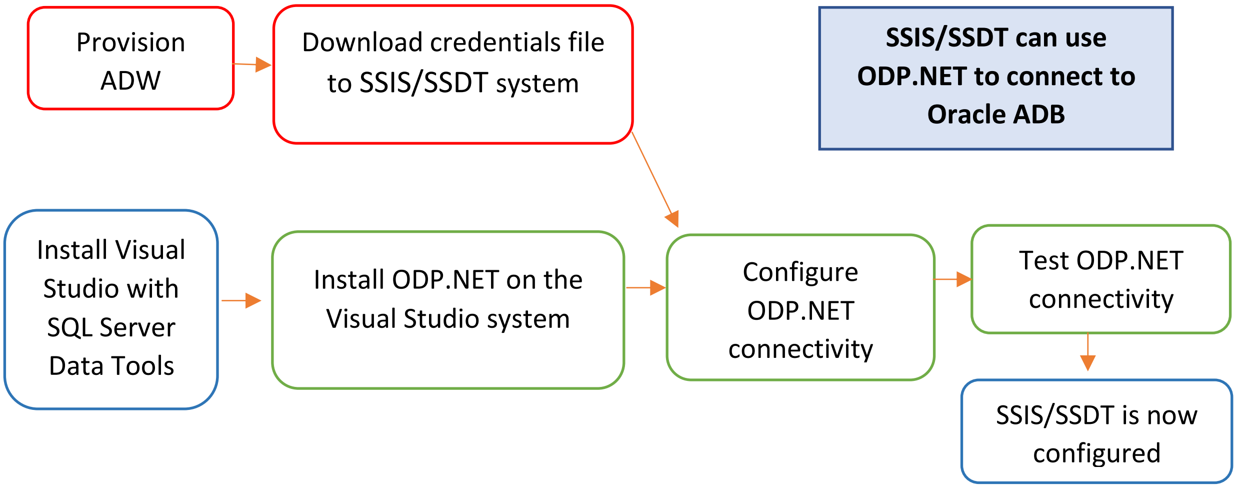
Connecting Microsoft SQL Server Integration Services and Microsoft SQL Server Data Tools to Oracle Autonomous Database

Vijay Balebail, Alex Keh

This step by step tutorial guides how to configure Microsoft SQL Server Integration Services (SSIS) and SQL Server Data Tools (SSDT) connectivity to Oracle Autonomous Database (ADB). SSDT is a tool that designs and deploys Integration Services and other BI packages generally from within Microsoft Visual Studio. These instructions use Oracle Data Provider for .NET (ODP.NET) for data access and work for both dedicated and shared infrastructure ADB.

You can use either managed or unmanaged ODP.NET. Managed ODP.NET is recommended for easier setup and deployment.

Certification Matrix	Version
Microsoft SSIS and SSDT	Independent of Version
ODP.NET, Managed Driver or Unmanaged Driver	12.2 or higher



Provision ADW

Download credentials file to SSIS/SSDT system

SSIS/SSDT can use ODP.NET to connect to Oracle ADB

Install Visual Studio with SQL Server Data Tools

Install ODP.NET on the Visual Studio system

Configure ODP.NET connectivity

Test ODP.NET connectivity

SSIS/SSDT is now configured

1. This document assumes that the Autonomous Data Warehouse (ADW) or Autonomous Transaction Processing (ATP) has been provisioned and the corresponding credentials.zip file has been downloaded to the system that has SSDT installed. For the Oracle documentation to provision ADW, please check here. Also check the Downloading Client Credentials (Wallets) documentation section.

2. Follow the instructions from the SSIS documentation to install SSDT software on your machine. As a prerequisite, make sure that you have installed the ODP.NET 12.2 or higher on the SSDT environment. You can download ODP.NET separately or as part of the Oracle

Database Client for Windows. Note: If you are using unmanaged ODP.NET, you must use 32-bit Oracle Client installation to match the fact that SSDT is a 32-bit app.

