Table of Contents

Index   0

1   What's New

05-Jan-22 New features in ODBC Driver for Oracle 4.2

- Added support for Windows 11

- Improved compatibility with FICO Mosel

- Improved compatibility with FileMaker

- Improved compatibility with JMP on macOS

- Improved support for an ODBC installer on Windows 2000

08-Jul-21 New features in ODBC Driver for Oracle 4.1

- MSI installer for deploying through GPO is added

25-May-21 New features in ODBC Driver for Oracle 4.0

- Apple Silicon M1 is supported

- Oracle 21c is supported

- Compatibility with macOS Big Sur is improved

07-Dec-20 New features in ODBC Driver for Oracle 3.3

- Oracle 20c is supported

- Connection via SSL protocol is supported

- Connection via SSH protocol is supported

- Connection via HTTP tunnel is supported

08-Apr-20 New features in ODBC Driver for Oracle 3.2

- Long database object names are supported

- Possibility of mapping DATE time to SQL_TYPE_DATE rather than SQL_TYPE_TIMESTAMP is added

- Compatibility with Power Pivot is improved

- Compatibility with Visual Basic is improved

- Compatibility with RStudio is improved

04-Oct-19 New features in ODBC Driver for Oracle 3.1

- Oracle 19c is supported

- Now ODBC driver is thread-safe

- Now ODBC driver activation does not require administrator privileges

- Improved compatibility with sandboxed applications for macOS

- Support for connection pooling is improved

08-Jan-19 New features in ODBC Driver for Oracle 3.0

- Oracle 18c is supported

- Now ODBC driver for macOS is distributed as a PKG package

- Now ODBC driver for Linux is distributed as DEB and RPM packages

- Possibility to force the ODBC 2.x behavior is added

14-Jun-18 New features in ODBC Driver for Oracle 2.4

- Possibility to return String Types as Ansi or Unicode is added

- Compatibility with MS Access is improved

- Compatibility with Tableau is improved

- Compatibility with Omnis Studio is improved

- Compatibility with Power Pivot is improved

- Compatibility with DBeaver is improved

26-Feb-18 New features in ODBC Driver for Oracle 2.3

- Performance of batch operations is significantly improved

23-Oct-17 New features in ODBC Driver for Oracle 2.2

- Compatibility with SAS JMP is improved

- Compatibility with MS Power Query is improved

- OUTER JOIN macros in SQL queries are supported

- DateTime macros in SQL queries are supported

- Scalar function macros in SQL queries are supported

- NChar literal replacement is supported

- Retrieving Metadata for Oracle 8 is improved

26-Oct-16 New features in ODBC Driver for Oracle 2.1

- Oracle Cloud in the Direct mode is supported

- Option for showing meta data only for the current scheme is added

- Compatibility with MS Visual Studio

- Compatibility with MS FoxPro is improved

- Compatibility with MapInfo is improved

- Compatibility with Libre Office is improved

- Compatibility with Qlik is improved

- Compatibility with Delphi & C++Builder is improved

- MS Access linked tables support is improved

05-Apr-16 New features in ODBC Driver for Oracle 2.0

- Linux is supported

- macOS is supported

- Support for stored procedures and functions is improved

- Backward compatibility of SQLExecDirect with ODBC 2.x is improved

- Compatibility with MS Excel is improved

- Compatibility with ODBC 2.x is improved

- Bug with Trial expiration in Microsoft SQL Server Management Studio is fixed

- Bug with fetching big data amount is fixed

06-Nov-15 New features in ODBC Driver for Oracle 1.2

- Compatibility with Microsoft Visual Studio is improved

- Compatibility with Microsoft Office is improved

- Compatibility with Microsoft SQL Server Management Studio is improved

- Compatibility with Crystal Reports is improved

- Compatibility with ClikView is improved

- Stored procedures metadata support is improved

- Interval data types support is improved

16-Jul-15 New features in ODBC Driver for Oracle 1.1

- Performance is improved

20-May-15 New features in ODBC Driver for Oracle 1.0

- First release of ODBC Driver for Oracle

- Windows 32-bit is supported

- Windows 64-bit is supported

    Request Support   ODBC Forum   Provide Feedback

2   General Information

    Request Support   ODBC Forum   Provide Feedback

2.1   Overview

Overview

Devart ODBC Driver for Oracle is a high-performance connectivity solution with enterprise-level features for accessing Oracle databases from ODBC-compliant reporting, analytics, BI, and ETL tools on both 32-bit and 64-bit Windows, macOS, and Linux. Our ODBC driver fully supports standard ODBC API functions and data types and enables easy and secure access to live Oracle data from anywhere.
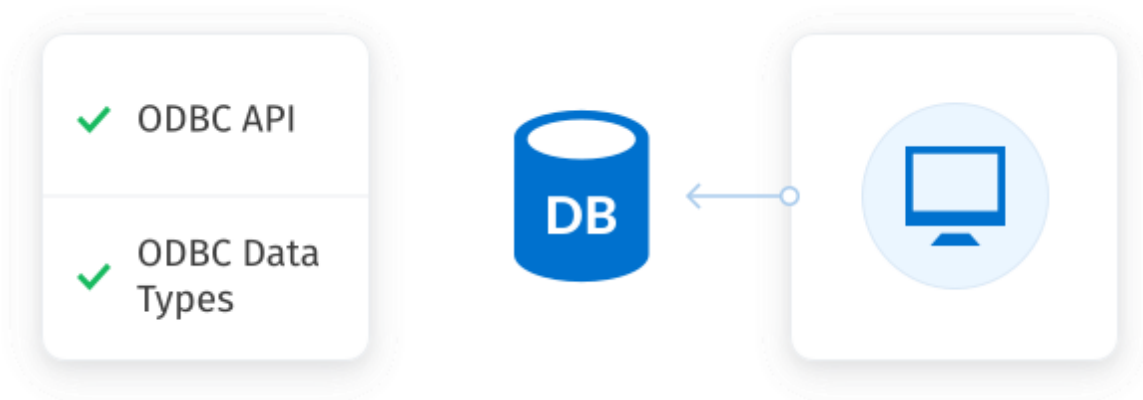
Direct Connection

Our data connector enables various ODBC-aware applications to establish a direct connection to Oracle via TCP/IP to eliminate the need for Oracle Client. Direct connection increases the speed of data transmission between an external application and Oracle, which is crucial for data analysis in real time. It also streamlines the deployment process, since there is no need to distribute any additional client software with the driver. You can also connect to Oracle using the native Oracle client library.

Compatibility

ODBC Driver for Oracle supports the following versions of Oracle:

- Oracle Server: 21c, 19c, 18c, 12c, 11g, 10g, 9i, 8i, 8.0, including Oracle Express Edition of 18c, 11g, and 10g
- Oracle Client (x86 and x64 versions): 21c, 19c, 18c, 12c, 11g, 10g, 9i, 8i, 8.0; Oracle Instant Client
- Oracle Cloud

Note: Support for the x64 version of Oracle Client is only available for 64-bit Windows.

© 2015-2022 Devart. All Rights Reserved.

Request Support   ODBC Forum   Provide Feedback

2.2   Features
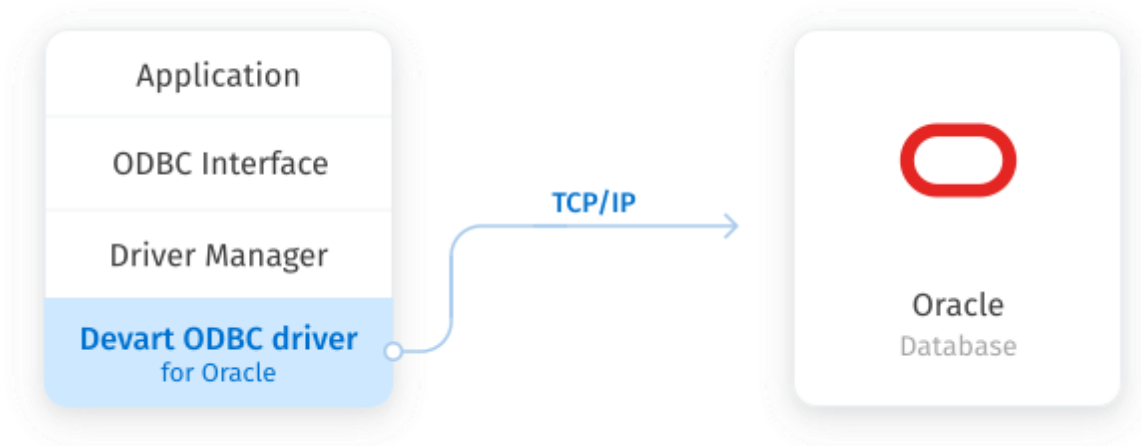
Direct Connection

Database applications based on our solution get an opportunity to establish connection to Oracle by means of native Oracle client software and in **Direct Mode**. Direct Mode gives your applications an unrivaled advantage - connection to Oracle databases directly via TCP/IP avoiding Oracle Client. That improves performance of your applications, their quality, reliability and especially the deployment process, since there is no need to supply additional client software together with your application.
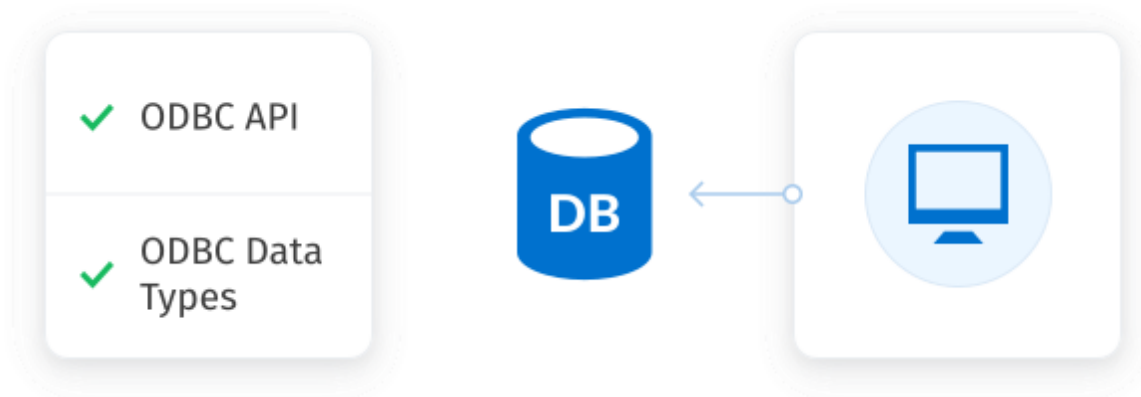
ODBC Conformance

Our ODBC driver provides full support for common ODBC interface:

- ODBC Data Types support

- ODBC API Functions support

In addition, we provide support for Advanced Connection String parameters. Thus allowing any desktop and web applications to connect to Oracle from various environments and platforms, that support ODBC.
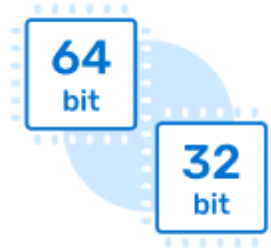
Development Platforms Variety

ODBC Driver for Oracle doesn't limit your choice of the development platform and environment. The driver installations are available for various operational systems and platforms. The current version supports Windows, macOS, Linux, both 32-bit and 64-bit. So you can develop both 32-bit and 64-bit cross-platform applications.

Database Compatibility

ODBC Driver for Oracle supports Oracle servers: 21c, 19c, 18c, 12c, 11g, 10g, 9i, 8i, 8.0, including Oracle Express Edition 11g and 10g; as well as Oracle Cloud.

ODBC Driver for Oracle supports both x86 and x64 versions of the following Oracle Clients: 21c, 19c, 18c, 12c, 11g, 10g, 9i, 8i, 8.0, Oracle Instant Client.

Note that support for x64 versions of Oracle Clients is available for 64-bit Windows.

High Performance

All our products are designed to help you write high-performance, lightweight data access layers, therefore they use advanced data access algorithms and techniques of optimization.

Support

Visit our Support page to get instant help from knowledgeable and experienced professionals, a quick resolution of your problems, and nightly builds with hotfixes.

© 2015-2022
Devart. All Rights
Reserved.

Request Support   ODBC Forum   Provide Feedback

2.3   Compatibility

Oracle Compatibility

ODBC Driver for Oracle supports the following database servers:

Oracle Clouds	Support

Oracle Cloud	✔

Oracle Servers	Support
21c	✔
19c	✔
18c (including Express Edition)	✔
12c	✔
11g (including Express Edition)	✔
10g (including Express Edition)	✔
9i	✔
8i	✔
8.0	✔

Oracle Clients	Support
21c	✔
19c	✔
18c	✔
12c	✔
11g	✔
10g	✔
9i	✔
8i	✔
8.0	✔
Oracle Instant Client	✔

Supported Platforms

- Windows x86 and x64 (including Windows Terminal Server)

- macOS x64 and ARM (Apple Silicon M1)

- Linux x86 and x64

Compatibility with Third-Party Tools

Application Development Tools

Adobe ColdFusion	✔
Embarcadero Delphi & C++Builder UniDAC, FireDAC, dbGo (ADO), BDE and dbExpress	✔
FileMaker	✔
Lazarus	✔
Microsoft Visual FoxPro	✔
Microsoft Visual Studio Server Explorer and ADO.NET ODBC Provider	✔
Omnis Studio	✔
PHP	✔
PowerBASIC	✔
Python	✔

Database Management

Aqua Data Studio	✔
dbForge Studio	✔
dBeaver	✔
EMS SQL Management Studio	✔

Informatica Cloud	✔
RazorSQL	✔
SQL Server Data Tools	✔
SQL Server Management Studio	✔
SQL Server Reporting Services	✔

BI & Analytics Software

Alteryx	✔
DBxtra	✔
Dundas BI	✔
IBM SPSS Statistics	✔
MicroStrategy	✔
Power BI	✔
Qlik Sense	✔
QlikView	✔
RStudio	✔
SAP Crystal Reports	✔
SAS JMP	✔
Tableau	✔
TARGIT	✔
TIBCO Spotfire	✔

Office Software Suites

LibreOffice	✓
Microsoft Access	✓
Microsoft Excel	✓
OpenOffice	✓
StarOffice	✓

Request Support   ODBC Forum   Provide Feedback

2.4   Requirements

No additional client software is required in the `Direct` mode. If you use the driver in the `OCI` mode, Oracle Client must be installed on the client machine.

You can have only one version of ODBC Driver for Oracle installed on your system.

Request Support   ODBC Forum   Provide Feedback

2.5   Licensing

ODBC Driver License Agreement

PLEASE READ THIS LICENSE AGREEMENT CAREFULLY. BY INSTALLING OR USING THIS SOFTWARE,YOU INDICATE ACCEPTANCE OF AND AGREE TO BECOME BOUND BY THE TERMS AND CONDITIONS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT INSTALL OR USE THIS SOFTWARE AND PROMPTLY RETURN IT TO DEVART.

INTRODUCTION

This Devart end-user license agreement ("Agreement") is a legal agreement between you (either an individual person or a single legal entity) and Devart, for the use of the ODBC Driver software application, demos, intermediate files, printed materials, and online or electronic documentation contained in this installation file. For the purpose of this Agreement, the

software program(s) and supporting documentation will be referred to as the "Software".

LICENSE

1. GRANT OF LICENSE

The enclosed Software is licensed, not sold. You have the following rights and privileges, subject to all limitations, restrictions, and policies specified in this Agreement.

1.1. If you are a legally licensed user, depending on the license type specified in the registration letter you have received from Devart upon purchase of the Software:

- the "Desktop License" allows you to install and use the Software on a single desktop computer, provided it is accessed by no more than one person at a time, either directly or remotely, for sole purposes only in accordance with this Agreement. If more than one person can simultaneously use the computer where you plan to install the product,you must purchase a Server License. A Desktop License is valid for one single desktop installation;

- the "Server License" allows you to install and use the Software on a single server,provided it is accessed by more than one person at a time, either directly or remotely.This definition includes, but is not limited to, Web servers, application servers,batch servers, and desktop workstations, where more than one concurrent users can access the Software. A Server License is valid for one single server installation,provided it is used by 1 (one) legal entity in accordance with this Agreement.

1.2. If you are a legally licensed user of the Software, you are also entitled to:

- make one copy of the Software for archival purposes only, or copy the Software onto the hard disk of your computer and retain the original for archival purposes;

- develop and test Applications with the Software, subject to the Limitations below.

1.3. You are allowed to use evaluation versions of the Software as specified in the Evaluation section.

No other rights or privileges are granted in this Agreement.

2. LIMITATIONS

Only legally registered users are licensed to use the Software, subject to all of the conditions of this Agreement. Usage of the Software is subject to the following restrictions.

2.1. You may not reverse engineer, decompile, or disassemble the Software.

2.2. You may not reproduce or distribute any Software documentation without express written permission from Devart.

2.3. You may not distribute and sell any portion of the Software integrating it into your Applications.

2.4. You may not transfer, assign, or modify the Software in whole or in part. In particular, the Software license is non-transferable, and you may not transfer the Software installation package.

2.5. You may not remove or alter any Devart's copyright, trademark, or other proprietary rights notice contained in any portion of Devart files.

3. REDISTRIBUTION

The license grants you a non-exclusive right to reproduce any new software programs(Applications) created using the Software. You cannot distribute the Software integrated into your Applications. Any Devart's files remain Devart's exclusive property.

4. TRANSFER

You may not transfer the Software to any individual or entity without express written permission from Devart. In particular, you may not share copies of the Software under "Desktop License" with other co-developers without obtaining proper license of these copies for each individual; you may not install the Software under "Server License" more than 1 (one) server without obtaining proper license of these installations for each server.

5. TERMINATION

Devart may immediately terminate this Agreement without notice or judicial resolution in the event of any failure to comply with any provision of this Agreement. Upon such termination you must destroy the Software, all accompanying written materials, and all copies.

6. EVALUATION

Devart may provide evaluation ("Trial") versions of the Software. You may transfer or distribute Trial versions of the Software as an original installation package only.If the Software you have obtained is marked as a "Trial" version, you may install and use the Software for a period of up to 30 calendar days from the date of installation(the "Trial Period"), subject to the additional restriction that it is used solely for evaluation of the Software and not in conjunction with the development or deployment of any application in production. You may not use

Applications developed using Trial versions of the Software for any commercial purposes. Upon expiration of the Trial Period, the Software must be uninstalled, all its copies and all accompanying written materials must be destroyed.

7. WARRANTY

The Software and documentation are provided "AS IS" without warranty of any kind.Devart makes no warranties, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or use.

8. SUBSCRIPTION AND SUPPORT

The Software is sold on a subscription basis. The Software subscription entitles you to download improvements and enhancement from Devart's web site as they become available, during the active subscription period. The initial subscription period is one year from the date of purchase of the license. The subscription is automatically activated upon purchase, and may be subsequently renewed by Devart, subject to receipt applicable fees. Licensed users of the Software with an active subscription may request technical assistance with using the Software over email from the Software development. Devart shall use its reasonable endeavors to answer queries raised, but does not guarantee that your queries or problems will be fixed or solved.

Devart reserves the right to cease offering and providing support for legacy IDE versions.

9. COPYRIGHT

The Software is confidential and proprietary copyrighted work of Devart and is protected by international copyright laws and treaty provisions. You may not remove the copyright notice from any copy of the Software or any copy of the written materials,accompanying the Software.

This Agreement contains the total agreement between the two parties and supersedes any other agreements, written, oral, expressed, or implied.

[Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

2.6   Getting Support

This document lists several ways you can find help with using ODBC Driver for Oracle

describes the Priority Support program.

Support Options

There are a number of resources for finding help on installing and using ODBC Driver for Oracle:

- You can find out more about ODBC Driver for Oracle installation or licensing by consulting Installation and License articles of this manual respectively.

- You can get community assistance and technical support on the Community Forum.

- You can get advanced technical assistance by ODBC Driver for Oracle developers through the ODBC Driver for Oracle Priority Support program.

Subscriptions

The ODBC Driver for Oracle Subscription program is an annual maintenance and support service for ODBC Driver for Oracle users.

Users with a valid ODBC Driver for Oracle Subscription get the following benefits:

- Product support through the ODBC Driver for Oracle Priority Support program

- Access to new versions of ODBC Driver for Oracle when they are released

- Access to all ODBC Driver for Oracle updates and bug fixes

- Notifications about new product versions

Priority Support

ODBC Driver for Oracle Priority Support is an advanced product support service for getting expedited individual assistance with ODBC Driver for Oracle-related questions from the ODBC Driver for Oracle developers themselves. Priority Support is carried out over email and has a two business day response policy. Priority Support is available for users with an active ODBC Driver for Oracle Subscription.

To get help through the ODBC Driver for Oracle Priority Support program, please send an email to odbc@devart.com describing the problem you are having. Make sure to include the following information in your message:

Your ODBC Driver for Oracle Registration number.

- Full ODBC Driver for Oracle edition name and version number. You can find the version

number in DLL version information.

- Versions of the Oracle server and client you are using.

- A detailed problem description.

- If possible, ODBC Administrator Log, scripts for creating and filling in database objects, and the application using ODBC Driver for Oracle.

If you have any questions regarding licensing or subscriptions, please see the FAQ or contact sales@devart.com

Request Support   ODBC Forum   Provide Feedback

3 Using ODBC Driver

1. Installation
2. Product Activation
3. Connecting to Oracle
4. Connection String Options
5. Secure Connections
6. Sandboxed Apps on macOS
7. Using with iODBC
8. Enabling ODBC Tracing
9. Supported Data Types
10. Supported ODBC API Functions

Request Support   ODBC Forum   Provide Feedback

3.1 Installation

ODBC Driver for Oracle currently supports the following platforms: Windows, macOS, and Linux, both 32-bit and 64-bit.

See how to install Devart ODBC Driver for Oracle :

- Windows

- Windows Silent

- [macOS](#)
- [Linux DEB Package](#)
- [Linux RPM Package](#)

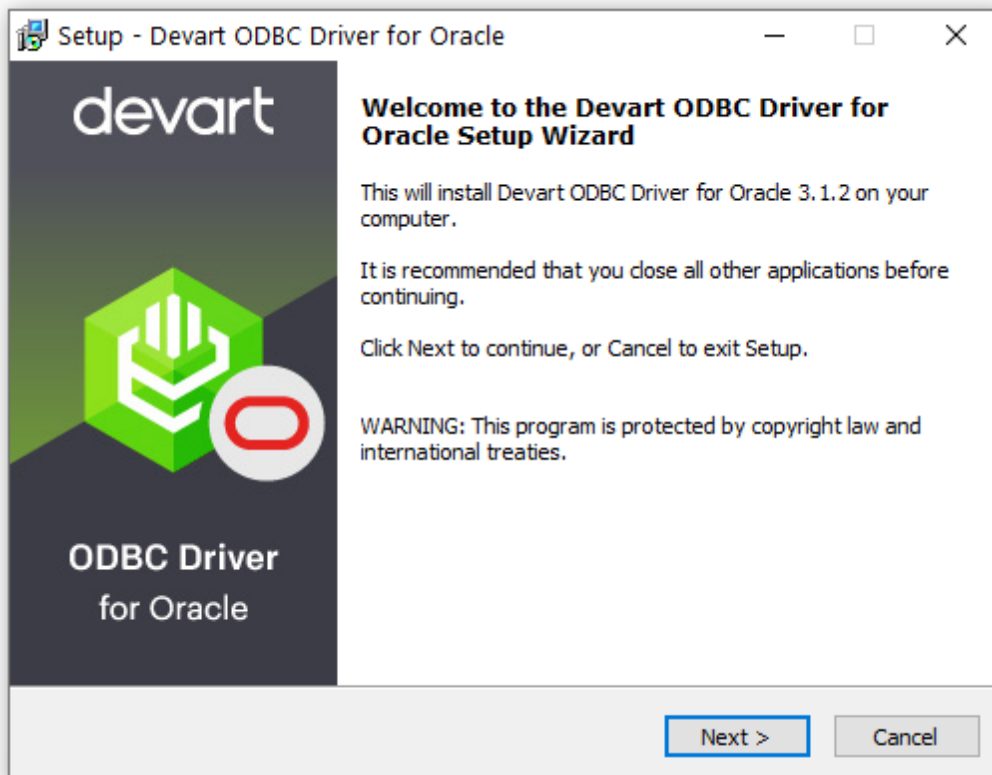   [Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

3.1.1   Windows

Installation

1. [Download](#) and run the installer.

2. Follow the instructions in the wizard.

3. In case if you already have the specified installation folder on the PC or another Driver

version is installed, you will get a warning. Click **Yes** to overwrite the old files with the

current installation, but it is recommended to completely uninstall the previous driver version first, and then install the new one.

4. On the Select Components page you can select whether to install the **64-bit** version of the driver or not. Clear the check box if you need no 64-bit installation. There is also a check box on this page, that allows you to select whether to install Help and Manual.
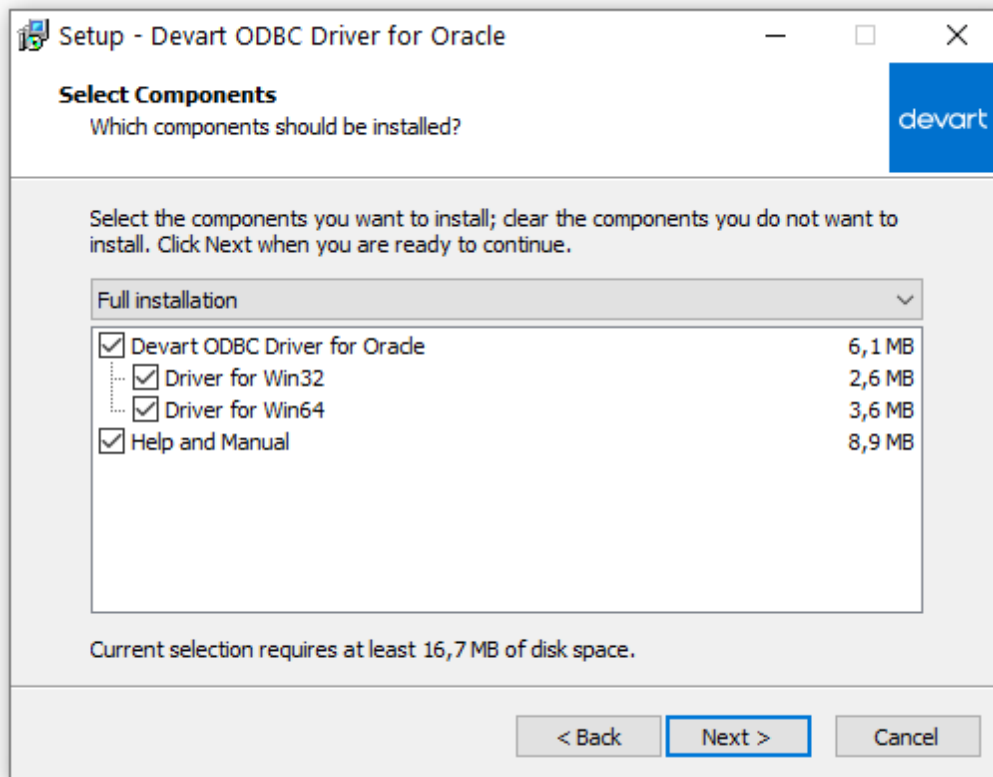
5. In the License Information dialog box, you should select the license type and activate the product. If you have no activation key, you can select Trial and use the driver for evaluation purposes.

6. If you have an activation key, select the Activation Key option. Copy the activation key from the registration email or your Customer Portal account and paste it into the Activation Key edit box.
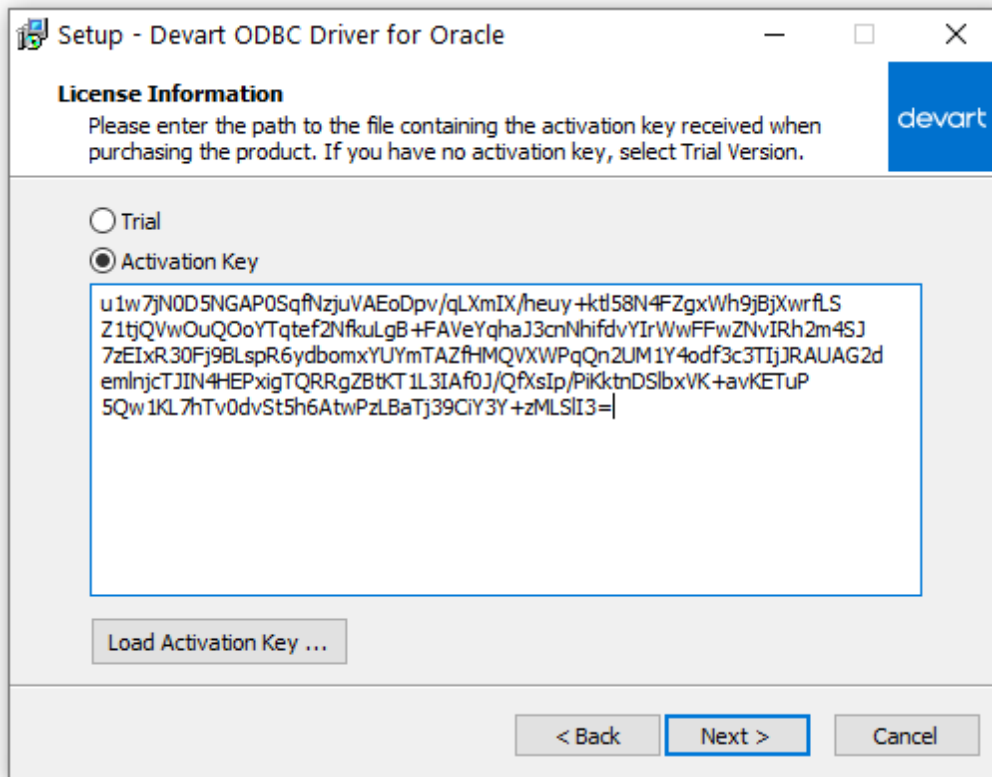
7. If you have the activation key file, click the Load Activation Key button and browse to it.
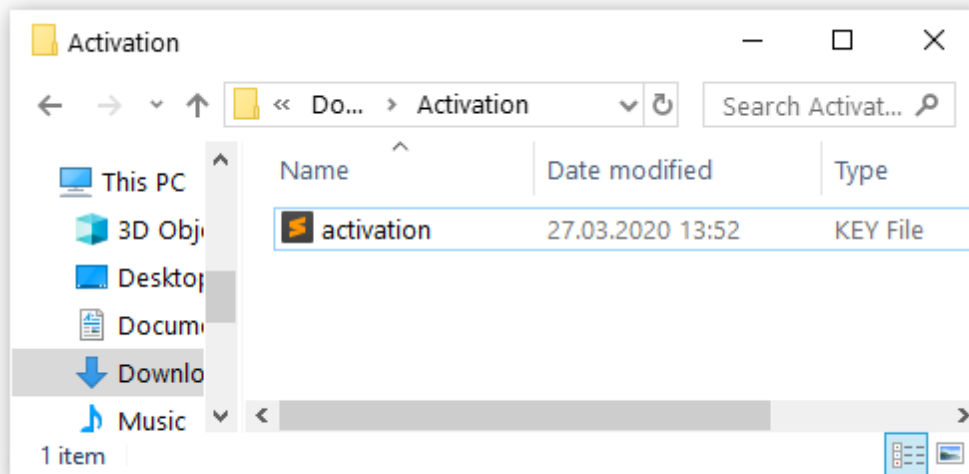
8. Click Next.

9. Click Install, then Finish.

10. After the installation is completed, you need to configure the driver.

See also:
- Installation on macOS
- Install Linux DEB package
- Install Linux RPM package

Request Support   ODBC Forum   Provide Feedback

3.1.2   Windows Silent

Silent Installation with OEM license on Windows

1. Run the Command Prompt as an administrator.

2. Use the following command-lines to perform the driver silent/very silent installation:

```
DevartODBCOracle.exe /SILENT /ActivationKey=y1c7nmgdu234laszxcvONGurjfhxm90L
```
```
DevartODBCOracle.exe /VERYSILENT /ActivationKey=ekhdh765mh09ukr237gfHRtrilwq
```

Note: The installation is performed by entering a license key.

```
DevartODBCOracle.exe /SILENT /ActivationFile=d:\lic.key
```
```
DevartODBCOracle.exe /VERYSILENT /ActivationFile=d:\lic.key
```

Note: The installation is performed by specifying the path to a license key file with any name.

When /SILENT is used, the installation progress is displayed, but no user interaction is required during installation.

When /VERYSILENT is used, the installation wizard dialog is hidden and the installation process is performed without user interference.

Request Support   ODBC Forum   Provide Feedback

3.1.3   macOS

Prerequisites

ODBC Driver for Oracle works under control of an ODBC driver manager. ODBC driver manager is not distributed along with our driver and must be installed separately.

ODBC Driver for Oracle is compatible with iODBC driver manager.

In case when using other ODBC driver managers, ODBC Driver for Oracle will be installed, but it will require manual modification of configuration files of these managers.
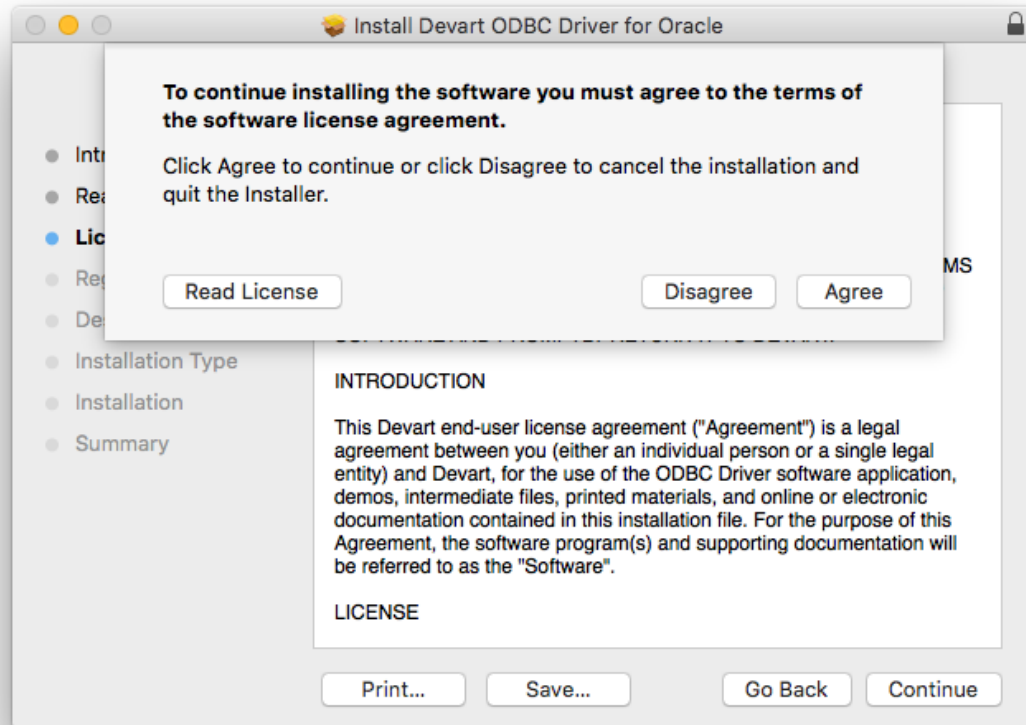
Installation

1. Download the PKG file from the Devart website.

2. Run the downloaded file and follow the instructions in the wizard.
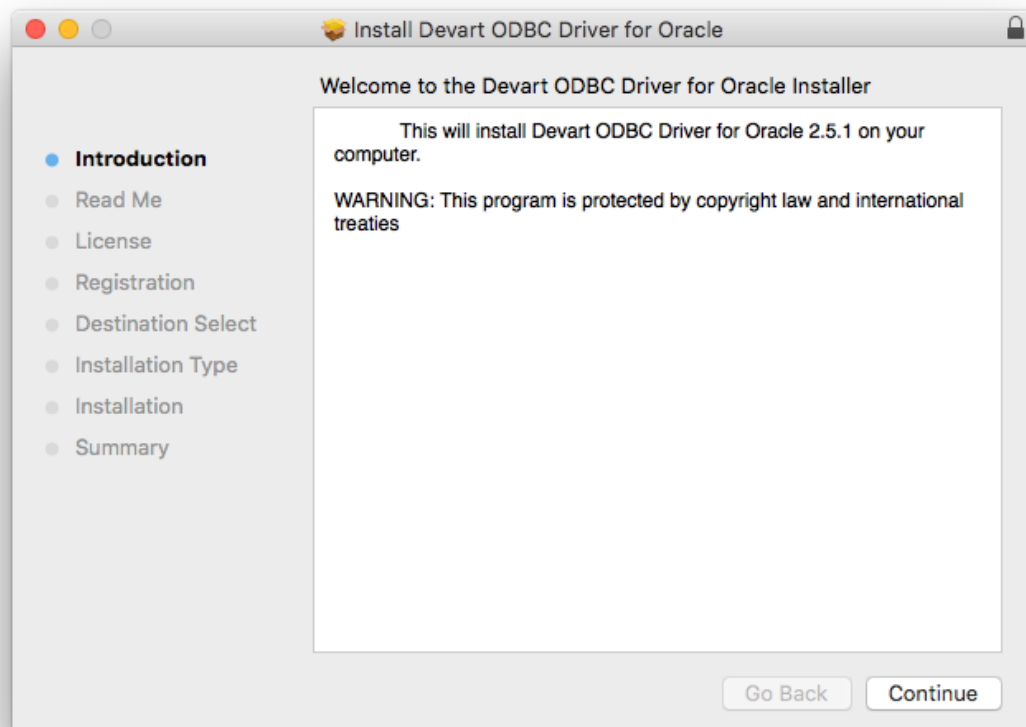
3. After reading the license agreement, click Agree to proceed with the installation.


4. In the License Information dialog box, you should select the license type and activate the product. If you have no activation key, you can select Trial and use the driver for evaluation purposes.

5. If you have an activation key, select the Activation Key option. Copy the activation key from the registration email or your Customer Portal account and paste it into the Activation Key edit box.
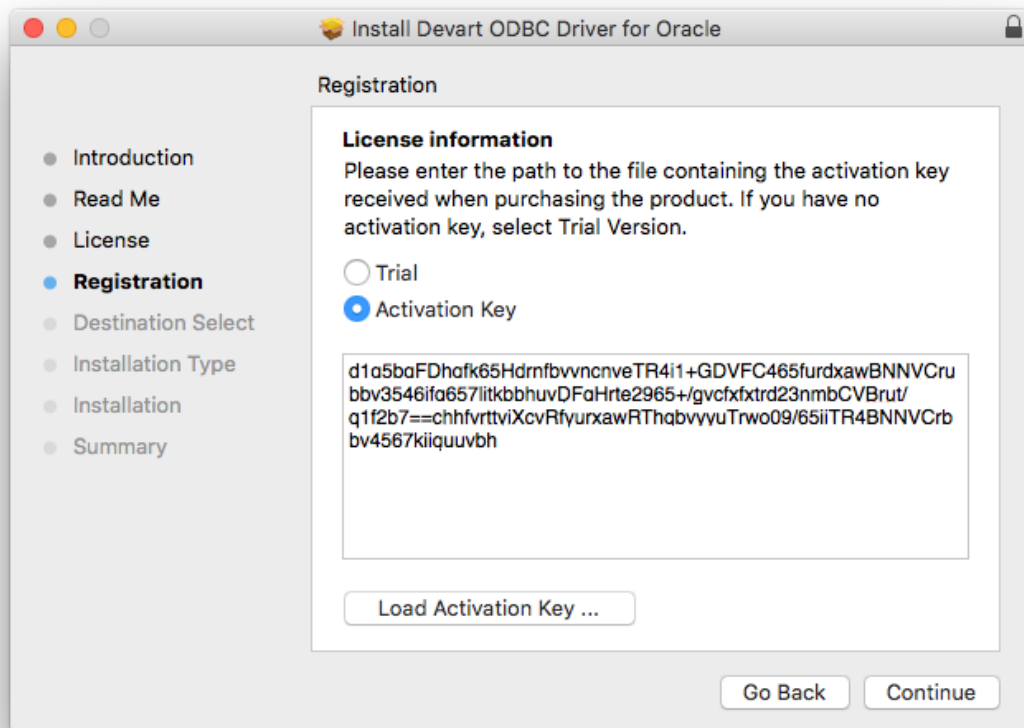
6. If you have the activation key file, click the Load Activation Key button and browse to it.
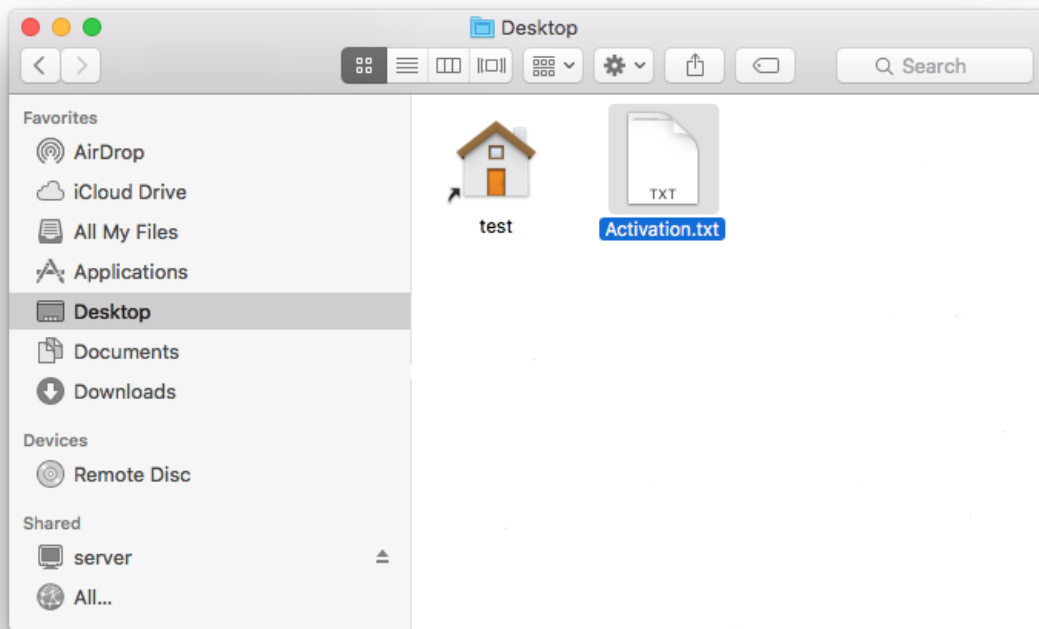
7. Click Continue.