3. All connections to Autonomous Database use certificate-based authentication and Transport Layer Security/Secure Sockets Layer (TLS/SSL). Copy the client credentials file that you downloaded in step 1 to the system running SSDT and uncompress it into a secure folder.

4. We first validate that the Oracle Database Client can communicate with ADB, and since it is installed on the same system as the SSDT and SSIS, it ensures that SSIS connectivity to Oracle is also configured correctly.

5. Steps 5, 6 and 7 configure and test the Oracle Database Client. Edit the `sqlnet.ora` file, replacing "?/network/admin" with the name of the folder containing the client credentials.

   For example:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file) (METHOD_DATA =
   (DIRECTORY=C:\extract\Oracle_wallet)))

   SSL_SERVER_DN_MATCH=yes
   ```

   Be sure to remove the quotes around the directory name.

6. Create the `TNS_ADMIN` system environment variable and set it to the location of the secure folder containing the credentials file you saved in Step 3. The `tnsnames.ora` file provided with the credentials zip file contains three database service names identifiable as `high`, `medium` and `low`. The predefined service names provide different levels of performance and concurrency for ADB.

7. Test the Oracle Client with Oracle SQL*Plus

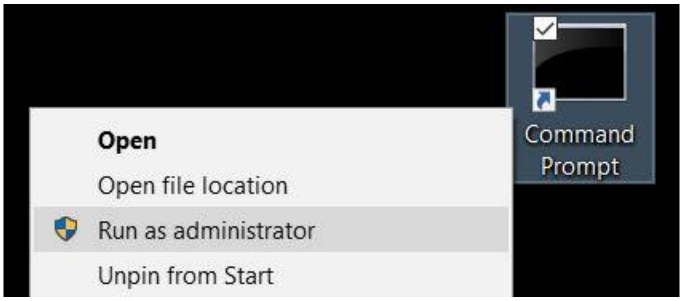
   ```
   sqlplus password/\"Password\"@ConnectString
   ```
   or
   ```
   sqlplus /nolog
   sql> set define off
   sql> connect username/password@connectString
   ```

8. SSDT and SSIS can use managed or unmanaged ODP.NET to connect to ADB. Transport Layer Security (TLS) 1.2 support has been included in version 12.2 and higher.

9. The next step is to configure managed or unmanaged ODP.NET in the .NET machine.config file so that it can be used by SSDT or SSIS. Note that SSDT and SSIS tools in Visual Studio are 32-bit applications and require a 32-bit-compatible data provider, such as managed ODP.NET or 32-bit unmanaged ODP.NET.