8. Click Install.

9. Driver is installed successfully.
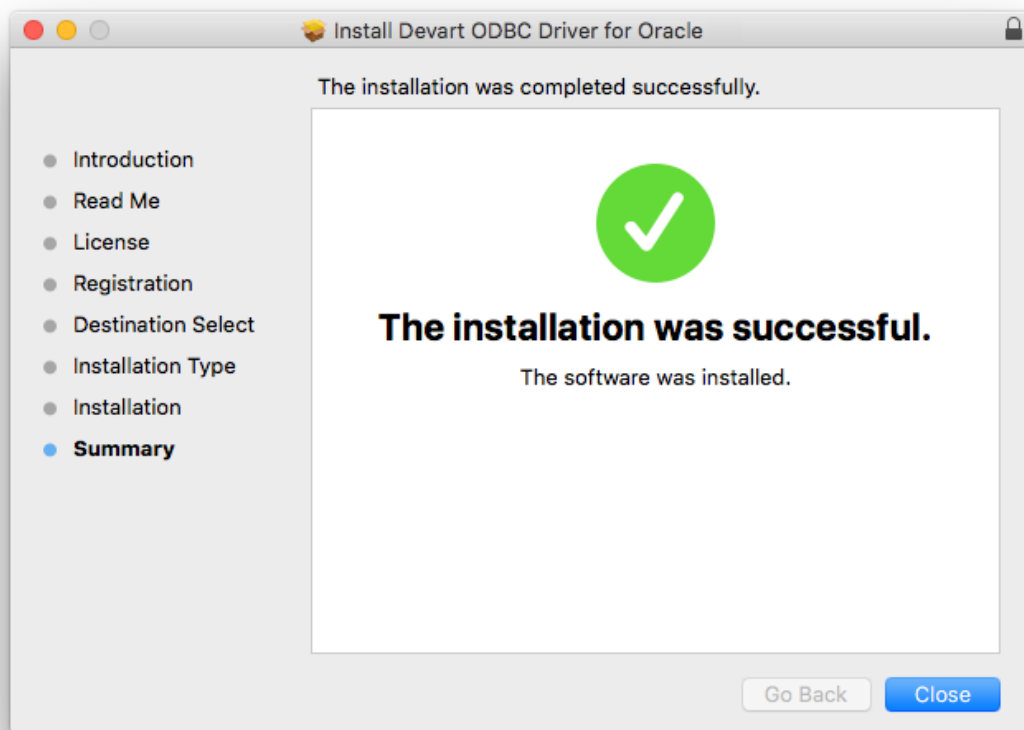
To activate the driver, perform the steps described in the [Product Activation](#) article.
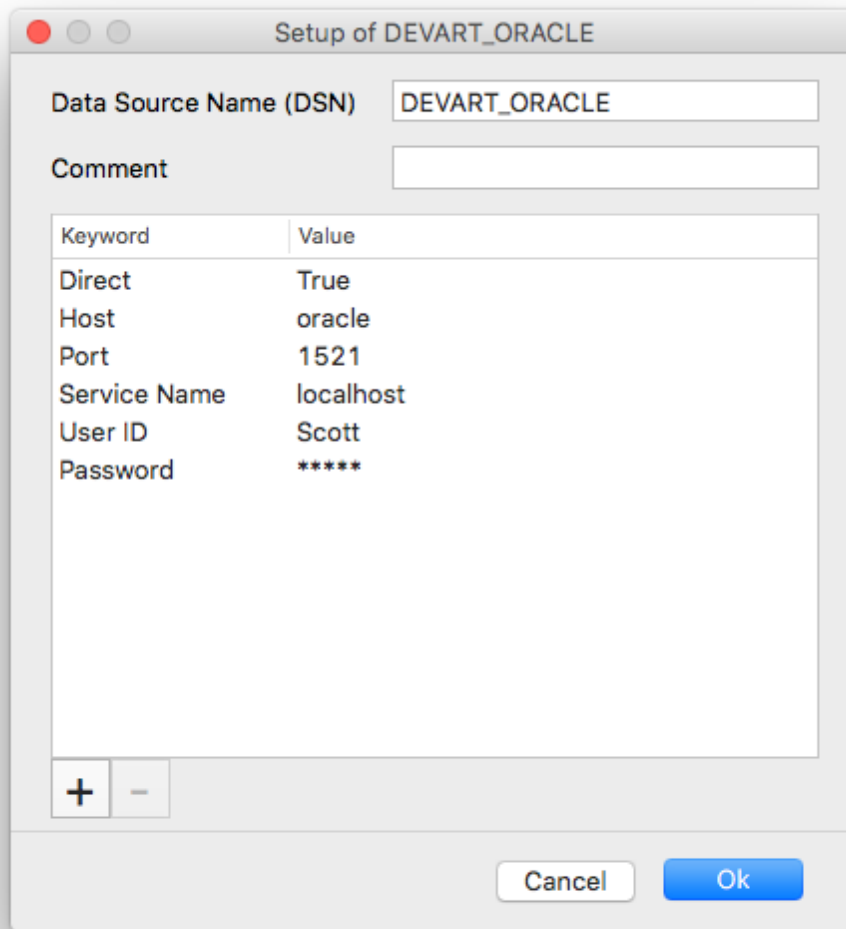
Test connection

After the driver is installed, DSN with the name DEVART_ORACLE is created. You can use it to test a connection with ORACLE server. For this, perform the following steps:

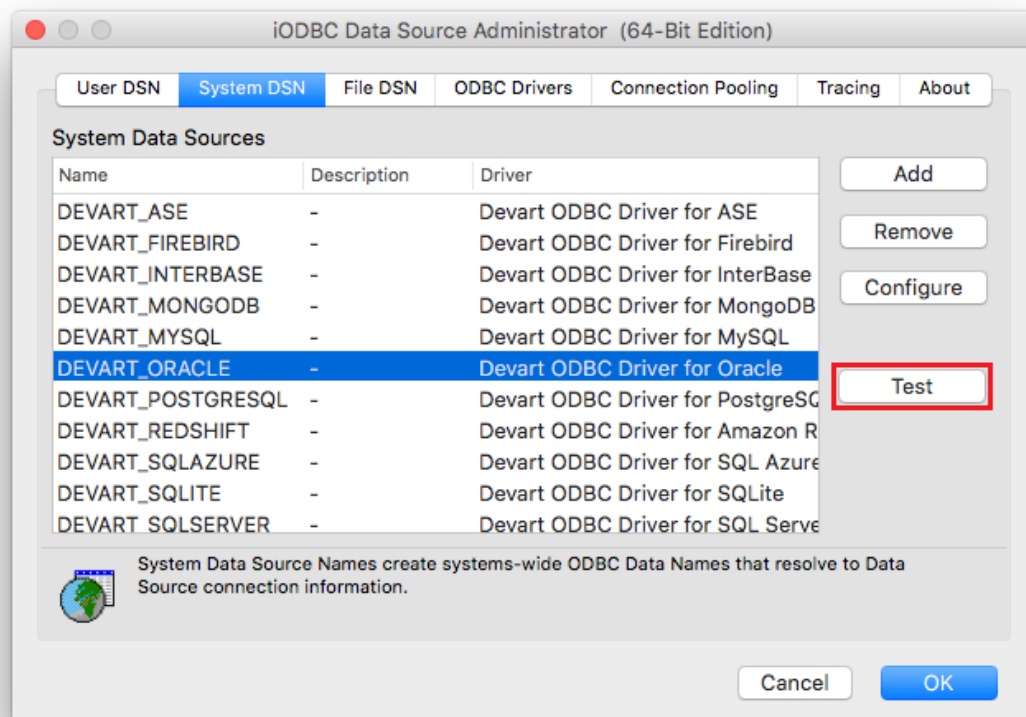1. Run the iODBC utility of the required bitness. Find the DEVART_ORACLE section and click the Configure button:
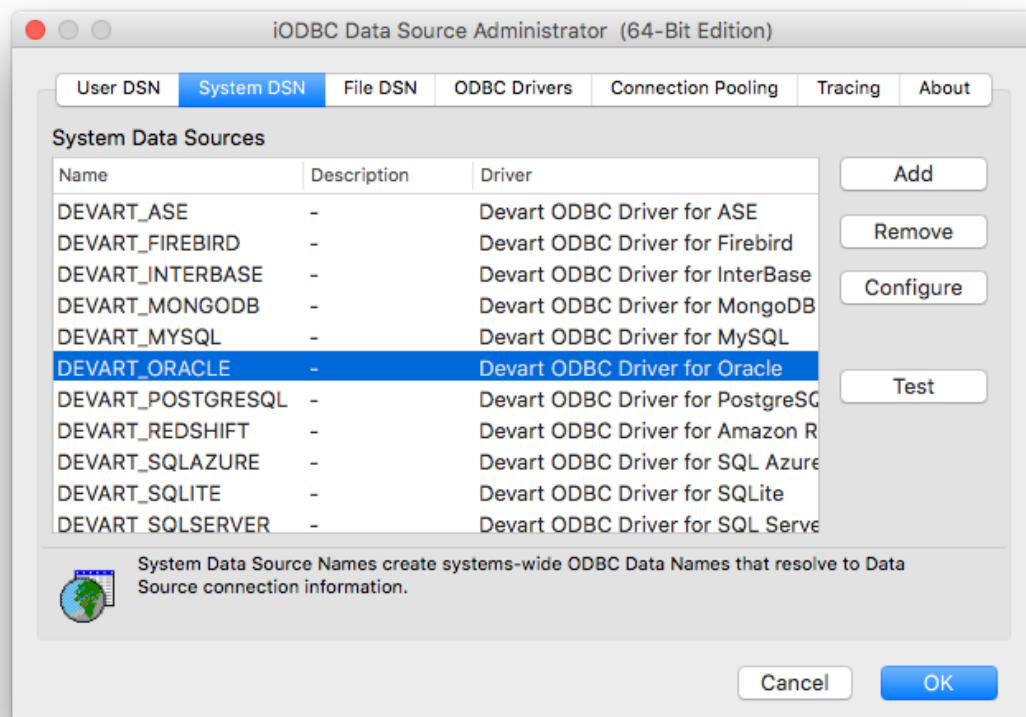
2. In the appeared dialog, specify the required connection settings and click OK.

3. Now click the Test button to establish a test connection to your data source.

See also:

- [Installation on Windows](#)
- [Install Linux DEB package](#)
- [Install Linux RPM package](#)

[Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

3.1.4   Linux DEB Package

Prerequisites

ODBC Driver for Oracle works under control of an ODBC driver manager. ODBC driver manager is not distributed along with our driver and must be installed separately.

ODBC Driver for Oracle is compatible with [UnixODBC](#) driver manager.

In case when using other ODBC driver managers, ODBC Driver for Oracle will be installed, but it will require manual modification of configuration files of these managers.

Installation

Let's consider how to install the Devart ODBC driver on Linux from a DEB package, for example, on Ubuntu. There are two ways to install the driver either manually or via the command line.

GUI installation

1. [Download](#) the DEB package of the required bitness from the Devart website.

2. Navigate to the folder with the downloaded package ("Downloads" by default) and double-click it.

3. In the opened dialog, click the **Install** button.
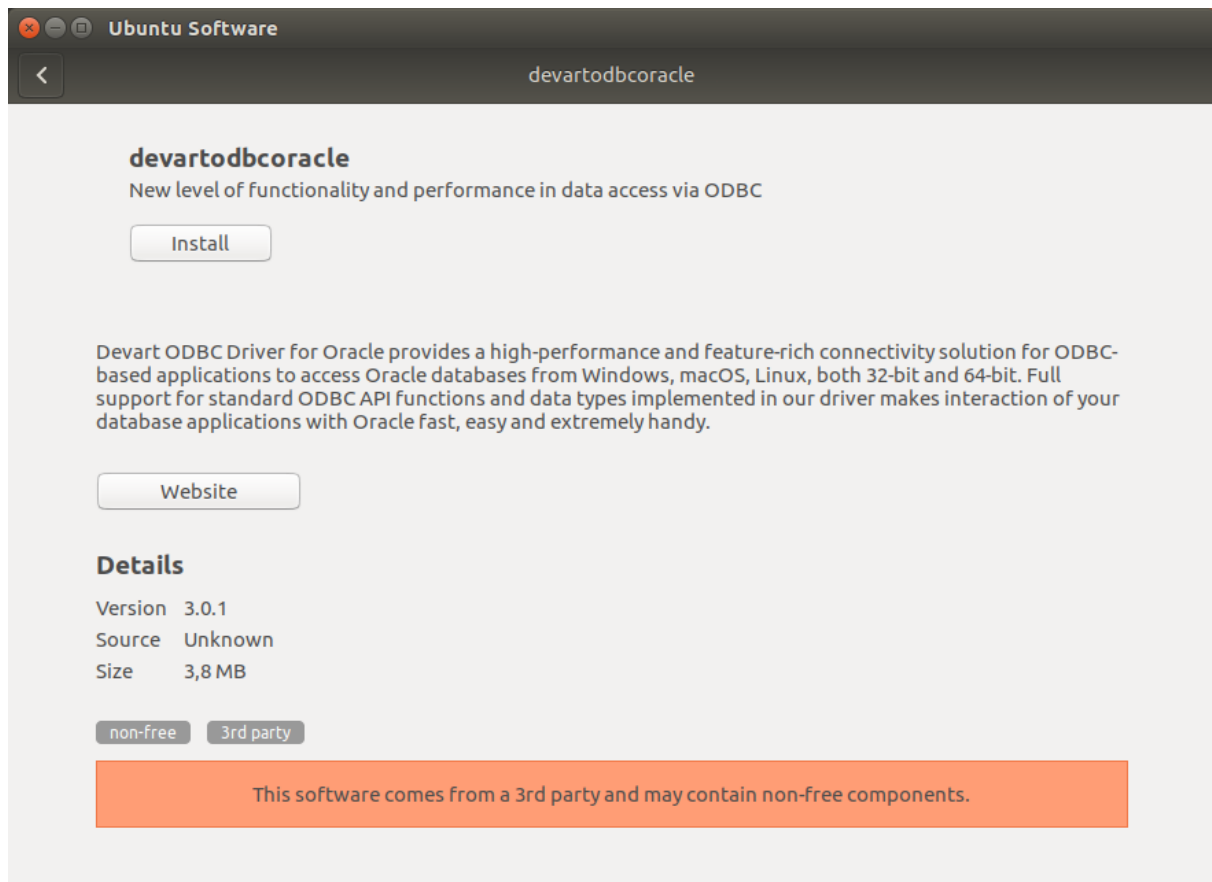
4. If the installation is successfully completed, the Install button changes into the Remove one.

Command-line installation

1. [Download](#) the DEB package from the Devart website.

By default the required package will be downloaded into the ~/Downloads folder (or the selected one);

2. Run the 'Terminal' program;

3. Navigate to the folder with the downloaded package cd ~/Downloads (if you downloaded the package into another folder, you need to specify the path to this folder as the cd command parameter):

```
cd ~/Downloads/
```

```
test@ubuntu:~$ cd ~/Downloads/
test@ubuntu:~/Downloads$
```

4. To install the devartodbcoracle_i386.deb on a 32-bit system, use the following command:

```
sudo dpkg -i devartodbcoracle_i386.deb
```

```
test@ubuntu:~$ cd ~/Downloads/
test@ubuntu:~/Downloads$ sudo dpkg -i devartodbcoracle_i386.deb
```

To install the devartodbcoracle_amd64.deb on a 64-bit system, use the following command:

```
sudo dpkg -i devartodbcoracle_amd64.deb
```

```
test@ubuntu:~$ cd ~/Downloads/
test@ubuntu:~/Downloads$ sudo dpkg -i devartodbcoracle_amd64.deb
```

5. Driver is installed successfully.

```
test@ubuntu:~/Downloads$ sudo dpkg -i devartodbcpostgresql_i386.deb
(Reading database ... 238056 files and directories currently installed.)
Preparing to unpack devartodbcpostgresql_i386.deb ...
Unpacking devartodbcpostgresql (3.0.1) over (2.5.1) ...
Setting up devartodbcpostgresql (3.0.1) ...
test@ubuntu:~/Downloads$
```

To activate the driver, perform the steps described in the Product Activation article.

Test connection

After the driver is installed, a DSN with the name DEVART_ORACLE is created. You can use it to test connection with ORACLE server. For this, perform the following steps:

1. Open the odbc.ini file located in the /etc folder. Find the DEVART_ORACLE section and specify the required connection settings:

```
User ID=<your Oracle User Name>
Password=<your Oracle password>
Host=<your Oracle Server Name>
Port=<your Oracle Listener Port>
Database=<your Oracle database name>
```

2. Run the UnixODBC Test Command utility and test a connection using the following command:

```
isql -v DEVART_ORACLE
```

```
test@ubuntu:~$ isql -v DEVART_ORACLE
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL>
```

See also:

- Install Linux RPM package

- Installation on Windows

- Installation on macOS

Request Support   ODBC Forum   Provide Feedback

3.1.5   Linux RPM Package

Prerequisites

ODBC Driver for Oracle works under control of an ODBC driver manager. ODBC driver manager is not distributed along with our driver and must be installed separately.

ODBC Driver for Oracle is compatible with UnixODBC driver manager.

In case when using other ODBC driver managers, ODBC Driver for Oracle will be installed, but it will require manual modification of configuration files of these managers.

Installation

Let's consider how to install the Devart ODBC driver on Linux from an RPM package, for example, on CentOS. To install the driver, you should download the .rpm package and install it via the command line. See the detailed description of these steps below:

1. [Download](#) the RPM package from the Devart website.

By default the required package will be downloaded into the ~/Downloads folder (or the selected one);

2. Run the 'Konsole' program;

3. Navigate to the folder with the downloaded package cd ~/Downloads (if you downloaded the package into another folder, you need to specify the path to this folder as the cd command parameter):

```
cd ~/Downloads/
```

```
[test@centos7x64 ~]$ cd ~/Downloads/
[test@centos7x64 Downloads]$ █
```

4. Use the following command to install the devart-odbc-oracle.i386.rpm on a 32-bit system:

```
sudo rpm -ivh devart-odbc-oracle.i386.rpm
```

```
[test@localhost ~]$ sudo rpm -ivh devart-odbc-oracle.i386.rpm
```

Use the following command to install the devart-odbc-oracle.x86_64.rpm on a 64-bit system:

```
sudo rpm -ivh devart-odbc-oracle.x86_64.rpm
```

```
[test@centos7x64 ~]$ cd ~/Downloads/
[test@centos7x64 Downloads]$ sudo rpm -ivh devart-odbc-oracle.x86_64.rpm █
```

5. Driver is installed successfully.

```
[test@centos7x64 ~]$ cd ~/Downloads/
[test@centos7x64 Downloads]$ sudo rpm -ivh devart-odbc-oracle.x86_64.rpm
Preparing...                          ############################### [100%]
Updating / installing...
   1:devart-odbc-oracle-3.0.1-1         ############################### [100%]
[test@centos7x64 Downloads]$ ▮
```

To activate the driver, perform the steps described in the Product Activation article.

Test connection

After the driver is installed, a DSN with the name DEVART_ORACLE is created. You can use it to test connection with ORACLE server. For this, perform the following steps:

1. Open the odbc.ini file located in the /etc folder. Find the DEVART_ORACLE section and specify the required connection settings:

```
User ID=<your Oracle User Name>
Password=<your Oracle password>
Host=<your Oracle Server Name>
Port=<your Oracle Listener Port>
Database=<your Oracle database name>
```

2. Run the UnixODBC Test Command utility and test a connection using the following command:

```
isql -v DEVART_ORACLE
```

```
[test@centos7x64 ~]$ isql -v DEVART_ORACLE
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL> █
```

See also:

- Install Linux DEB package

- Installation on Windows

- Installation on macOS

Request Support   ODBC Forum   Provide Feedback

3.2   Product Activation

See how to activate Devart ODBC Driver for Oracle :

- Obtaining Activation Key

- Activation on Windows

- Activation on macOS

- Activation on Linux

- Where to see the license information

Request Support   ODBC Forum   Provide Feedback

3.2.1 Obtaining Activation Key

To obtain a product activation key, follow these instructions:

1. After purchasing the license, you receive a registration email to the email address, specified when ordering the product.

2. This email contains a Driver Activation Key and Login Credentials for the Customer Portal. Keep this information secret.

3. You can copy the Activation Key either from the registration email or at the Customer Portal account.

4. To login to the Customer Portal, use your Username and Password from the registration email.

5. To obtain your Activation Key, click the View link on the right. You will get the following dialog box:

Activation Key   X

u1w7jC0D8NGAE0FqbNzjuTAToDjv/qLNmIX/heuy+kt2Yi5V39cp39ih89JZ6mCb
MhlQfsILhFJgr6hQWxTEmuz1Ji+IQycFEPkmX+Z5YQPXt7e5xqMUSHsUdPjVsKfs
93Ee03mqH6a9wJ9fSQXuKK8zPlxDI81to5ff5vmppjL8EXvIaOkPRh4araau9UfK
r64rzGQxuk5jd0NfxzeykQ==

COPY TO CLIPBOARD   CLOSE

6. Copy the Activation Key with the Copy to Clipboard button.

See also:

- Activation on Windows
- Activation on MacOS
- Activation on Linux

Request Support   ODBC Forum   Provide Feedback
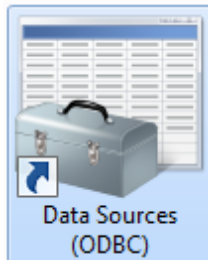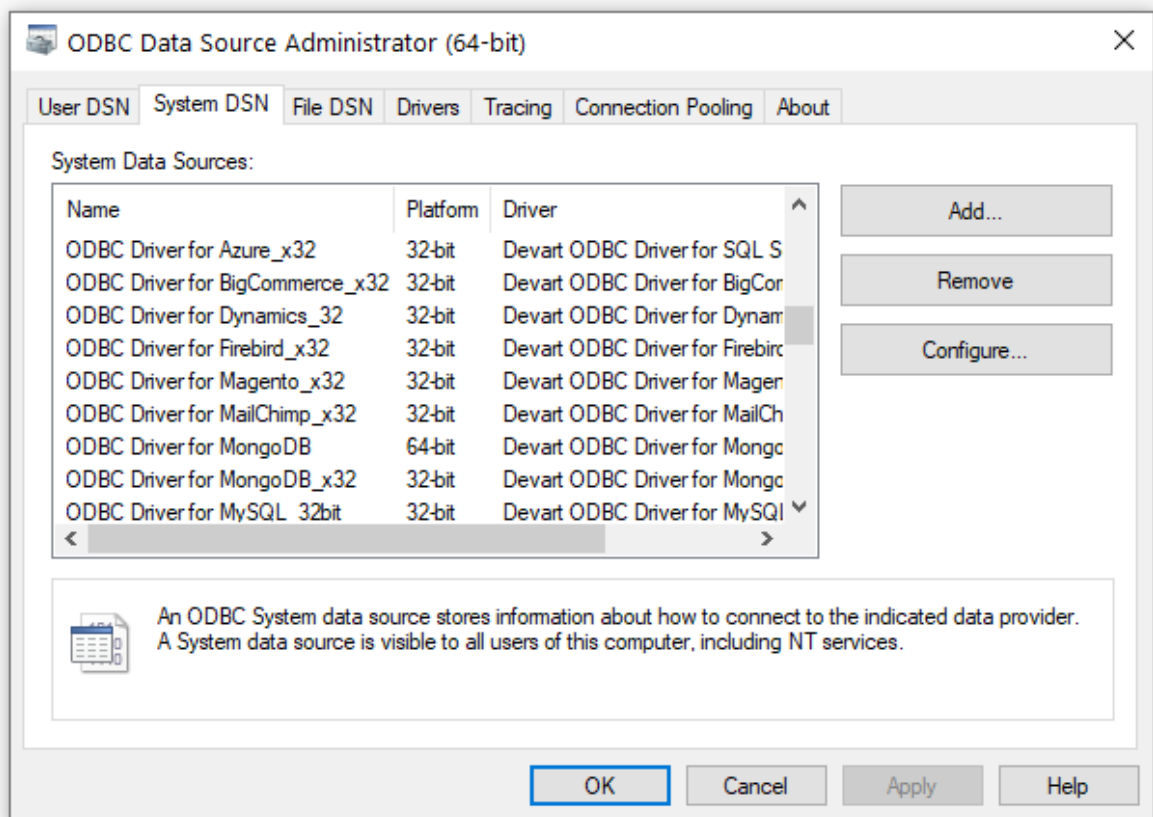
3.2.2 Activation on Windows

Driver Activation After Installation

To activate your installed driver using ODBC Administrator, perform the following steps:

1. Run ODBC Administrator.
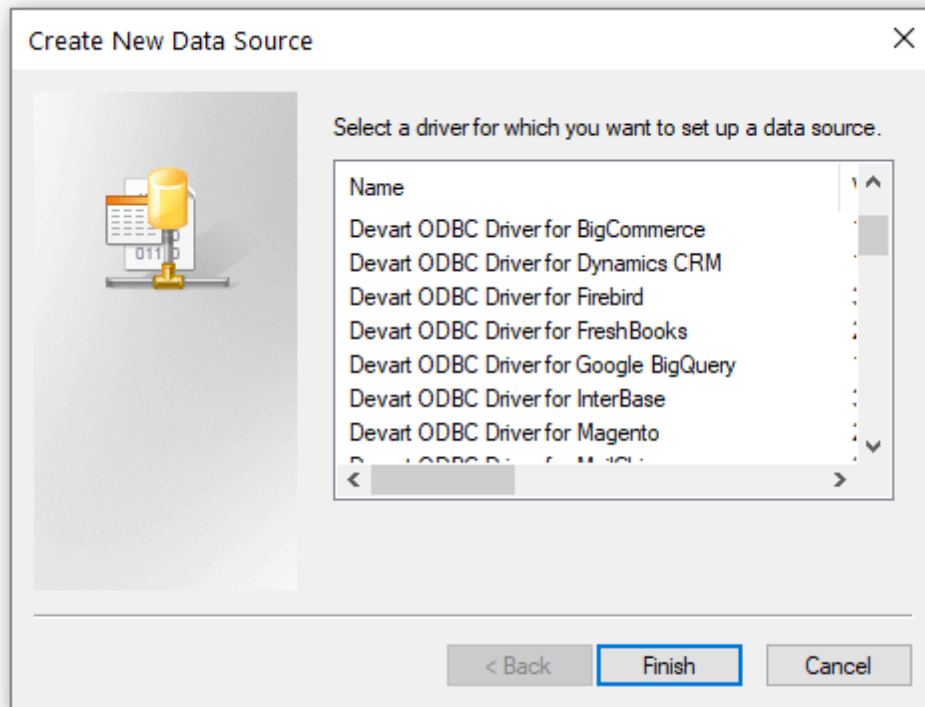

Data Sources
(ODBC)

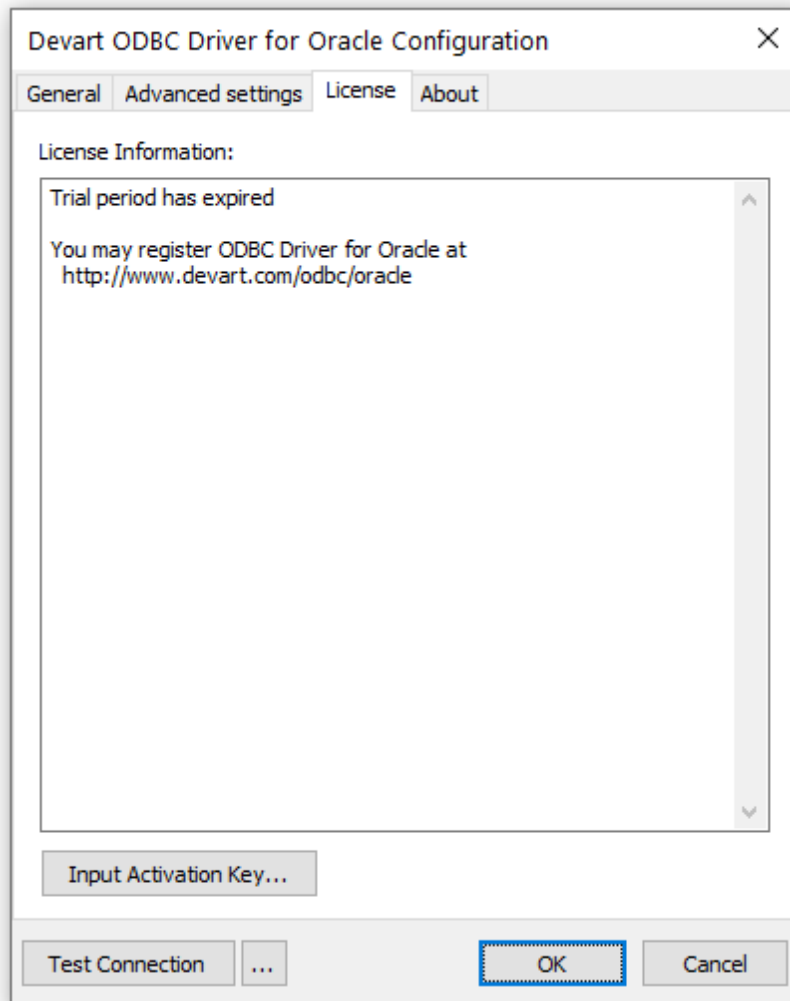2. In the System DSN tab click the Add button.

3. In the appeared dialog box, select the installed driver, click Finish.
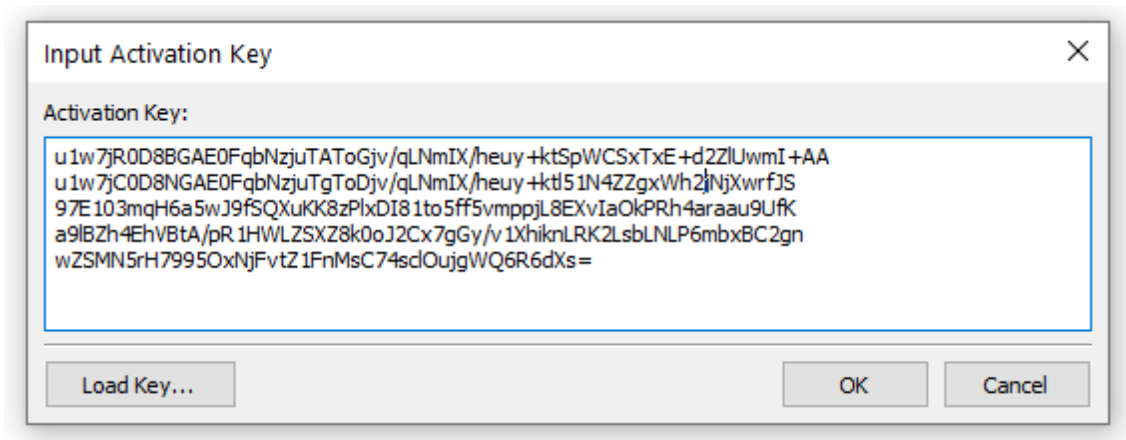
4. In the Driver Configuration dialog box, on the License tab, click the Input Activation Key button.

5. Copy the activation key from the registration email carefully and paste it into the Input

Activation Key edit box.

6. If you have the activation key file, click the Load Key button and browse to it.

7. Click OK.

Request Support   ODBC Forum   Provide Feedback

3.2.3 Activation on macOS

Driver Activation After Installation

If you don't activate your driver during installation, you can activate it later by following the
steps:

1. Create a file with the "activation.key" name.

2. Copy the activation key from the registration email or your Customer Portal account and

paste it into the created file.

3. Place the "activation.key" file into the folder where the driver was installed (for Devart

ODBC Driver for Oracle it is /Library/ODBC/Devart/Oracle by default).

See also:

- Activation on Windows
- Activation on Linux

Request Support   ODBC Forum   Provide Feedback

3.2.4 Activation on Linux

Driver Activation After Installation

If you did not activate the driver during installation, you can activate it later:

1. Create a file with the "activation.key" name.

2. Copy the activation key from the registration email or your Customer Portal account and

   paste it into the created file.

3. Place the "activation.key" file into the folder where the driver was installed:

- for the DEB package of Devart ODBC Driver for Oracle, it is */usr/share/devart/odbcoracle*
by default;

- for the RPM package of Devart ODBC Driver for Oracle, it is */usr/local/devart/odbcoracle* by
default.

See also:

- Activation on Windows

- Activation on macOS

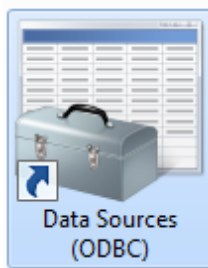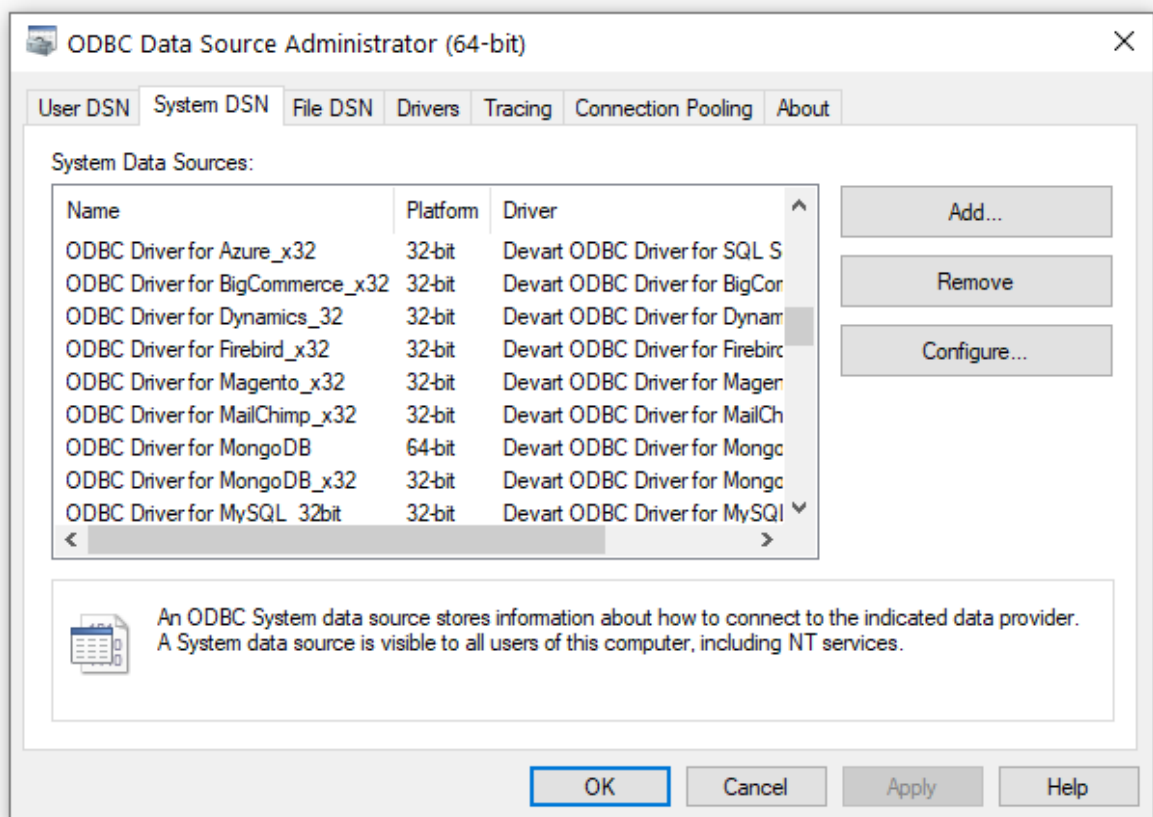Request Support   ODBC Forum   Provide Feedback

3.2.5 Where to See the License Information?

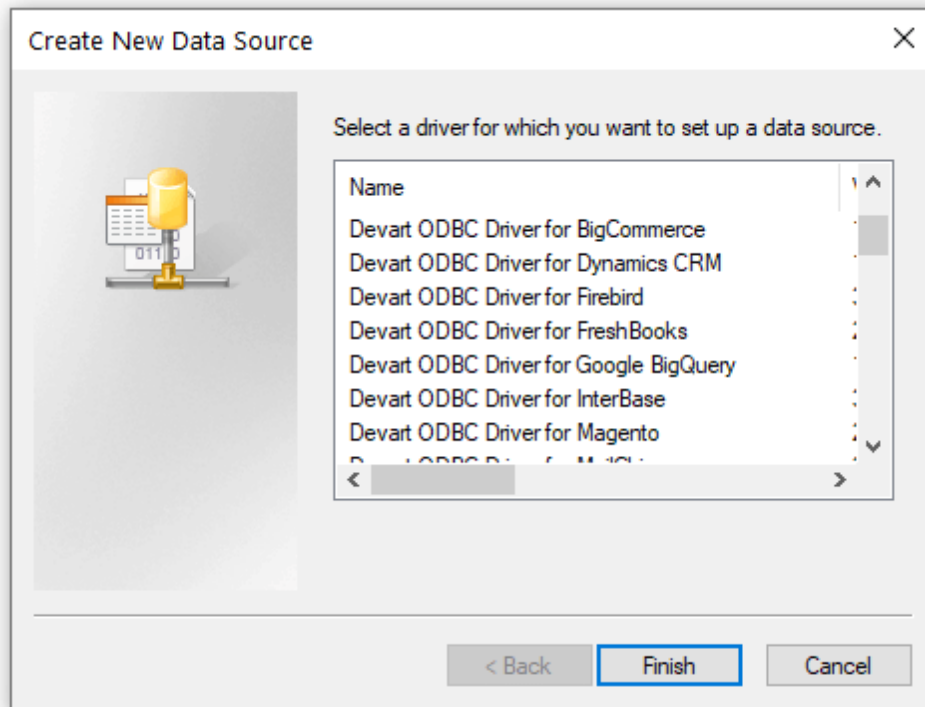To see the license information of your installed driver, do the following:

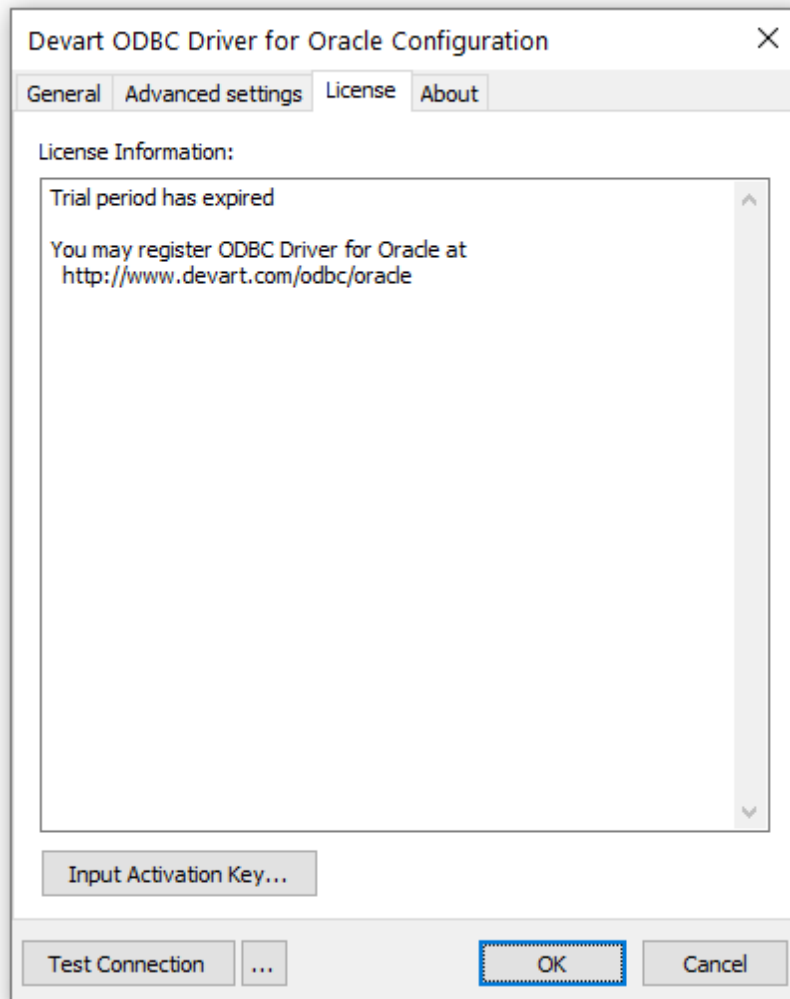1. In the Control Panel run ODBC Administrator

Data Sources
(ODBC)

2. Open the System DSN tab and click the Add button

3. Select the driver and click Finish

4. In the appeared dialogue, select the License tab

See also
- Product Activation

Request Support   ODBC Forum   Provide Feedback

3.3   Connecting to Oracle

Windows DSN Configuration

After installing the driver, create a DSN for Oracle in the ODBC Data Source Administrator.

1. Open the ODBC Data Source Administrator.

- ○ Type `ODBC Data Sources` in the Windows 10 search box and choose the ODBC Data Sources application that matches the bitness of your application (32-bit or 64-bit). You can also open ODBC Data Sources from `Control Panel` > `Administrative Tools`. Note that before Windows 8, the icon was named Data Sources (ODBC).

- ○ Alternatively, you can run `C:\Windows\SysWOW64\odbcad32.exe` to create a 32-bit DSN or `C:\Windows\System32\odbcad32.exe` to create a 64-bit DSN.

2. Select the `User DSN` or `System DSN`. Most applications work with any of them, yet some applications require a specific type of DSN.

3. Click `Add`. The Create New Data Source dialog appears.

4. Select Devart ODBC Driver for Oracle and click `Finish`. The driver setup dialog opens.

5. Enter the connection information in the appropriate fields. You can connect to Oracle directly or through the Oracle Client.
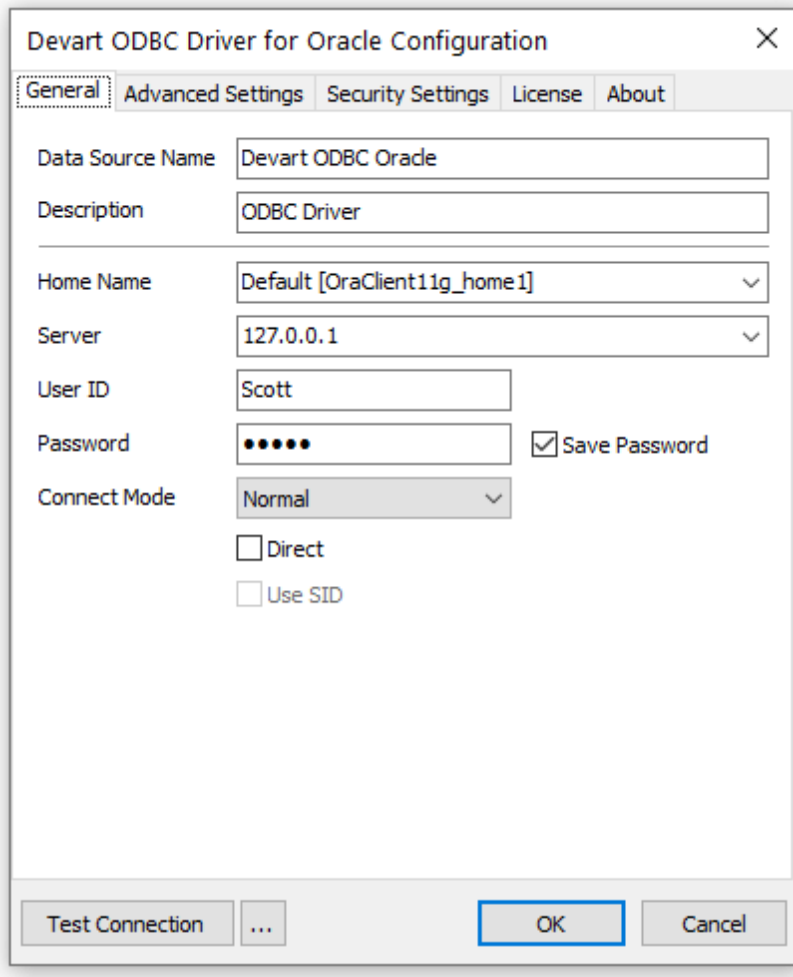
Direct Mode

OCI Mode

6. To test the connectivity, click `Test Connection`.

7. Click `OK` to save the DSN.

See Also

[Connection Options](#)

[Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

3.4   Connection String Options

Oracle ODBC Connection String Options

The following table lists the connection string options for Oracle.

Option	Description
`Direct`	Enables the Direct mode.
`Home`	Used to select the Oracle client to use with the application.
`Host`	Used to specify the host of the data source.
`Password`	Used to specify the password.
`Port`	Used to specify the port number for the connection. 1521 by default.
`Schema`	Used to change the current schema of the session to the specified schema.
`Server`	Used to specify the server name.
`Service Name`	Used to specify the name of the Oracle Client. The value may be selected from the drop-down list of available clients.
`SID`	Used to specify the Oracle system identifier of the database instance.
`User ID`	Used to specify the username.

Advanced Settings

Option	Description
`AllSchemas`	Enables the use of all schemas that an Oracle user has been granted privileges on. The default value is False.
`Allow NULL strings`	To retrieve metadata, not all parameters according to MSDN can accept a null value. If NULL, the driver should return an error. But some 3rd-party tools pass NULL to the parameters. These options should be enabled for compatibility with such tools.
`Empty strings as NULL`	
`Character Set`	Used to set the character set that the driver uses to read and write character data.
`ConnectMode`	Used to specify the system privilege for the user who connects to the server. The privilege must be granted to the user before connecting to the server. `Normal` The default value. Connect as a normal user. `SysOper`

	Connect with the `SYSOPER` privilege.
	
	`SysDBA`
	
	Connect with the `SYSDBA` privilege.
	
	`SysASM`
	
	Connect with the `SYSASM` privilege.
	
	`SysBkp`
	
	Connect with the `SysBackup` privilege.
	
	`SysDG`
	
	Connect with the `SysDG` privilege.
	
	`SysKM`
	
	Connect with the `SYSKM` privilege.

`Connection Timeout`	The time (in seconds) to wait for a connection to open before terminating an attempt. The default value is 0 — an attempt to connect waits indefinitely.
`Date Type`	• Default - The default value. The driver maps the Oracle DATE data type to the ODBC data type SQL_TYPE_TIMESTAMP. • Date Only - The driver maps the Oracle DATE data type to the ODBC data type SQL_TYPE_DATE.

`IP Version`	The Internet Protocol Version.
	`ivIPv4`
	
	The default value. Internet Protocol Version 4 (IPv4) is used.
	
	`ivIPv6`
	
	Internet Protocol Version 6 (IPv6) is used.
	
	`ivIPBoth`
	
	Either Internet Protocol Version 6 (IPv6) or Version 4 (IPv4) is used.
	
	Note: When the property is set to `ivIPBoth`, a connection attempt is made via `IPv6` if it is enabled in the operating system. If the connection attempt fails, a new connection attempt is made via `IPv4`.