10. If you don’t have ODP.NET installed, download 32-bit ODAC 12.2 xcopy or higher. The xcopy downloads are located in the middle of this download page. The ODAC xcopy download includes both managed ODP.NET and 32-bit unmanaged ODP.NET. Unzip the contents to a staging directory (e.g. c:\xcopy32). This tutorial assumes you have downloaded ODAC 19c and the instructions will reflect the download’s specifics.

11. Open a command prompt in **Administrator** mode. You can do this by right-clicking on the command prompt, then choosing “Run as administrator”.



12. Navigate to the 32-bit staging directory and run:

- **install.bat odp.net4 <installation directory> odp**

Enter an install location (e.g. c:\odp32) for the directory parameter. This command installs 32-bit unmanaged ODP.NET and managed ODP.NET into that directory.

13. Navigate to the following subdirectory in that installation location: **<installation location>\odp.net\bin\4**. Execute one of the following set of commands based on the provider you plan to use.

For managed ODP.NET:

- **OraProvCfg /action:config /product:odpm /frameworkversion:v4.0.30319 /providerpath:"..\..\managed\common\Oracle.ManagedDataAccess.dll"**
- **OraProvCfg /action:gac /providerpath:" ..\..\managed\common\Oracle.ManagedDataAccess.dll"**

For unamanged ODP.NET:

- **OraProvCfg /action:config /product:odp /frameworkversion:v4.0.30319 /providerpath:"Oracle.DataAccess.dll"**
- **OraProvCfg /action:gac /providerpath:"Oracle.DataAccess.dll"**

Below is the output from running the above command with managed ODP.NET 19c. You should see the following output or similar depending on the ODP.NET version and type you use:

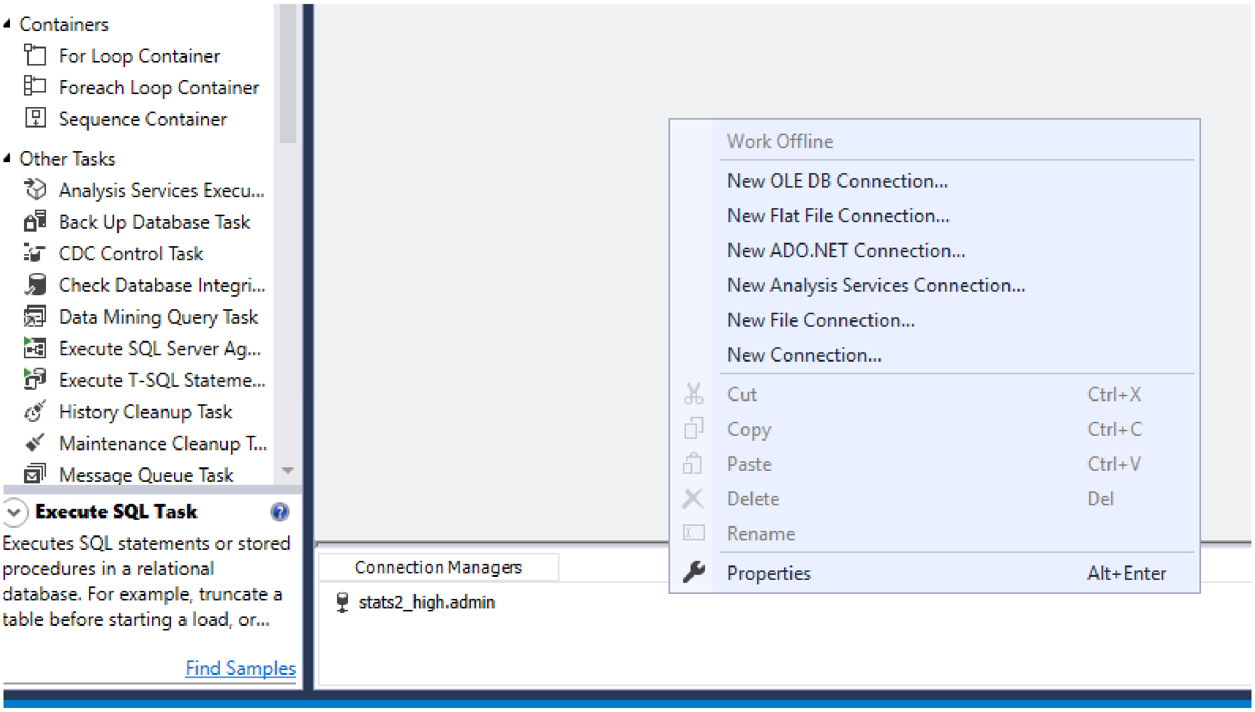
```
INFO: The following section has been added.
<section name="oracle.manageddataaccess.client"
type="OracleInternal.Common.ODPMSectionHandler,
Oracle.ManagedDataAccess, Version=4.122.19.1, Culture=neutral,
PublicKeyToken=89b483f429c47342" />

INFO: The following element added under DbProviderFactories.
<add name="ODP.NET, Managed Driver"
invariant="Oracle.ManagedDataAccess.Client" description="Oracle Data
Provider for .NET, Managed Driver"
type="Oracle.ManagedDataAccess.Client.OracleClientFactory,
Oracle.ManagedDataAccess, Version=4.122.19.1, Culture=neutral,
PublicKeyToken=89b483f429c47342" />
```

The following output indicates successfully GACing managed or unmanaged ODP.NET:

```
INFO: ..\..\managed\common\oracle.manageddataaccess.dll is registered
successfully in GAC.
INFO: oracle.dataaccess.dll is registered successfully in GAC.
```

14. Newer Visual Studio versions include SSDT as an optional install. Once installed, SSDT is integrated into Visual Studio. Open Visual Studio and create a new SSIS project.

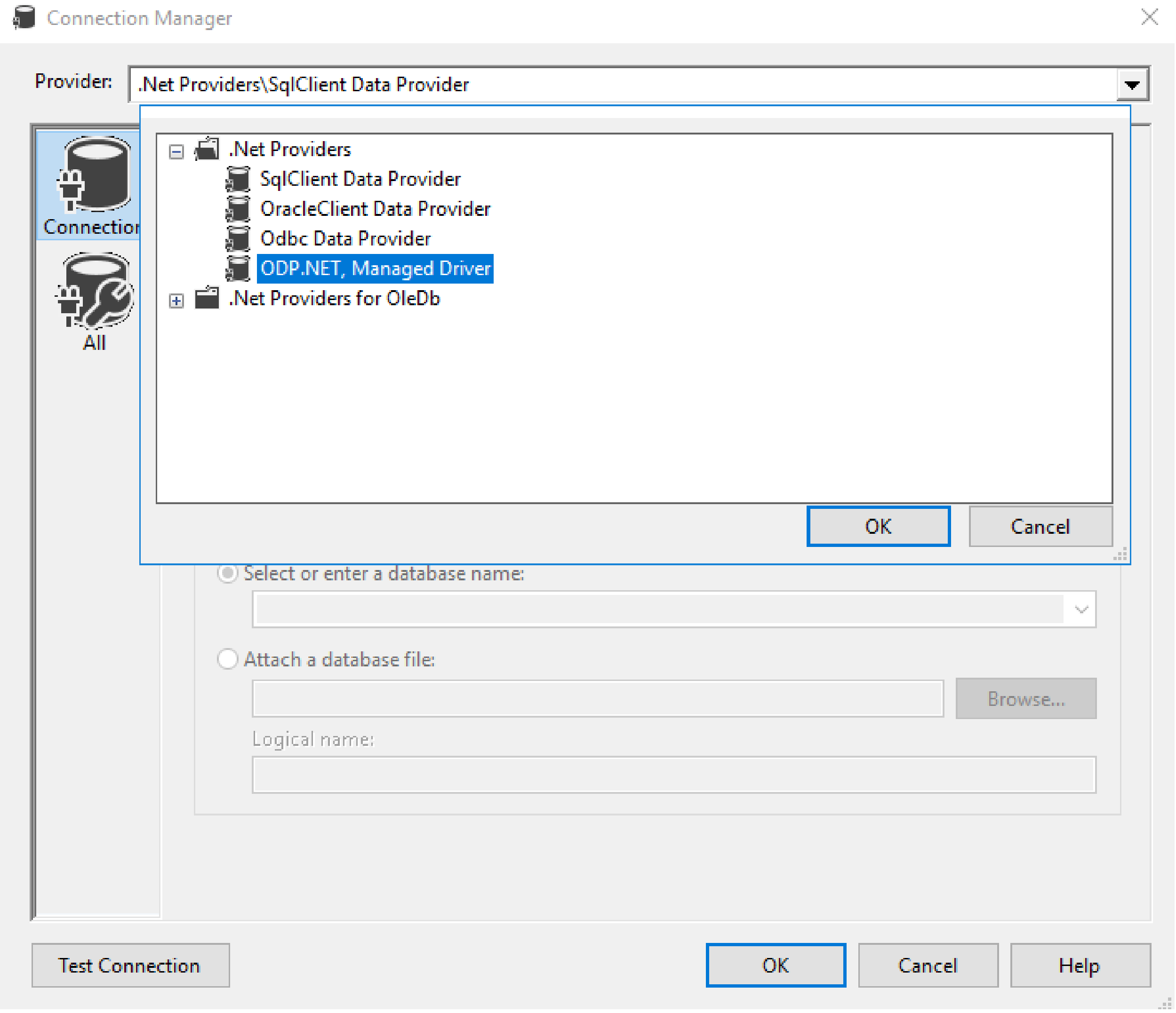
15. Right click on Connection Managers area at the bottom of the project and click on New ADO.NET Connection…



Executes SQL statements or stored procedures in a relational database. For example, truncate a table before starting a load, or...

16. In the Configure ADO.NET Connection Manager pop-up, click "New…" to create a new connection.

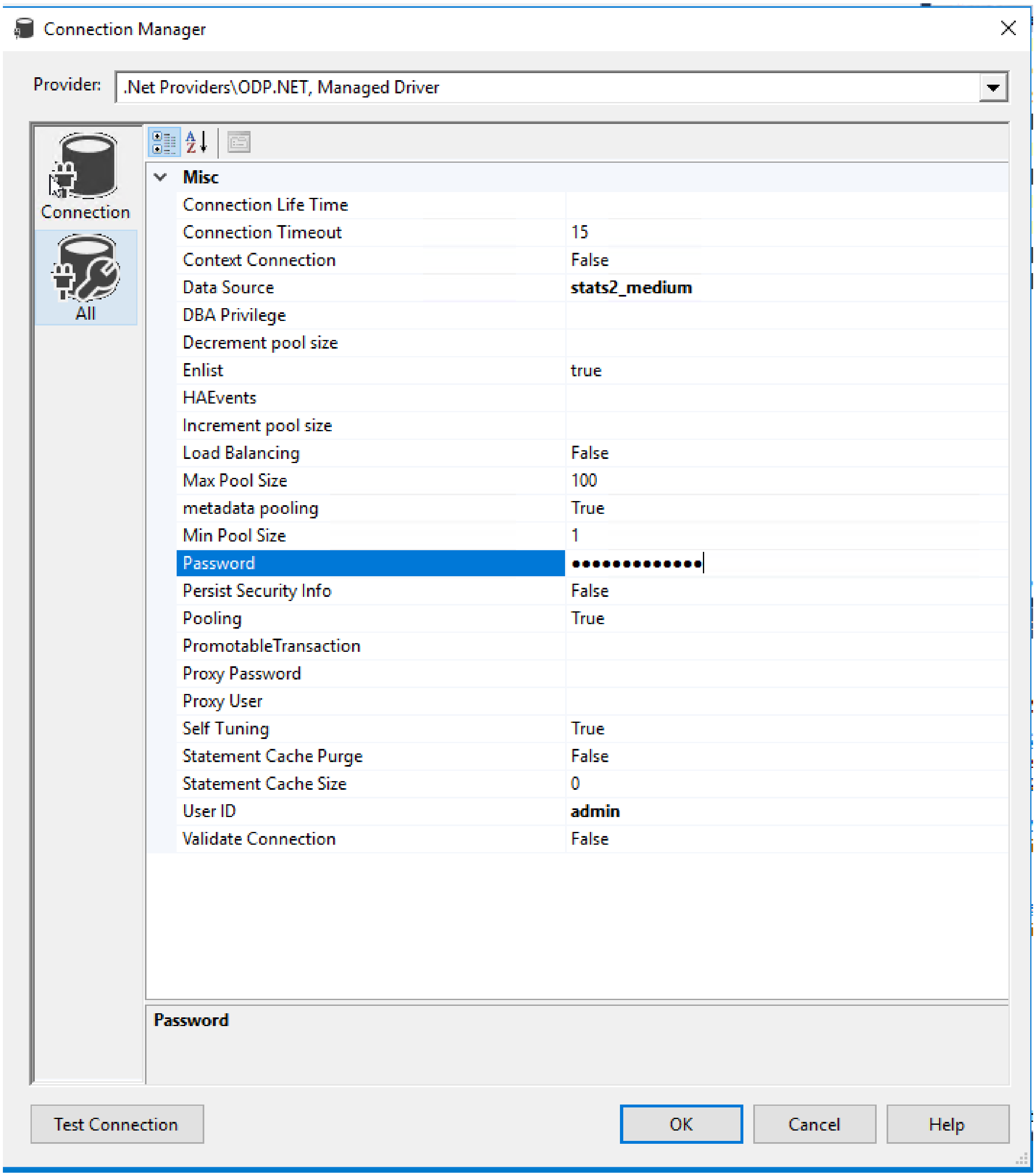
17. In the .NET Provider tree control, pick "ODP.NET, Managed Driver" or "ODP.NET, Unmanaged Driver". Only the ODP.NET providers you have specifically configured with the OraProvCfg utility will appear. Press OK.



Provider: .Net Providers\SqlClient Data Provider

18. Enter in your ADB net service name in Data Source (e.g. stats2_medium), user id (e.g. admin), and password information into the respective fields.