`MinFetchRows`	Respectively, if the SQL_ATTR_ROW_ARRAY_SIZE attribute value is

	less than MinFetchRows, then the number of rows queried from the server will equal MinFetchRows. Otherwise - it will be equal to SQL_ATTR_ROW_ARRAY_SIZE. In order to make the driver to query from the server exactly the same number of rows as specified in the SQL_ATTR_ROW_ARRAY_SIZE attribute, it is enough to set the MinFetchRows parameter to 0. However, in this case, you should take into account, that the SQL_ATTR_ROW_ARRAY_SIZE value is 1 by default, and therefore performance decreases dramatically at this value.
`ODBC Behavior`	Used to set the behavior corresponding to the ODBC specification version that a third-party tool expects. The behavior of ODBC driver can be changed by setting a value for the SQL_ATTR_ODBC_VERSION attribute by calling the SQLSetEnvAttr function. But some third-party tools expect the driver to exhibit ODBC 2.x behavior, but forget to call SQLSetEnvAttr with the specified version or pass an incorrect value there. In this case, the required behavior can be explicitly specified in the Connection String by setting the ODBC Behavior parameter. The possible values are: • Default - default ODBC behavior determined by a third-party tool. • Ver 2.x - ODBC 2.x behavior is explicitly set. • Ver 3.x - ODBC 3.x behavior is explicitly set.
`RegionalNumberSettings`	Enables the use of local regional settings when converting numbers to strings.
`RegionalDateTimeSettings`	Enables the use of local regional settings when converting dates and times to a strings.
`String Types`	Sets the string value types returned by the driver as Default, Ansi or Unicode.

	• Default - the driver defines the string types. • Ansi - all string types will be returned as SQL_CHAR, SQL_VARCHAR and SQL_LONGVARCHAR. • Unicode - all string types will be returned as SQL_WCHAR, SQL_WVARCHAR and SQL_WLONGVARCHAR. The option value should be changed if any third-party tool supports only Ansi string types or Unicode ones.
`Use Unicode`	Enables Unicode support.

Sample Oracle ODBC Connection String in Direct Mode

Using Service Name:

```
DRIVER={Devart ODBC Driver for
Oracle};Direct=True;Host=myServer;Service Name=myServiceName;User
ID=myUsername;Password=myPassword
```

Using SID (deprecated format):

```
DRIVER={Devart ODBC Driver for
Oracle};Direct=True;Host=myServer;SID=mySID;User
ID=myUsername;Password=myPassword
```

Sample Oracle ODBC Connection String in OCI Mode

```
DRIVER={Devart ODBC Driver for Oracle};Data Source=myServer/
myServiceName;Home Name=myHomeName;User
ID=myUsername;Password=myPassword
```

See also:

- SSL Connection
- SSH Connection
- HTTP Tunneling

Request Support   ODBC Forum   Provide Feedback

3.5 Secure Connections

This section describes how to establish secure connections to Oracle with ODBC Driver for Oracle.

- SSL Connection
- SSH Connection
- HTTP Tunneling

Request Support   ODBC Forum   Provide Feedback

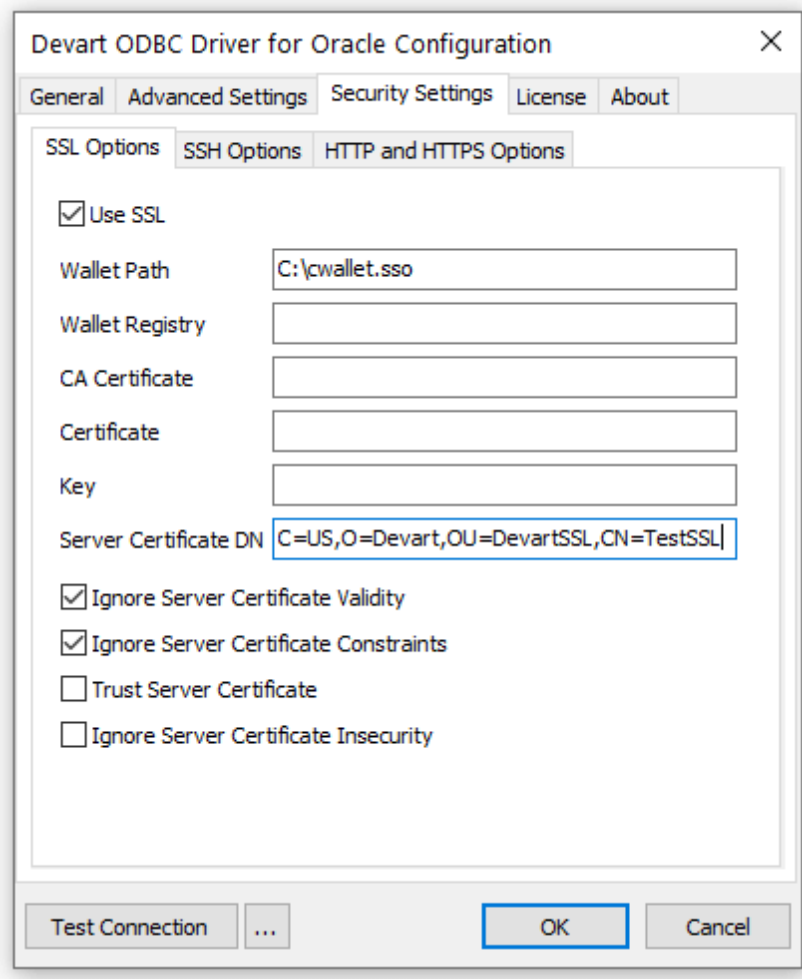### 3.5.1 SSL Connection

Connecting to Oracle Using SSL

SSL (Secure Sockets Layer) is a standard protocol for secure access to a remote machine over untrusted networks. It runs on top of TCP/IP to secure client-server communications by allowing an SSL-enabled client to authenticate itself to an SSL-enabled server and vice versa. During server authentication, an SSL-enabled client application uses standard techniques of public-key cryptography to verify the server's identity by checking that the server's certificate is issued by a trusted certificate authority (CA) and proves the ownership of the public key.

Conversely, SSL client authentication allows the server to validate the client's identity. The client and server can also authenticate each other using self-signed certificates, however, you will almost never want to use a self-signed certificate, except for an Intranet or a development server. After establishing an SSL connection, the client and server can exchange messages that are symmetrically encrypted with the shared secret key. SSL is the recommended method to establish a secure connection to Oracle due to easier configuration and higher performance, compared to SSH.

Connecting to Oracle Using Wallet Path

A wallet is container for storing authentication and signing credentials, including keys and certificates needed by SSL. To establish an SSL connection to Oracle Database using a wallet file, specify the path to the wallet in the `Wallet` Path field. See the Oracle documentation for information on creating an Oracle wallet. If you are using Oracle Cloud, see this document
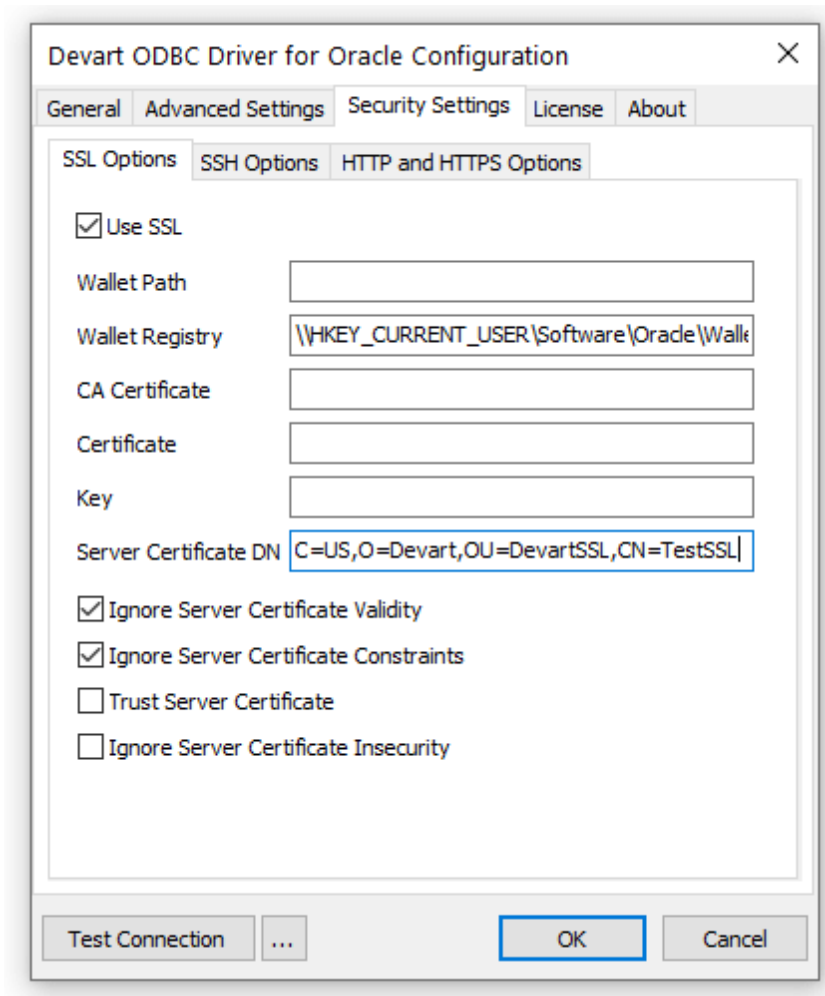
for information on obtaining a wallet.

## Connecting to Oracle Using Oracle Wallet (Registry)

To establish an SSL connection to Oracle Database using a wallet that is stored in the Windows registry, specify the registry key in the `Wallet Path` field. See this [document](#) for information on storing a wallet in the registry.

Connecting to Oracle Using SSL Certificates and Keys

To establish an SSL connection to Oracle Database using certificates and keys, specify the CA certificate, client certificate, and private client key in the `CA Certificate`, `Certificate`, and `Key` fields, respectively.
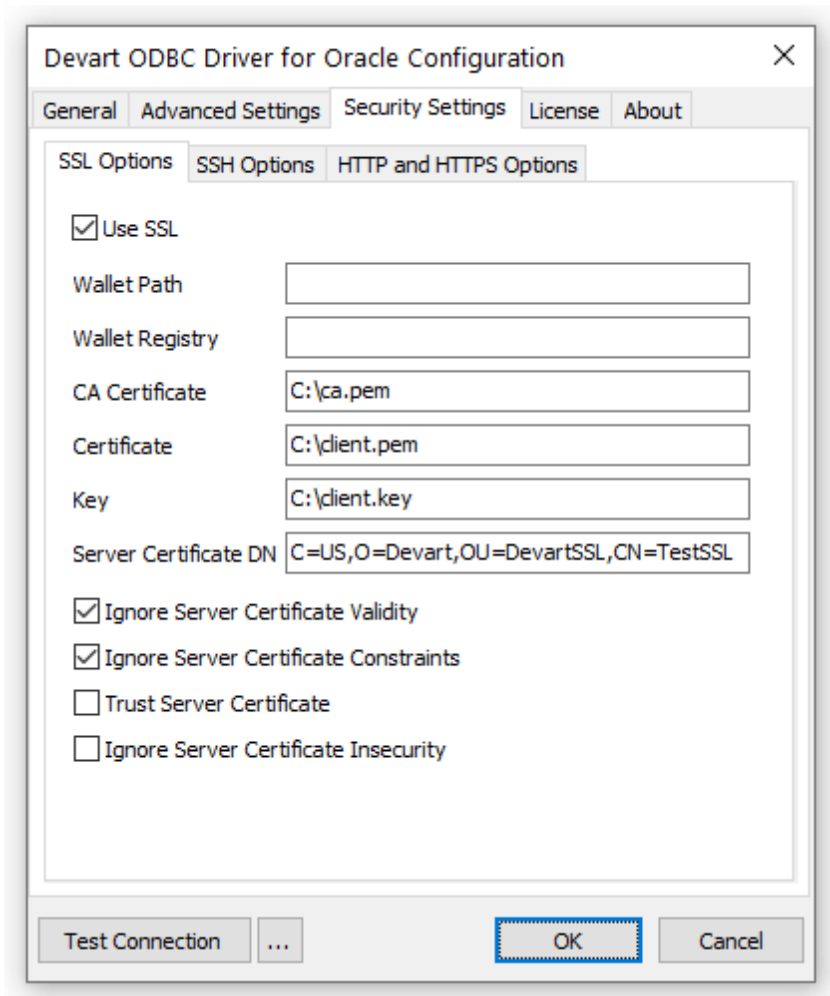
SSL Options

Option	Description
Use SSL	Enables SSL connections.
Wallet Path	The directory where the wallet is stored.
Wallet Registry	The registry key where the wallet is stored.
CA Certificate	The CA certificate.
Certificate	The client certificate.
Key	The private client key.
Server Certificate DN	The server's distinguished name (DN) to

	enable server DN matching. It checks whether the server is genuine by matching the server's global database name against the DN from the server certificate.
`Ignore Server Certificate Validity`	Specifies whether to check the server certificate validity period during an SSL handshake. The default value is `True`.
`Ignore Server Certificate Constraints`	Specifies whether to verify the server certificate for compliance with constraints during an SSL handshake. The default value is `True`.
`Trust Server Certificate`	Specifies whether to verify the server certificate during an SSL handshake. By default, the driver verifies the server certificate. When the option is set to `True`, the driver will bypass walking the certificate chain to verify the certificate.
`Ignore Server Certificate Insecurity`	Specifes whether to check the server certificate signature security during an SSL handshake. The default value is `False`.

Sample Connection String Using Oracle Wallet (Path)

```
DRIVER={Devart ODBC Driver for Oracle};Direct=True;Host=tcps://
myHost;Service Name=myServiceName;User
ID=myUsername;Password=myPassword;Use SSL=True;Wallet
Path=myWallet.sso
```

Sample SSL Connection String Using Oracle Wallet (Registry)

```
DRIVER={Devart ODBC Driver for Oracle};Direct=True;Host=tcps://
```

```
myHost;Service Name=myServiceName;User
ID=myUsername;Password=myPassword;Use SSL=True;Wallet Registry=\
\HKEY_CURRENT_USER\Software\Oracle\Wallets\Key1\myWallet.sso
```

Sample Connection String Using SSL Certificates and Keys

```
DRIVER={Devart ODBC Driver for Oracle};Direct=True;Host=tcps://
myHost;Service Name=myServiceName;User
ID=myUsername;Password=myPassword;Use SSL=True;SSL CA Cert=C:
\myCaCertificate.pem;SSL Cert=C:\myClientCertificate.pem;SSL
Key=C:\myPrivateClient\myKey.pem
```

Request Support   ODBC Forum   Provide Feedback

3.5.2 SSH Connection

Connecting to Oracle Using SSH

This section discusses how to connect to Oracle through SSH. Secure Shell (SSH) is cryptographic network protocol for secure remote login, command execution and file transfer over untrusted networks. SSH uses client-server architecture, connecting an SSH client with an SSH server. The client and server authenticate each other and pass commands and output back and forth. To secure the transmitted data, SSH employs forms of symmetric encryption, assymetric encryption, and hashing.

In symmetric key cryptography, a single key is used by the sending and receiving parties to encrypt and decrypt messages. Assymetric encryption requires two associated keys, the private key and the public key. The public key encrypts messages that can only be decrypted by the private key. The public can key can be freely shared with anyone to autenticate another party, while the private key must be kept secret. The client public key must be stored in a location that is accessible by the SSH server to authenticate the server by the client; conversely, the server public key must be placed on the client side to authenticate the client by the server. Assymetrical encryption is used during the initial key exchange process to produce the shared secret (session key) to encrypt messages for the duration of the session.

The SSH server listens on default port 22 (this port can be changed) for incoming TCP connections. The SSH client begins the initial TCP handshake with the server and verifies the

server's identity. The client and server agree upon the encryption protocol and negotiate a session key. The server then authenticates the client and spawns the right environment. The ODBC driver for Oracle implements the SSH client feature to connect to the SSH server on the remote machine at the specified port. The SSH server authenticates the client and enables the driver to establish a secure direct connection to Oracle. Below is a simplified diagram representing the SSH tunneling.


Note: You don't have to install the SSH client since ODBC Driver for Oracle implements the SSH client functionality.

If you're connecting to Oracle Cloud, you can use the default username `ops` and leave the password and server key empty.

SSH Connection Options

To establish an SSH connection to Oracle, specify the connection parameters on the `SSH Options` tab under `Security Settings`.
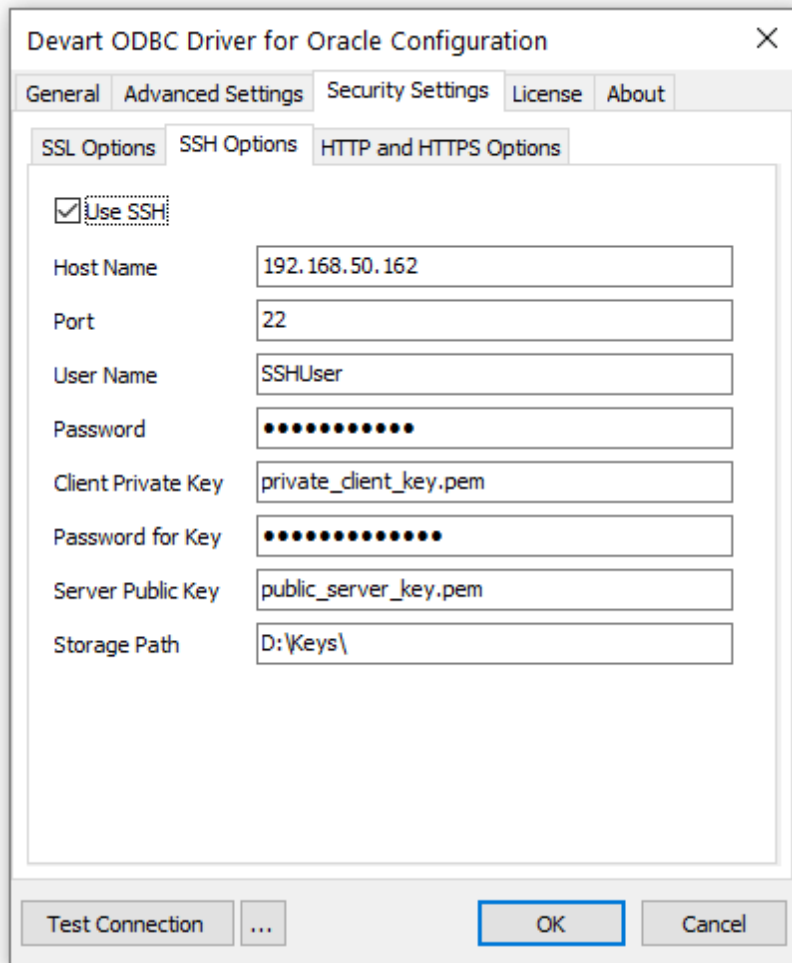
SSH Connection Options:

Option	Description
Use SSH	Enables SSH connections.
SSH Host name	The host name or IP address of the SSH server.
SSH Port	The SSH port number (22 by default).
SSH User Name	The username for the account on the SSH server. For Oracle Cloud, you can use the default user, ops.

SSH Password	The password for the account on the SSH server. If you use the default user, *ops* in Oracle Cloud, leave the option empty.
SSH Client Key	The filename of the client private key for key-based authentication.
SSH Client Key Password	The passphrase for the client private key.
SSH Server Key	The filename of the SSH server public key. For Oracle Cloud, leave the option empty.
SSH Storage Path	The directory where the encryption keys are stored.

Sample Connection String:

```
DRIVER=Devart ODBC Driver for
Oracle;Direct=True;Host=myHost;Service Name=myServiceName;User
ID=myUsername;Password=myPassword;Use SSH=True;SSH Host
name=mySshHost;SSH User Name=mySshUsername;SSH
Password=mySshPassword;SSH Client Key=myPrivateClientKey.pem;SSH
Client Key Password=myClientKeyPassphrase;SSH Server
Key=myPublicServerKey.pem;SSH Storage Path=myDirectoryWithKeys
```

[Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

3.5.3 HTTP Tunneling

Connecting to Oracle Using HTTP Tunneling

This section discusses how to connect the ODBC driver to Oracle through an HTTP tunnel. If you need to connect to Oracle in conditions of restricted connectivity, e.g. when a database server is hidden behind a firewall, or you need to transmit private network data through a public network, you can set up an HTTP tunnel to create a direct network link between two locations. The tunnel is created by an intermediary called a proxy server.

When Oracle server is hidden behind a firewall, the client is not able to connect to the server

directly on a specified port. If the firewall allows HTTP connections, you can use the ODBC driver with a properly configured web server to connect to the database server. The driver supports HTTP tunneling based on the PHP script.

A possible scenario of using HTTP tunneling: the client needs to access the database of a website from a remote machine, but access to the designated port of the database server is forbidden — only connections on the HTTP port 80 are allowed. To establish a connection in this situation, you must deploy the `tunnel.php` script, which is distributed with the driver, on the web server. It enables access to the database server through an HTTP tunnel. The script must be accessible through HTTP. You can verify the script accessibility using any web browser. The script file is located in the "C:\Program Files (x86)\Devart\ODBC\Oracle\http\tunnel.php" folder. The web server must support PHP 5 or later.