    In some cases, managed ODP.NET user id is case-sensitive. If so, capitalize the user id characters (e.g. ADMIN) when entering in the credentials.



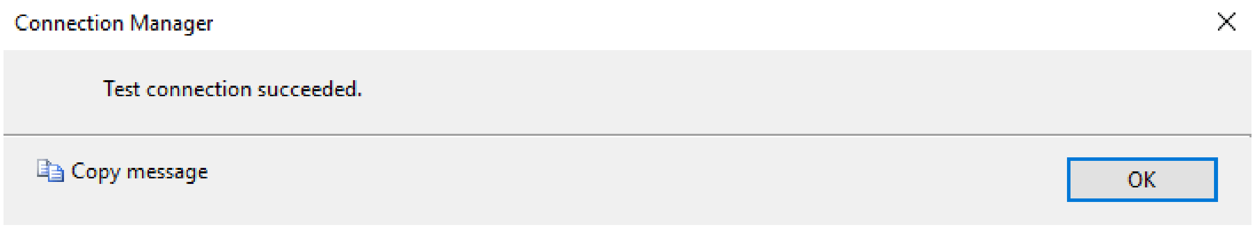
Provider: .Net Providers\ODP.NET, Managed Driver

Misc

Connection Life Time	
Connection Timeout	15
Context Connection	False
Data Source	**stats2_medium**
DBA Privilege	
Decrement pool size	
Enlist	true
HAEvents	
Increment pool size	
Load Balancing	False
Max Pool Size	100
metadata pooling	True
Min Pool Size	1
Password	●●●●●●●●●●●●●
Persist Security Info	False
Pooling	True
PromotableTransaction	
Proxy Password	
Proxy User	
Self Tuning	True
Statement Cache Purge	False
Statement Cache Size	0
User ID	**admin**
Validate Connection	False

Password

19. Click Test Connection


Connection Manager

Test connection succeeded.

If you have successfully tested the connection, your SSIS project and SSDT can connect to ADB.

ODP.NET Performance Tuning for Large Data Retrievals

Typically, BI and ETL applications retrieve large amounts of data from a source database for further processing. To speed up Oracle data retrieval via SSIS or SSDT, the ODP.NET FetchSize can be increased from its default 128K value (131,072 bytes) to as large as int.MaxValue. The FetchSize determines the amount of data ODP.NET fetches into its internal cache upon each database round trip. It’s possible to improve performance by an order of magnitude by significantly increasing FetchSize when retrieving large result sets.

To increase the FetchSize, modify the .NET machine.config file. Modifying the machine.config requires Windows Administrator privileges. This file is generally located in the following directory: **C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\Config**.

Add an <oracle.manageddataaccess.client> section in the machine.config file for managed ODP.NET. For unmanaged ODP.NET, add an <oracle.unmanageddataaccess.client> section. This section should be placed within the <configuration> section and after the <configSections> </configSections>. Here’s an example setting the FetchSize to 4 MB:

```
<configuration>
 <configSections>
…
  </configSections>

 <oracle.manageddataaccess.client>
  <version number="*">
   <settings>
    <setting name="FetchSize" value="4194304" />
   </settings>
  </version>
 </oracle.manageddataaccess.client>

</configuration>
```