To set up an HTTP tunnel, specify the connection parameters on the `HTTP and HTTPS Options` tab under `Security Settings`.
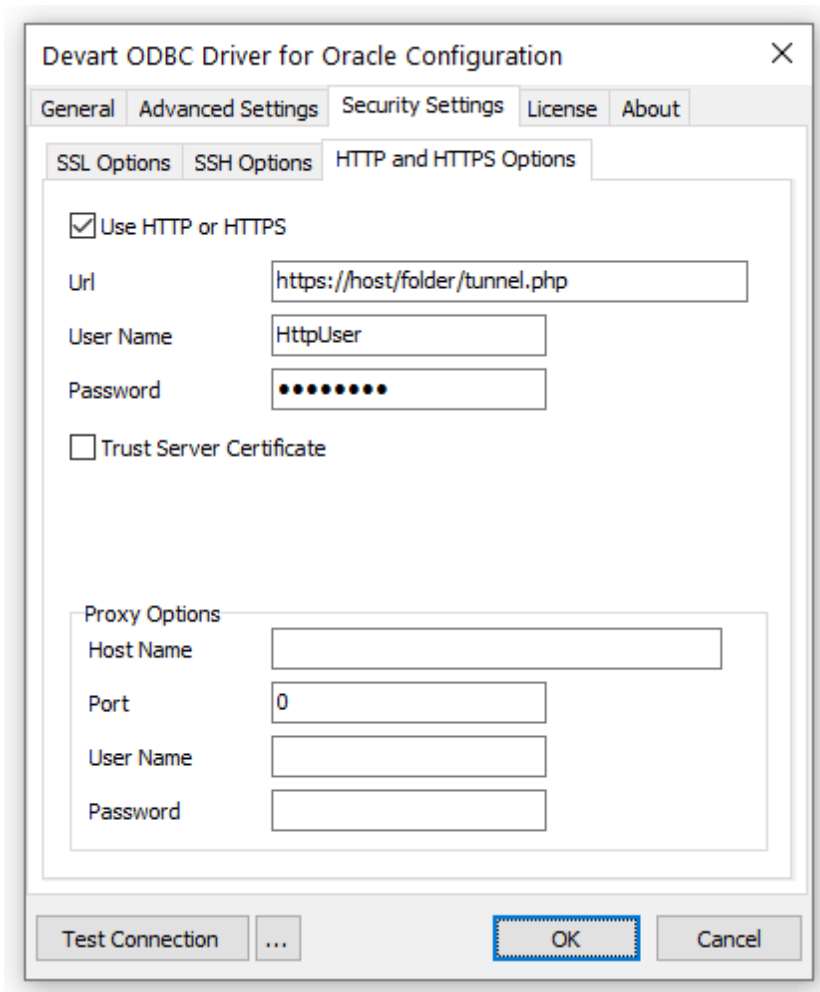
HTTP Tunneling Options

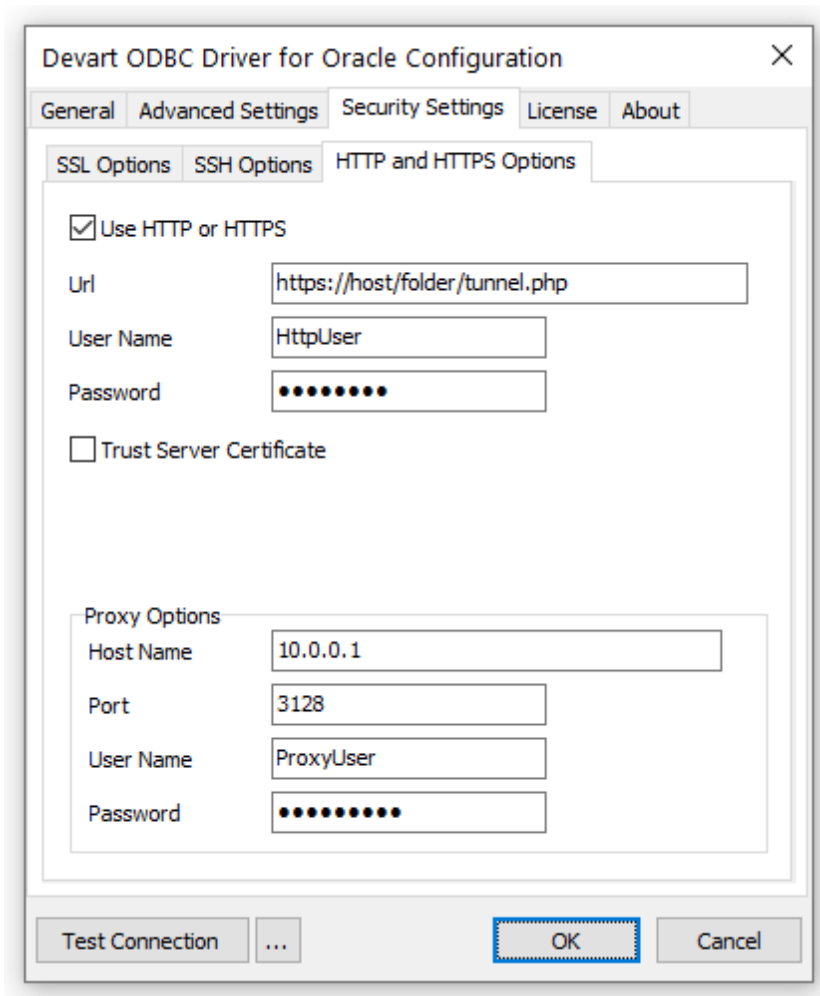| Option	Description
Use Http	Enables HTTP tunneling.
Http Url	The URL of the PHP script for HTTP tunneling.
Http User Name	The username for the password-protected directory that contains the HTTP tunneling script.
Http Password	The password for the password-protected directory that contains the HTTP tunneling script.
Http Trust Server	Specifies whether to verify the server certificate during an SSL

`Certificate`	handshake. When `True`, the driver bypasses walking the certificate chain to verify the certificate. The default value is `False`.

Sample Connection String Using HTTP Tunneling

```
DRIVER=Devart ODBC Driver for
Oracle;Direct=True;Host=myHost;Service Name=myServiceName;User
ID=myUsername;Password=myPassword;Use Http=True;Url=https://host/
folder/tunnel.php;Http User Name=myHttpUsername;Http
Password=myHttpPassword
```

Connecting Through HTTP Tunnel and Proxy Server

The HTTP tunneling server may be not be directly accessible from the client machine. In this case, you need to additionally provide connection information for the proxy server.

Proxy Options

Option	Description
Proxy Host Name	The proxy hostname or IP address.
Proxy Port	The proxy port.
Proxy Username	The proxy username.
Proxy Password	The proxy password.

Sample Connection String Using HTTP Tunneling and Proxy Server

```
DRIVER=Devart ODBC Driver for
Oracle;Direct=True;Host=myHost;Service Name=myServiceName;User
```

```
ID=myUsername;Password=myPassword;Use Http=True;Url=https://host/
folder/tunnel.php;Http User Name=myHttpUsername;Http
Password=myHttpPassword;Proxy Host Name=myProxyHost;Proxy
Port=myProxyPort;Proxy User Name=myProxyUsername;Proxy
Password=myProxyPassword
```

Additional Information

There is one more way to tunnel network traffic. The Secure Shell forwarding, or SSH, can be used for data forwarding. However, SSH is designed to encrypt traffic rather than traverse firewalls. The Connecting via SSH document describes how to set up an SSH connection in the ODBC Driver for Oracle.

Note that traffic tunneling or encryption increases the CPU and bandwidth usage. It is recommended that you use direct connection whenever possible.

   Request Support   ODBC Forum   Provide Feedback

3.6   **Sandboxed Apps on macOS**

Sandboxed Apps on macOS

Sandboxed applications don't have permission to access iODBC Driver Manager on macOS. This is caused by the System Integrity Protection (SIP) technology on macOS which protects your files and folders from potentially malicious software by locking the application. When accessing a data source from an application like Excel through the ODBC driver for Oracle, you may get an error message saying that the driver is unable to create a file.

Note that all third-party applications distributed through the Mac App Store are sandboxed.

Disabling System Integration Protection (SIP) on macOS

To resolve the issue, you should turn off SIP on your computer:

1. Restart your computer in **Recovery mode** (hold down **Command + R** until you see the Apple logo).

2. Select **Utilities** > **Terminal**.

3. In the Terminal window, enter `csrutil disable`.

```
● ● ●              Terminal — -bash — 80×24
[-bash-3.2# csrutil disable                                      ]
Successfully disabled System Integrity Protection. Please restart the machine fo
r the changes to take effect.
-bash-3.2# reboot█
```

4. Restart your computer.

5. Enter `csrutil status` to check the status of SIP.

```
● ● ●         ⌂ test — paserver — -bash — 80×24
tests-imac-2:~ test$ csrutil status                             ]
System Integrity Protection status: disabled.
tests-imac-2:~ test$ █
```

Enable SIP after you finish working with an ODBC data source. To turn on SIP, enter `csrutil enable` and restart your computer.

Request Support   ODBC Forum   Provide Feedback

3.7   Using with iODBC

Using the Driver with iODBC

Among known issues with iODBC driver manager is incorrect handling of the following ODBC data types:

- SQL_WCHAR
- SQL_WVARCHAR
- SQL_WLONGVARCHAR

For this reason, we recommend using the following data types instead:

- SQL_CHAR
- SQL_VARCHAR
- SQL_LONGVARCHAR

If you have to work with the SQL_WCHAR, SQL_WVARCHAR, and SQL_WLONGVARCHAR data types, we recommend that you use the unixODBC driver manager rather than iODBC.

Request Support   ODBC Forum   Provide Feedback

3.8   Enabling ODBC Tracing

Creating an ODBC Trace Log on Windows

When you start or stop tracing in the 64-bit ODBC Administrator, the tracing is also enabled or disabled in the 32-bit ODBC Administrator, and vice versa.

If the ODBC client application you need to trace runs under Local System account or any other user login than your own, select `Machine-Wide tracing for all user identities`. For example, this option may be necessary for SSMS.

To generate a trace file using ODBC Source Administrator on Windows, follow the steps below.

1. Type `ODBC Data Sources` in the Windows 10 search box (in earlier versions of Windows, open `Control Panel` > `Administrative Tools`) and choose the application of the needed bitness.
2. Select the `Tracing` tab.
3. If necessary, change the default `Log File Path`. Make sure that the path is writable by the application, then click `Apply`.
4. Click `Start Tracing Now`.
5. Restart all application processes.
6. Click `Test Connection` in the DSN settings to make sure the driver is able to connect.
7. Reproduce the issue.
8. Click `Stop Tracing Now` on the `Tracing` tab.
9. Send us the obtained log file (for example, devart.log).

Creating an ODBC Trace Log on macOS

To enable the trace option on macOS, use the Tracing tab within ODBC Administrator.

1. Open the ODBC Administrator.
2. Select the `Tracing` tab.
3. If necessary, change the default `Log file path`.
4. Select `All the time` in the `When to trace` option.

Creating an ODBC Trace Log on Linux

To trace the ODBC calls on Linux, set the `Trace` and `TraceFile` keyword/value pairs in the `[ODBC]` section of the `/etc/odbcinst.ini` file, for example:

```
[ODBC]
Trace=Yes
TraceFile=/home/test/devart.log
```

Make sure to disable logging after obtaining a log file since it affects the read/write speed.

Request Support   ODBC Forum   Provide Feedback

3.9   Supported Data Types

Data Type Mapping

The Devart ODBC Driver for Oracle supports all Oracle data types.

The following table describes how the Oracle data types are mapped to the ODBC data types.

Oracle Data Types	ODBC Data Types
CHAR	SQL_CHAR
VARCHAR2	SQL_VARCHAR
NCHAR	SQL_WCHAR
NVARCHAR2	SQL_WVARCHAR
NUMBER	SQL_NUMERIC
BINARY_FLOAT	SQL_FLOAT
BINARY_DOUBLE	SQL_REAL
DOUBLE PRECISION	SQL_DOUBLE
DATE TIMESTAMP TIMESTAMP WITH TIME ZONE TIMESTAMP WITH LOCAL TIME ZONE	SQL_TYPE_TIMESTAMP

BLOB	SQL_LONGVARBINARY
CLOB	SQL_LONGVARCHAR
NCLOB	SQL_WLONGVARCHAR
BFILE	SQL_LONGVARBINARY
LONG	SQL_LONGVARCHAR
RAW	SQL_VARBINARY
LONG RAW	SQL_LONGVARBINARY
ROWID UROWID	SQL_VARCHAR
INTERVAL YEAR TO MONTH	SQL_INTERVAL_YEAR_TO_MONTH
INTERVAL DAY TO SECOND	SQL_INTERVAL_DAY_TO_SECOND
XML	SQL_LONGVARCHAR
else	SQL_UNKNOWN_TYPE

[Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

3.10 Supported ODBC API Functions

Supported ODBC Functions

The SQLGetInfo function returns information about the driver and data source. To find out whether a specific function is supported in the driver, call SQLGetFunctions.

For more information about the ODBC interface, see the [ODBC Programmer's Reference](#).

ODBC Driver for Oracle supports all deprecated functions for backward compatibility.

The following table lists the currently supported ODBC functions.

Function Name	Support	Standard	Purpose
SQLAllocHandle	✔	ISO 92	Obtains an environment,

			connection, statement, or descriptor handle.
SQLConnect	✓	ISO 92	Connects to a specific driver by data source name, user ID, and password.
SQLDriverConnect	✓	ODBC	Connects to a specific driver by connection string or requests that the Driver Manager and driver display connection dialog boxes for the user.
SQLAllocEnv	✓	Deprecated	Obtains an environment handle allocated from driver.
SQLAllocConnect	✓	Deprecated	Obtains a connection handle

ODBC API Calls for Obtaining Information about a Driver and Data Source

Function Name	Support	Standard	Purpose
SQLDataSources	✓	ISO 92	Returns the list of available data sources, handled by the Driver Manager

SQLDrivers	✓	ODBC	Returns the list of installed drivers and their attributes, handles by Driver Manager
SQLGetInfo	✓	ISO 92	Returns information about a specific driver and data source.
SQLGetFunctions	✓	ISO 92	Returns the functions supported by the driver.
SQLGetTypeInfo	✓	ISO 92	Returns information about supported data types.

ODBC API Calls for Setting and Retrieving Driver Attributes

Function Name	Support	Standard	Purpose
SQLSetConnectAttr	✓	ISO 92	Sets a connection attribute.
SQLGetConnectAttr	✓	ISO 92	Returns the value of a connection attribute.
SQLSetConnectOption	✓	Deprecated	Sets a connection option
SQLGetConnectOption	✓	Deprecated	Returns the value of a connection option
SQLSetEnvAttr	✓	ISO 92	Sets an environment

			attribute.
SQLGetEnvAttr	✓	ISO 92	Returns the value of an environment attribute.
SQLSetStmtAttr	✓	ISO 92	Sets a statement attribute.
SQLGetStmtAttr	✓	ISO 92	Returns the value of a statement attribute.
SQLSetStmtOption	✓	Deprecated	Sets a statement option
SQLGetStmtOption	✓	Deprecated	Returns the value of a statement option

ODBC API Calls for Preparing SQL Requests

Function Name	Support	Standard	Purpose
SQLAllocStmt	✓	Deprecated	Allocates a statement handle
SQLPrepare	✓	ISO 92	Prepares an SQL statement for later execution.
SQLBindParameter	✓	ODBC	Assigns storage for a parameter in an SQL statement.
SQLGetCursorName	✓	ISO 92	Returns the cursor name associated with a statement handle.
SQLSetCursorName	✓	ISO 92	Specifies a cursor name.

SQLSetScrollOptions	✅	ODBC	Sets options that control cursor behavior.

ODBC API Calls for Submitting Requests

Function Name	Support	Standard	Purpose
SQLExecute	✅	ISO 92	Executes a prepared statement.
SQLExecDirect	✅	ISO 92	Executes a statement
SQLNativeSql	✅	ODBC	Returns the text of an SQL statement as translated by the driver.
SQLDescribeParam	✅	ODBC	Returns the description for a specific parameter in a statement.
SQLNumParams	✅	ISO 92	Returns the number of parameters in a statement.
SQLParamData	✅	ISO 92	Used in conjunction with SQLPutData to supply parameter data at execution time. (Useful for long data values.)
SQLPutData	✅	ISO 92	Sends part or all of a data value for a

			parameter. (Useful for long data values.)

ODBC API Calls for Retrieving Results and Information about Results

Function Name	Support	Standard	Purpose
SQLRowCount	✔	ISO 92	Returns the number of rows affected by an insert, update, or delete request.
SQLNumResultCols	✔	ISO 92	Returns the number of columns in the result set.
SQLDescribeCol	✔	ISO 92	Describes a column in the result set.
SQLColAttribute	✔	ISO 92	Describes attributes of a column in the result set.
SQLColAttributes	✔	Deprecated	Describes attributes of a column in the result set.
SQLFetch	✔	ISO 92	Returns multiple result rows.
SQLFetchScroll	✔	ISO 92	Returns scrollable result rows.
SQLExtendedFetch	✔	Deprecated	Returns scrollable result rows.
SQLSetPos	✔	ODBC	Positions a cursor within a fetched

			block of data and enables an application to refresh data in the rowset or to update or delete data in the result set.
SQLBulkOperations	✔	ODBC	Performs bulk insertions and bulk bookmark operations, including update, delete, and fetch by bookmark.

ODBC API Calls for Retrieving Error or Diagnostic Information

Function Name	Support	Standard	Purpose
SQLError	✔	Deprecated	Returns additional error or status information
SQLGetDiagField	✔	ISO 92	Returns additional diagnostic information (a single field of the diagnostic data structure).
SQLGetDiagRec	✔	ISO 92	Returns additional diagnostic information (multiple fields of the

			diagnostic data structure).

ODBC API Calls for Obtaining Information About Database Objects (Catalog Functions)

Function Name	Support	Standard	Purpose
SQLColumnPrivileges	✓	ODBC	Returns a list of columns and associated privileges for one or more tables.
SQLColumns	✓	X/Open	Returns the list of column names in specified tables.
SQLForeignKeys	✓	ODBC	Returns a list of column names that make up foreign keys, if they exist for a specified table.
SQLPrimaryKeys	✓	ODBC	Returns the list of column names that make up the primary key for a table.
SQLProcedureColumns	✓	ODBC	Returns the list of input and output parameters, as well as the columns that constitute the result set for the specified

			procedures.
SQLProcedures	✔	ODBC	Returns the list of procedure names stored in a specific data source.
SQLSpecialColumns	✔	X/Open	Returns information about the optimal set of columns that uniquely identifies a row in a specified table, or the columns that are automatically updated when any value in the row is updated by a transaction.
SQLStatistics	✔	ISO 92	Returns statistics about a single table and the list of indexes associated with the table.
SQLTablePrivileges	✔	ODBC	Returns a list of tables and the privileges associated with each table.
SQLTables	✔	X/Open	Returns the list of table names stored

			in a specific data source.

ODBC API Calls for Performing Transactions

Function Name	Support	Standard	Purpose
SQLTransact	✓	Deprecated	Commits or rolls back a transaction
SQLEndTran	✓	ISO 92	Commits or rolls back a transaction.

ODBC API Calls for Terminating a Statement

Function Name	Support	Standard	Purpose
SQLFreeStmt	✓	ISO 92	Ends statement processing, discards pending results, and, optionally, frees all resources associated with the statement handle.
SQLCloseCursor	✓	ISO 92	Closes a cursor that has been opened on a statement handle.
SQLCancel	✓	ISO 92	Cancels an SQL statement.

ODBC API Calls for Terminating a Connection

Function Name	Support	Standard	Purpose
SQLDisconnect	✓	ISO 92	Closes the connection.

SQLFreeHandle	✓	ISO 92	Releases an environment, connection, statement, or descriptor handle.
SQLFreeConnect	✓	Deprecated	Releases connection handle.
SQLFreeEnv	✓	Deprecated	Releases an environment handle.

Request Support   ODBC Forum   Provide Feedback

4   Using in Third-Party Tools

This section discusses how to use ODBC Driver for Oracle with ODBC-compliant tools.

- DBeaver
- Denodo
- Oracle Database Link
- Microsoft Access
- Microsoft Excel
- OpenOffice and LibreOffice
- PHP
- Power BI
- Python
- QlikView
- SQL Server Management Studio
- SSIS
- Tableau

Request Support   ODBC Forum   Provide Feedback

4.1 Using in Microsoft Access

Connecting Microsoft Access to Oracle Using an ODBC Driver

This article explains how to connect Microsoft Access to Oracle through the standard ODBC interface. Microsoft Access is a dababase management system that combines the relational database engine with a graphical user interface. Access can be used as a substitution for spreadsheet applications like Excel to organize, store, and retrieve large amounts of related data that can be difficult to manage in spreadsheets.

In Microsoft Access, you can connect to your Oracle data either by importing it or creating a table that links to the data. Devart ODBC drivers support all modern versions of Access. It is assumed that you have already installed and configured a DSN for ODBC driver for Oracle. For the purpose of this article, we tested an ODBC connection to Oracle through our ODBC drivers in Microsoft Access 2003, Microsoft Access 2007, Microsoft Access 2010, Microsoft Access 2013, Microsoft Access 2016, Microsoft Access 2019. The following steps describe how to use Microsoft Access 2019 to import or link to your data in Oracle.

Importing Oracle Data Into Microsoft Access Through an ODBC Connection

1. Open your Microsoft Access database.

2. Select the **External Data** tab in the ribbon.

3. Expand the **New Data Source** drop-down and select **From Other Sources**, then select **ODBC Dababase**.

4. In the **Get External Data - ODBC Database** dialog box, select **Import the source data into a new table in the curent database**, and click **OK**.

5. In the **Select Data Source** dialog box, select the **Machine Data Source** tab.

6. Select the DSN that you have configured for Oracle and click **OK**.

7. In the **Import Objects** dialog box, select the tables that you want to import, and click **OK**.

8. If the database objects have been successfully imported, you should the see the corresponding message in the dialog box. If you want to save the import steps to quickly repeat the process without using the wizard at a later time, select the **Save import steps** checkbox. Click **Close**.

9. The imported tables should appear in the **Tables** navigation pane on the left.

10. Double-click on the needed table to display its contents.

Linking to Oracle Data in Microsoft Access Through an ODBC Connection

1. Open your Microsoft Access database.

2. Select the **External Data** tab in the ribbon.

3. Expand the **New Data Source** drop-down and select **From Other Sources**, then select **ODBC Dababase**.

4. In the **Get External Data - ODBC Database** dialog box, select **Link to the data source by creating a linked table**.

5. In the **Select Data Source** dialog box, select the **Machine Data Source** tab.

6. Select the DSN that you have configured for Oracle and click **OK**.

7. In the **Link Tables** dialog box, select the table or tables that you want to link to, and click **OK**.

8. The **Select Unique Record Identifier** dialog box will prompt you to choose a field or fields that uniquely identify each record in the table. To avoid inconsistencies, it is recommended to select the primary key in the Oracle table as the unique record identifier. You are linking multiple tables, you will be prompted to select unique record identifiers for each of the selected tables.

9. The linked tables should appear in the **Tables** navigation pane on the left.

10. Double-click on the needed table to display its contents.

[Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

4.2 Using in DBeaver

DBeaver Overview

DBeaver is a free, open source multiplatform database management tool and SQL client for developers and database administrators. DBeaver can be used to access any database or cloud application that has an ODBC or JDBC driver, such as Oracle, SQL Server, MySQl, Salesforce, or Mailchimp. Devart DBeaver provides you with the most important features you'd need when working with a database in a GUI tool, such as:

- SQL queries execution

- Metadata browsing and editing

- SQL scripts management

- Data export/import

- Data backup

- DDL generation

- ER diagrams rendering

- Test data generation

- BLOB/CLOB support

- Database objects browsing

- Scrollable resultsets

The tool comes in two editions — Community and Enterprise. Enterprise Edition supports NoSQL databases, such as MongoDB or Cassandra, persistent query manager database, SSH tunneling, vector graphics (SVG) and a few other enterprise-level features. Note though that you can access a MongoDB database from DBeaver Community Edition using the respective Devart ODBC driver. For the purposes of this guide, we'll use the Community Edition of DBeaver to retrieve data from Oracle via the Open Database Connectivity driver.

Creating an ODBC Data Source to Use Oracle Data in DBeaver

1. Click the **Start** menu and select **Control Panel**.

2. Select **Administrative Tools**, then click **ODBC Data Sources**.

3. Click on the **System DSN** tab if you want to set up a DSN name for all users of the system or select **User DSN** to configure DSN only for your account.

4. Click the **Add** button and double-click Devart ODBC Driver for Oracle in the list.

5. Give a name to your data source and set up the connection parameters.

6. Click the **Test Connection** button to verify that you have properly configured the DSN.

When using ODBC driver for Oracle with DBeaver, SQL_WVARCHAR data types may be displayed incorrectly in DBeaver. To prevent this, you need to set the string data types to Ansi either in the **Advanced Settings** tab of the driver configuration dialog or directly in the connection string (String Types=Ansi) — all string types will be returned as SQL_CHAR, SQL_VARCHAR and SQL_LONGVARCHAR.
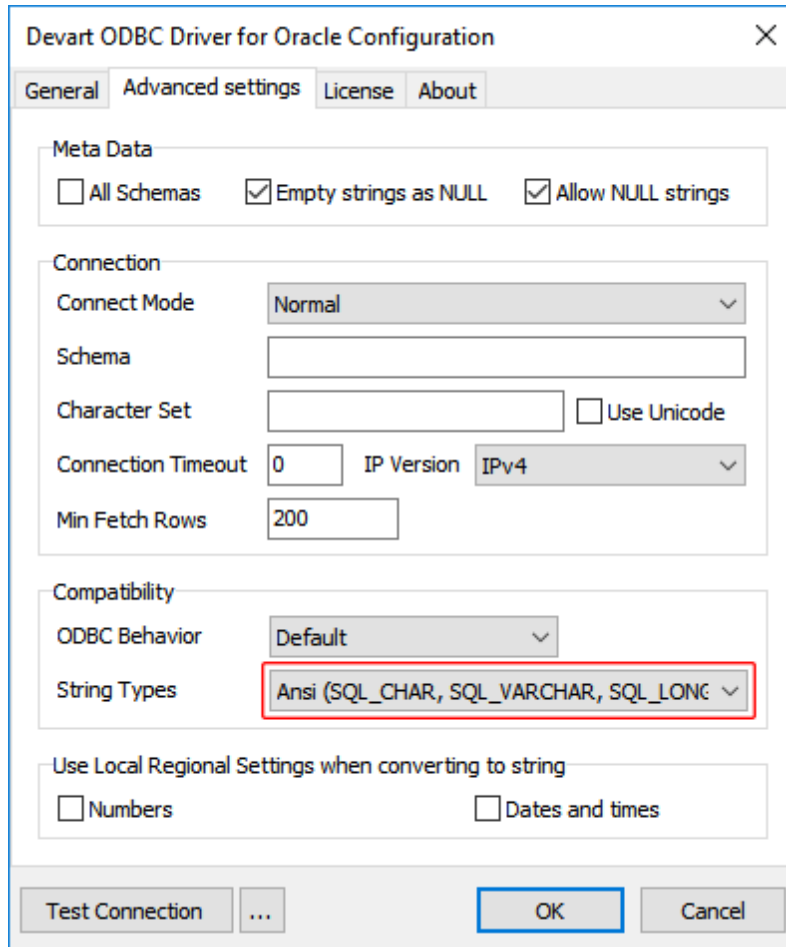
Connecting to Oracle Data from DBeaver via ODBC Driver for Oracle

Follow the steps below to establish a connection to Oracle in DBeaver.

1. In the **Database** menu, select **New Database Connection.**
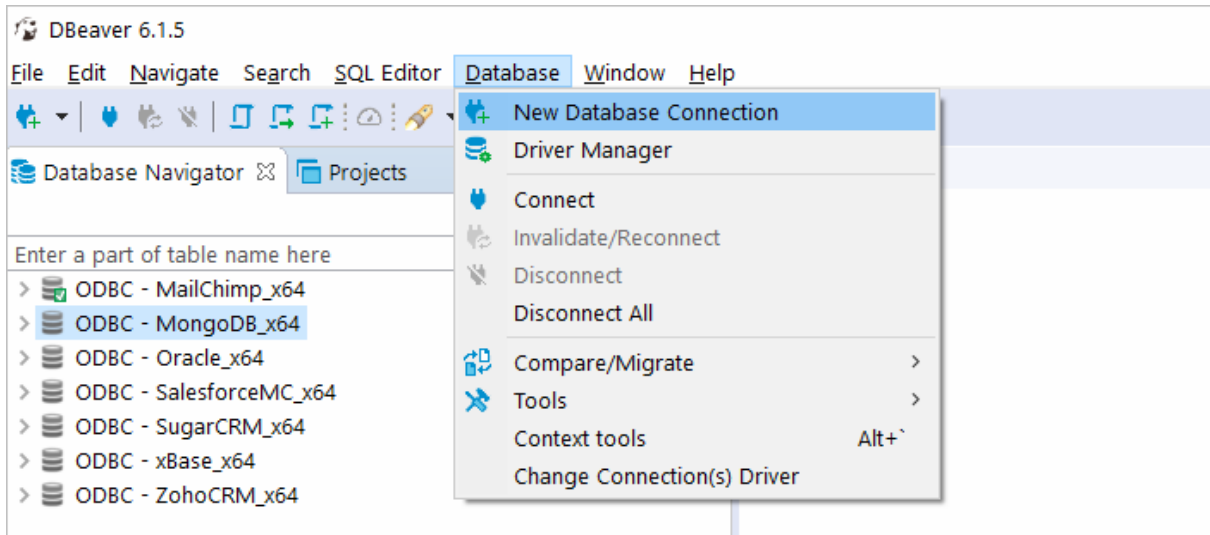
2. In the **Connect to database** wizard, select **ODBC** and click **Next**.
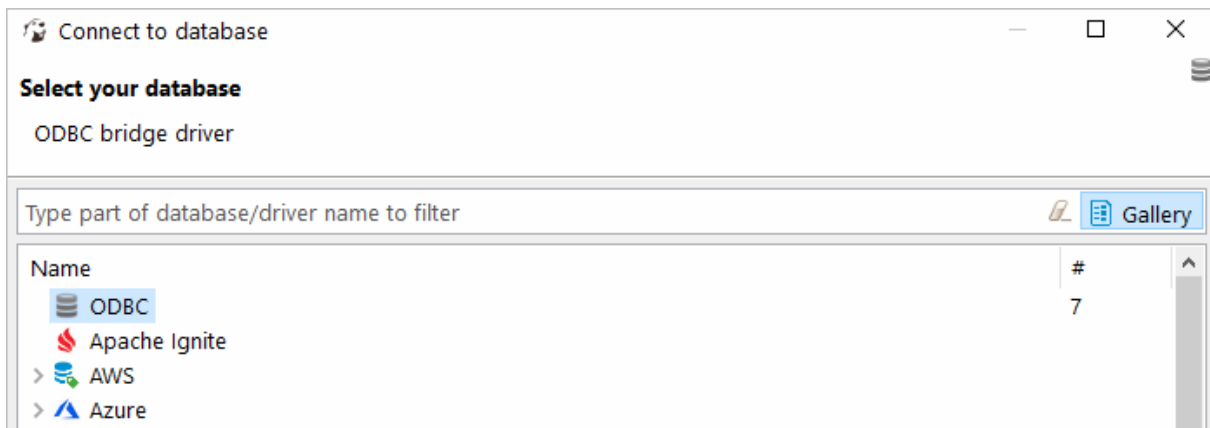
3. Enter the previously configured DSN in the **Database/Schema** field.
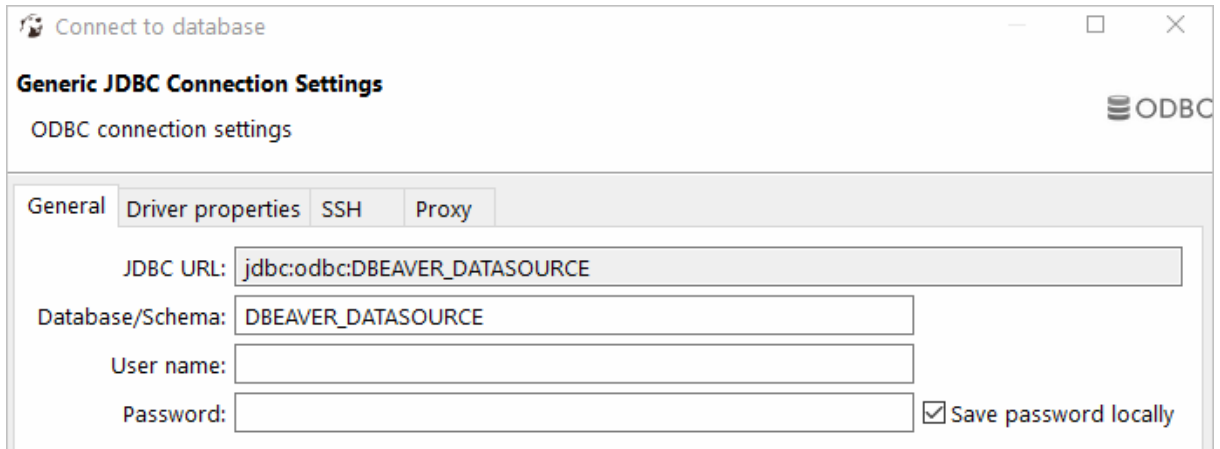
4. Click **Test Connection**. If everything goes well, you'll see the **Success** message.

Viewing Oracle Database Objects and Querying Data

You can expand out the database structure in DBeaver's **Database Navigator** to visualize all the tables in Oracle database. To view and edit the data in a table, you need to right-click on the target table name and select **View data**.The content of the table will be displayed in the main workspace.
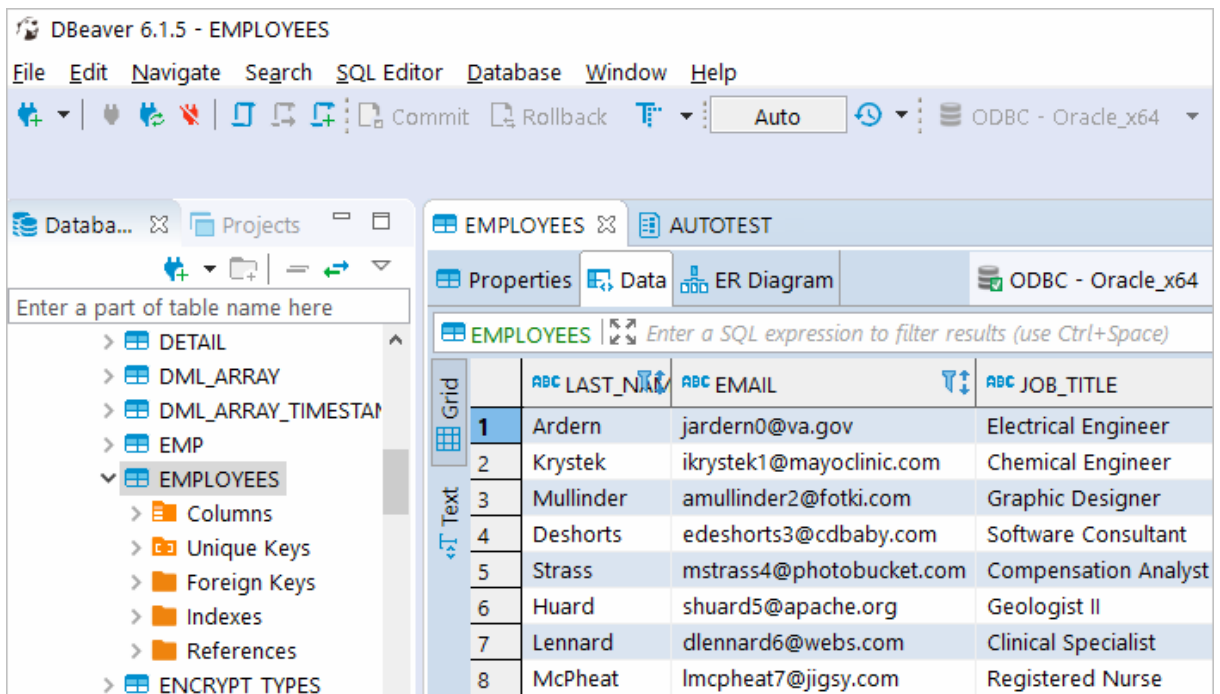
If you want to write a custom SQL query that will include only the necessary columns from the

table, you can select **New SQL Editor** in the **SQL Editor** main menu. Create your query and run it by clicking **Execute SQL Statement** to view the results in the same window.
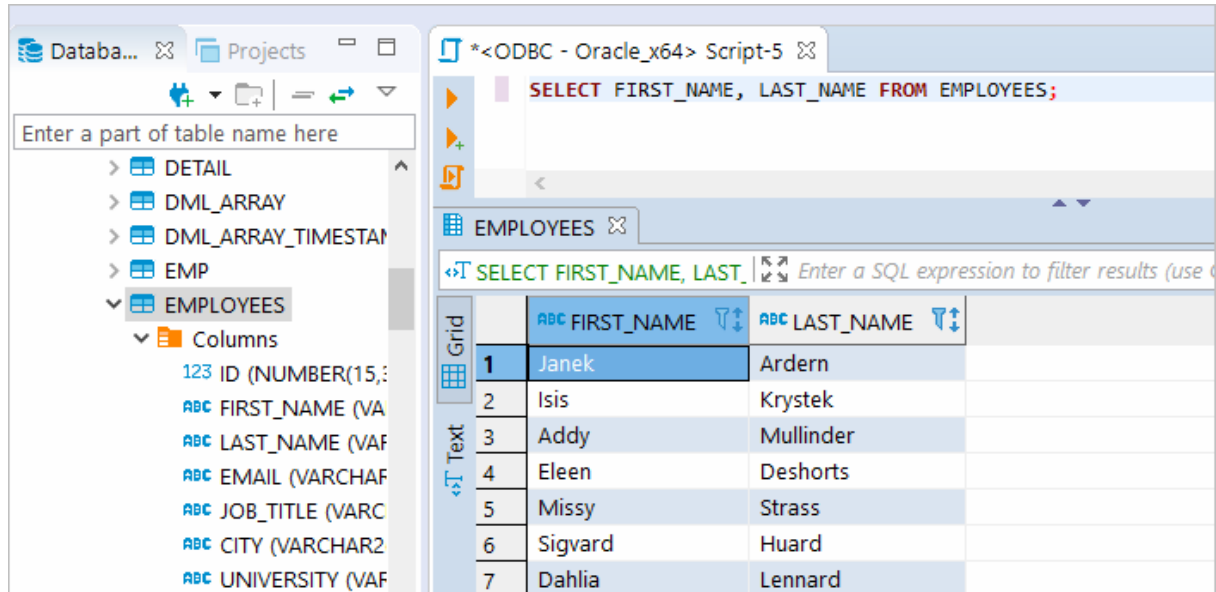
Request Support   ODBC Forum   Provide Feedback

4.3   Using in Denodo

Troubleshooting ODBC Connection in Denodo

When you attempt to access Oracle from Denodo with default settings, you may get an *'Unexpected error creating a connection: Unable to validate object Received exception with message 'Unable to validate object''*. To resolve the error:

1. Select **File** > **New** > **Data Source** > **ODBC** in Virtual DataPort Administration Tool.

2. Click **Connection Pool configuration**.

3. In **Ping query**, change `Select 1` to `Select 1 From DUAL`, then click **Ok**.
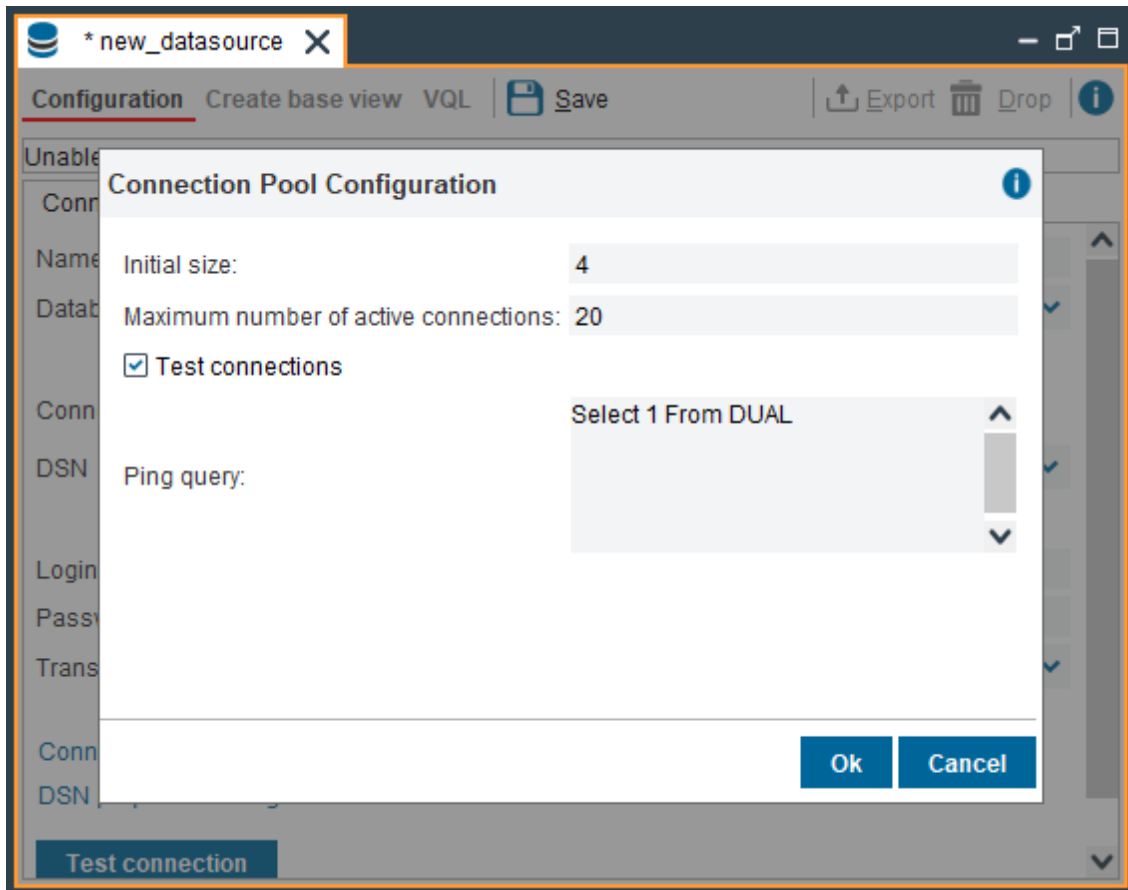
Request Support   ODBC Forum   Provide Feedback

4.4   Using in Oracle DBLink

Configuring Oracle Database Gateway for ODBC

This article explains how to configure Oracle Database Gateway for ODBC. If your data is stored in a non-Oracle database system or cloud application, and you need to access it from an Oracle Database server, you can create a database link to an Oracle Database Gateway for ODBC. The gateway works with an ODBC driver to access non-Oracle systems or other, remote Oracle servers. Any ODBC-compatible data source can be accessed using the gateway and the appropriate ODBC driver. The driver must be installed on the same machine as the gateway. The non-Oracle system can run on the same machine as the Oracle server or on a different machine. The gateway can be installed on the machine running the non-Oracle system, the machine running the Oracle database or on a third machine as a

standalone.

Configure the Initialization File

After installing the gateway and the ODBC driver for Oracle, create an initialization file for your Oracle Database Gateway for ODBC. The sample file `initdg4odbc.ora` is stored in the `ORACLE_HOME\hs\admin` directory. To create an initialization file for the gateway, copy the sample initialization file and rename it. The name must be prefixed with init — for example, `initOracle.ora`. You need a separate initialization file for each ODBC data source. After creating the file, set the `HS_FDS_CONNECT_INFO` parameter to the system DSN that you created earlier, for example:

```
HS_FDS_CONNECT_INFO=Oracle
```

Configure Oracle Net Listener

After configuring the gateway, you need to configure Oracle Net Listener to communicate with the Oracle database. Information about the gateway must be added to the `listener.ora` configuration file which is located in the `ORACLE_HOME\NETWORK\ADMIN\` directory. The following example is the address on which the Oracle Net Listener listens (`HOST` is the address of the machine on which the gateway is installed):

```
LISTENER =
    (DESCRIPTION_LIST =
        (DESCRIPTION =
            (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
        )
    )
```

Add an entry to the `listener.ora` file to start the gateway in response to connection requests. The SID of the gateway (`SID_NAME`) must be the same in `listener.ora` and `tnsnames.ora`. `ORACLE_HOME` is the Oracle home directory where the gateway resides. To apply the new settings, stop and restart the Oracle Net Listener service.

```
SID_LIST_LISTENER=
   (SID_LIST=
      (SID_DESC=
          (SID_NAME=Oracle)
          (ORACLE_HOME=D:\ORACLE_HOME)
          (PROGRAM=dg4odbc)
      )
   )
```

Configure Oracle for Gateway Access

Add a connect descriptor for the gateway to the `tnsnames.ora file,` which is located in `ORACLE_HOME\NETWORK\ADMIN` directory. The `SID` must match the value specified in the `listener.ora` file.

```
Oracle =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = tcp)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SID = Oracle)
    )
    (HS = OK)
  )
```

Create Database Links

To access an ODBC data source, you must create a database link using a database tool like SQL Developer. Connect to your database server and execute the `CREATE DATABASE LINK` statement, as follows:

```
CREATE DATABASE LINK dblink CONNECT TO "username"  IDENTIFIED BY "password"
```

`dblink` is the complete database link name. `tns_name_entry` is the Oracle Net connect descriptor specified in the `tnsnames.ora` file.

After creating the database link, refresh and expand the connection in the left pane of SQL Developer. You should see the newly created link in **Database Links**. You can run a query against the ODBC data source using the syntax:

```
SELECT * FROM table_name@"dblink_name"
```

See also
Configuring Oracle Database Gateway for ODBC

Request Support   ODBC Forum   Provide Feedback

4.5   **Using in Microsoft Excel**

Connecting to Oracle from Microsoft Excel using ODBC Driver for Oracle

You can use Microsoft Excel to access data from a Oracle database using ODBC connector. With ODBC Driver, you can import the data directly into an Excel Spreadsheet and present it

as a table. Make sure that you use matching Excel and ODBC Driver, e.g. if you have installed a 64-bit ODBC Drive, you will need to use the 64-bit version of Excel.

When working with Microsoft Excel, there are different ways of retrieving data from various data sources using our ODBC drivers:

- Connecting Excel to Oracle with Get & Transform (Power Query)
- Connecting Excel to Oracle with Data Connection Wizard (Legacy Wizard)
- Connecting Excel to Oracle with the Query Wizard
- Connecting Excel to Oracle with Microsoft Query
- Connecting Excel to Oracle with PowerPivot
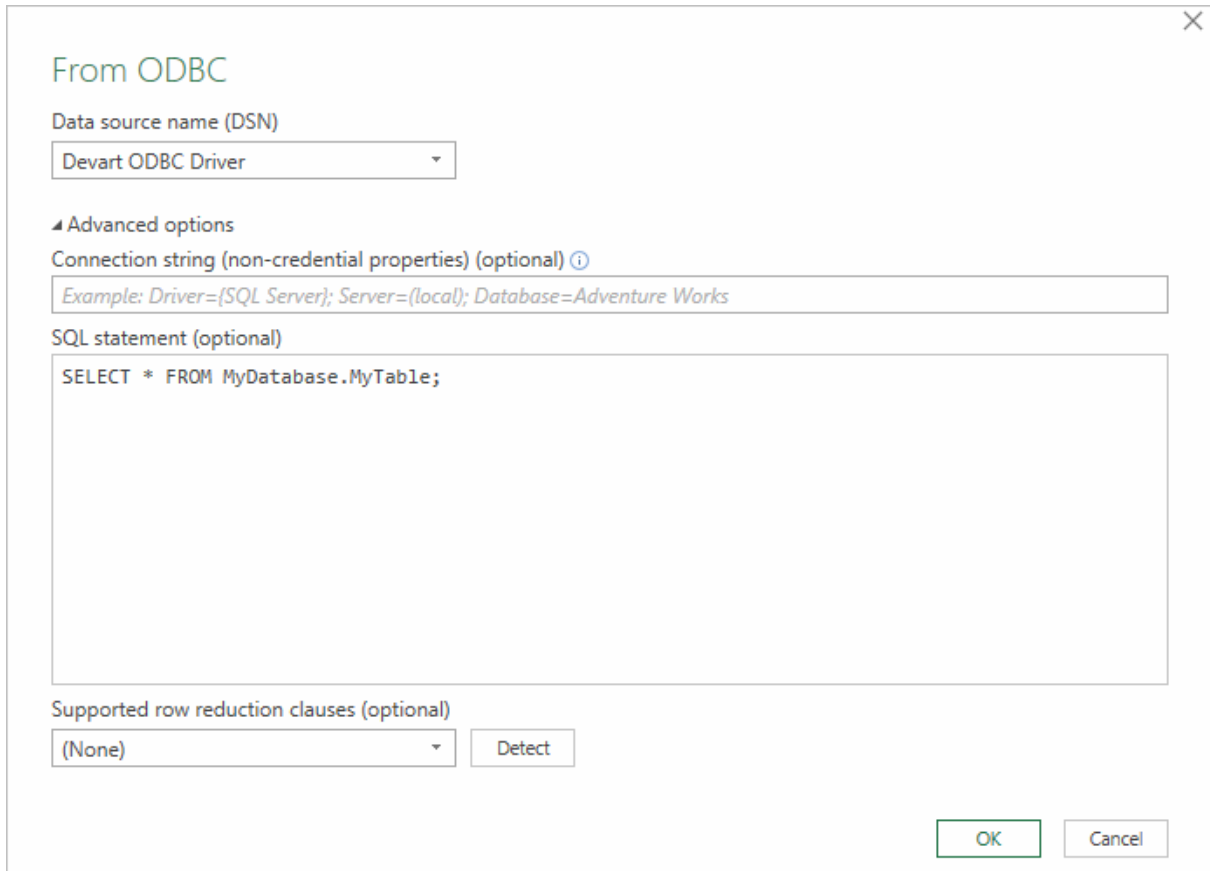
Connecting Excel to Oracle with Get & Transform (Power Query)

You can use Get & Transform (Power Query) to connect to Oracle from Excel with ODBC. This method assumes that you've installed an ODBC driver for Oracle.

1. Click the **Data** in Excel, then expand the **Get Data** drop-down list. Click **From Other Sources > From ODBC**.
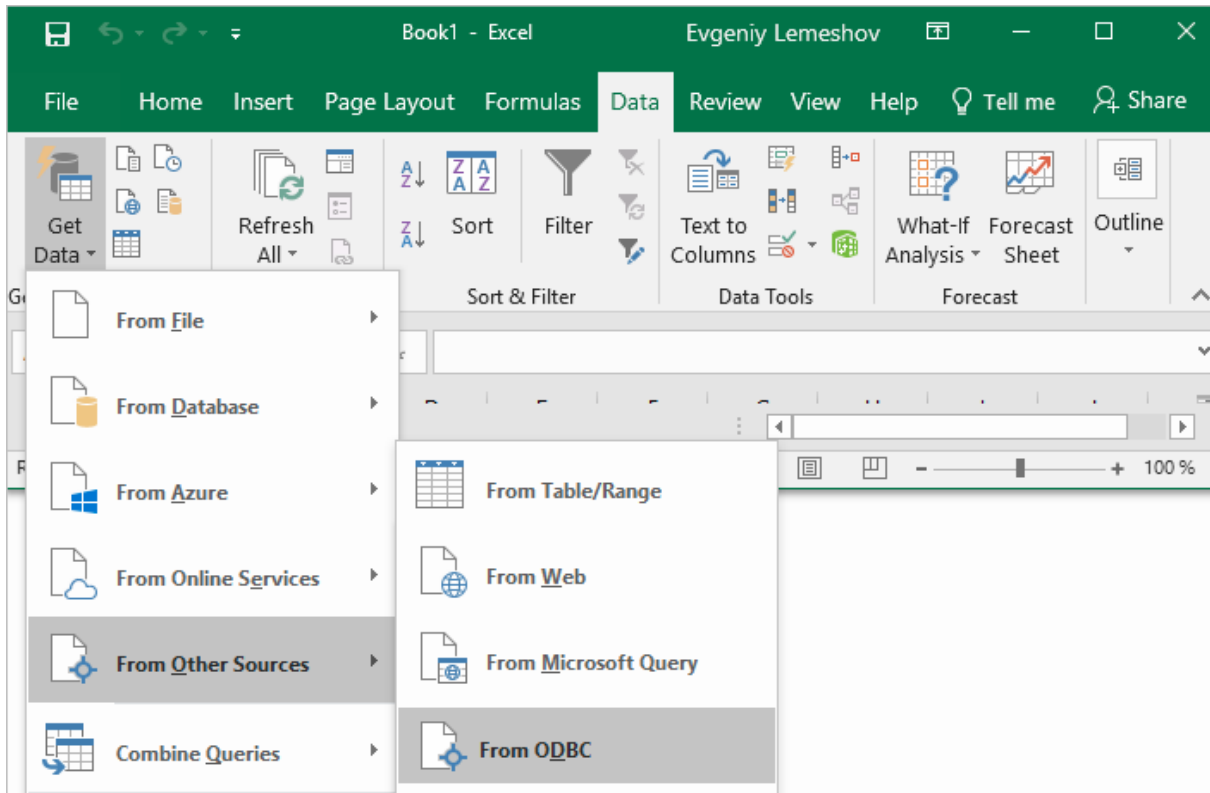
2. In the **From ODBC** dialog, choose your data source name (DSN). If you haven't configured your ODBC driver yet, you can expand the **Advanced Options** dialog box and enter the connection string for your data source (without credentials, which are defined in the credentials dialog box in the next step). Additionally, you can enter an SQL statement that will be executed right after establishing a connection to the data source. Click **OK**.
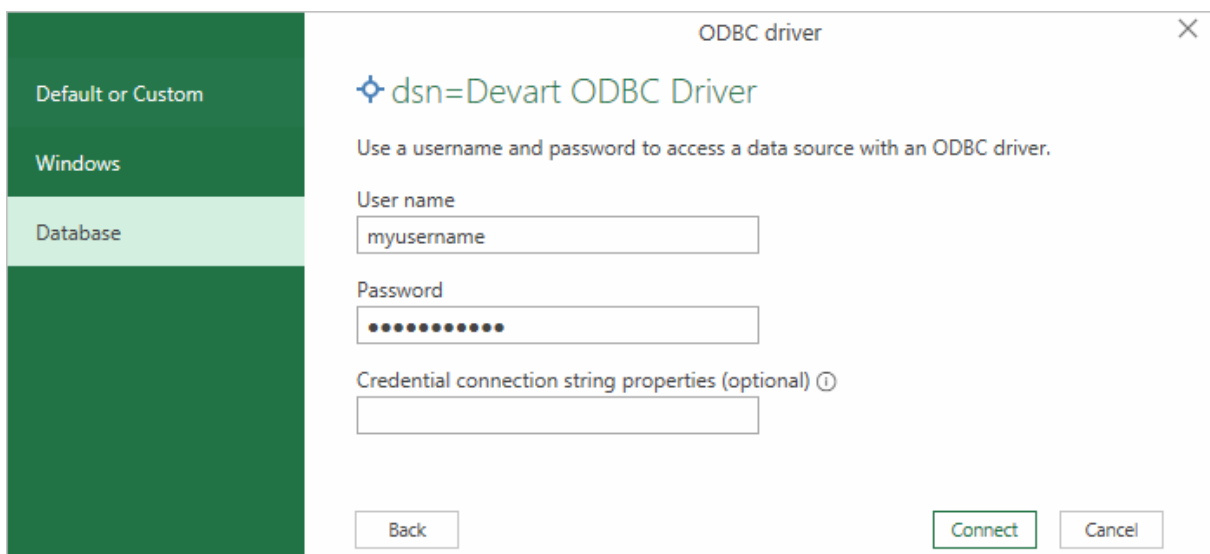
3. If you're using a database username or password, select **Database** and enter your credentials in the dialox bog, then click **Connect**.

If your database is not password-protected or you've already specified your credentials in the ODBC data source settings, select **Default or Custom** and press **Connect**
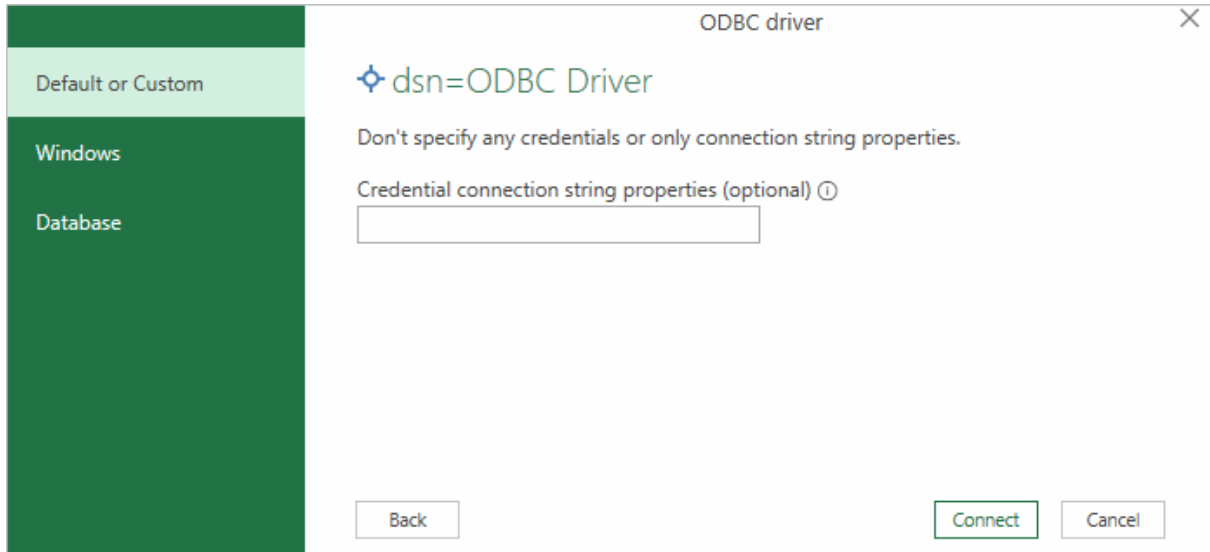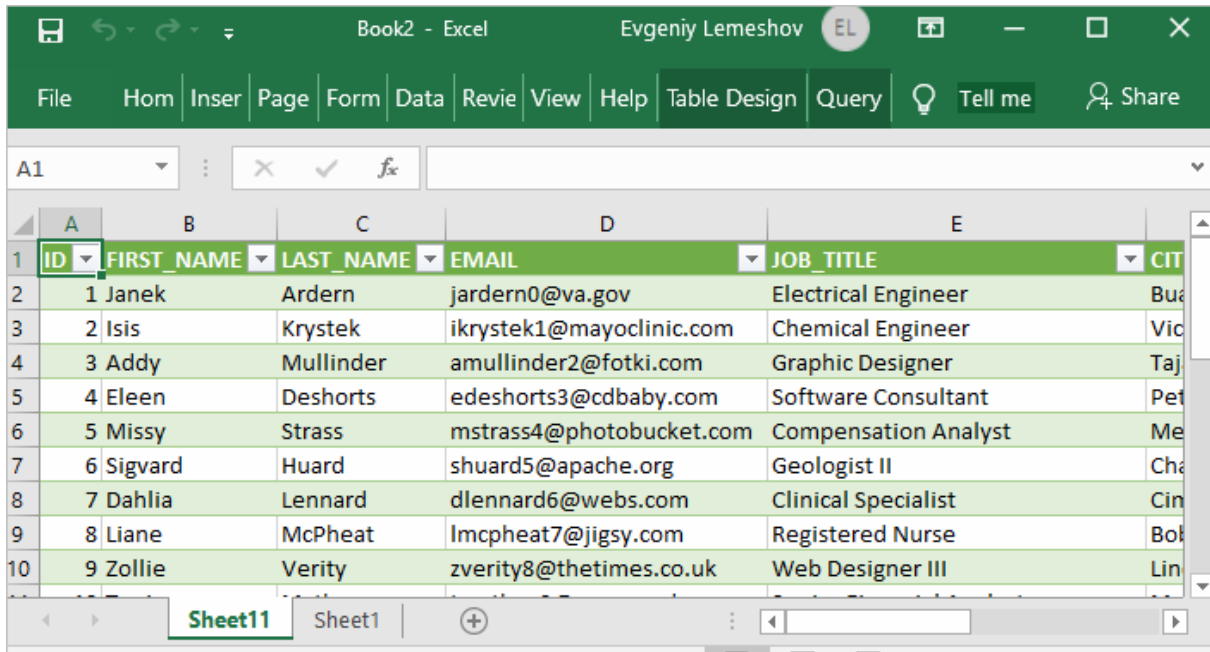
4. In the window that appears, select the table you want to retrieve data from, and click **Load**.
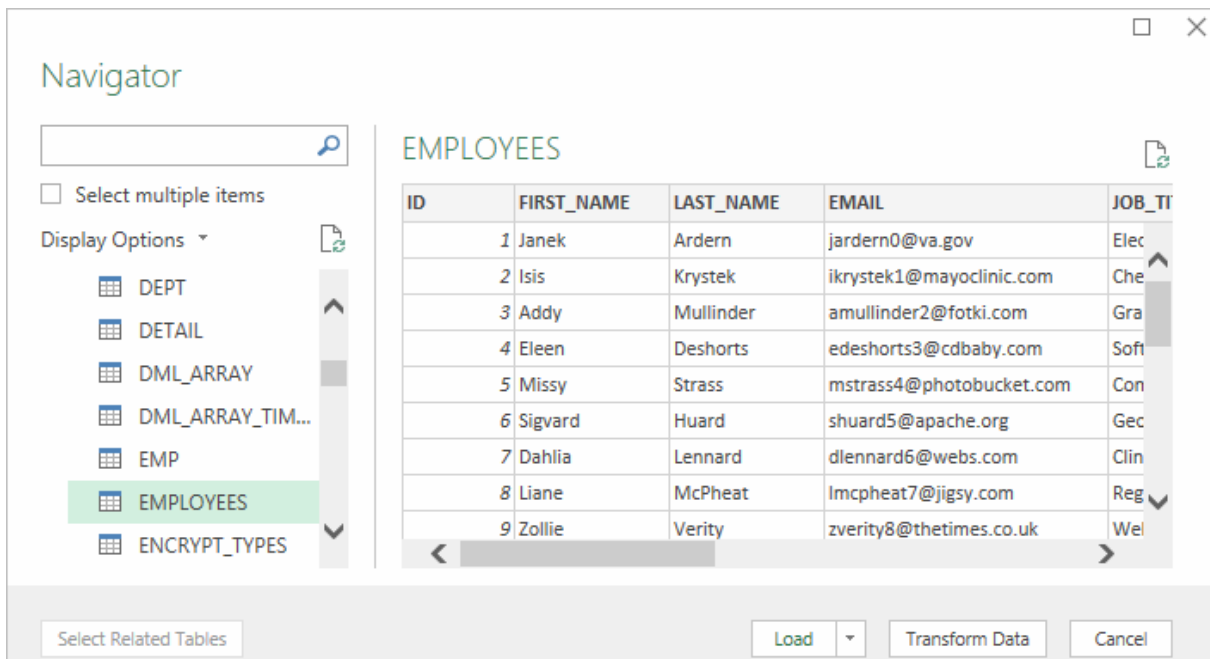
The data from the table will be a displayed in an Excel spreadsheet where you can further work with it.

Connecting Excel to Oracle with Data Connection Wizard (Legacy Wizard)

You can use this option to connect to OLE DB or ODBC external data source that has already been defined.

1. In Excel, go to the **Data** tab. Click **From Other Sources**, and then click **From Data Connection Wizard**.

2. In the opened dialog, select **ODBC DSN** and click **Next** to continue.

3. Now select a data source you want to connect to, and click **Next**.

4. To connect to the table containing the required data, select its name and click **Next** to enter and save information about your new file or click **Finish**.

5. In the **Import data** dialog, you can select the way your data will be viewed in Excel and the place where to put it in the worksheet, and click **OK**.

6. The required data is now displayed in the existing Excel worksheet.

Connecting Excel to Oracle with the Query Wizard

You can use this option to create a simple query for retrieving data from Oracle to Excel via ODBC driver.

1. Open Excel, in the main menu, click the **Data** tab.

2. Click the **From Other Sources** dropdown menu, and then click **From Microsoft Query**.

3. In the appeared dialog, you can choose the data source you want to connect to.

4. After a successful connection, you can select the data you want to be displayed in Excel and click **Next**.

5. The next two steps allow filtering and sorting the data. Click **Next** to skip these procedures.

6. If you plan to further use the query, you can save it by clicking the **Save** button on the right.

7. Select **Return Data To Microsoft Excel** and click **Finish**.

8. In the **Import data** dialog, you can select the way your data will be viewed in Excel and the place where to put it in the worksheet, and click **OK**.

9. The required data is successfully imported to Excel.

Connecting Excel to Oracle with Microsoft Query

You can use this option to create a more complex query for retrieving Oracle data to Excel via ODBC driver.

1. Start Excel, click the **Data** tab.

2. In the appeared ribbon, click **From Other Sources**, and then click **From Microsoft Query**.

3. In the next dialog, choose the data source you want to connect to (e.g., using data source name - Devart ODBC Oracle). Uncheck **Use the Query Wizard to Create/Edit Queries** and click **OK**.

4. Now you can select the tables you want to add to your query. When you finish, just click the **Add** button.

5. In the graphical editor, you can filter rows or columns of data, sort data, join multiple tables, create a parameter query, etc.

Connecting Excel to Oracle with PowerPivot

You can use PowerPivot - an Excel add-in to perform data analysis and create complex data models. To load the required data, do the following:

1. In Excel, click the **PowerPivot** tab, then click **Manage** to go to the PowerPivot window.

2. In the opened window, click **From Other Sources**.

3. When the **Table Import Wizard** opens, select **Others (OLEDB/ODBC)** and click **Next**.

4. In the **Specify a Connection String** window, click the **Build** button.

5. In the **Data Link Properties** dialog, specify the data source you want to connect (e.g., using data source name - Devart ODBC Oracle), and then click **Next**.

6. Now you should choose how to import the data (either select a table from the list or write a query to specify the data to be imported).

7. When the Import operation succeeded, click the **Close** button. The retrieved data is inserted in the active worksheet.

Request Support   ODBC Forum   Provide Feedback

4.6   Using in SQL Server Management Studio

This section describes how to establish and troubleshoot a connection to Oracle from SQL Server Management Studio using ODBC Driver for Oracle.

- Using in SQL Server Management Studio (SSMS)

- Troubleshooting in SSMS

Request Support   ODBC Forum   Provide Feedback

4.6.1   Using in SSMS

Requirements

In order to avoid incorrect integration with MS SSMS, the working environment must meet the following conditions:

- The data source must be a configured system DSN. Refer to the Driver Configuration article to learn how to configure a System DSN

- The driver, studio, and SQL Server must be of the same bitness. For example, if you are using 64-bit SQL Server Management Studio on 64-bit Windows platform, then configure the 64-bit version of the driver using ODBC Administrator launched from %windir% \system32\odbcad32.exe. Otherwise, configure the driver using the 32-bit version of ODBC Administrator - launch it from %windir%\SysWOW64\odbcad32.exe.

- ODBC Driver for Oracle and SQL Server must be installed on the same computer.

- .NET Framework 4.5 must be installed on the computer.

Connecting to Oracle from SQL Server Management Studio using ODBC Driver for Oracle

You can use the Microsoft SQL Server Management Studio to connect your Oracle data to an SQL Server instance. Linked Server is a tool of MS SQL Server that allows to execute distributed queries to refer tables stored on non-SQL Server datbase in a single query. With linked servers, you can execute commands against different data sources such as Oracle and merge them with your SQL Server database. You can create a linked server with one of these methods: by using the options in the Object Explorer or by executing stored procedures.

Below are major advantages of using SQL Server Linked Servers to connect to Oracle:

1. The ability to connect other database instances on the same or remote server.
2. The ability to run distributed queries on heterogeneous data sources across the organization.
3. The ability to work with diverse data sources in the same way.

How to configure a SQL Server Linked Server to connect to Oracle

You can follow the steps to create a linked server for Oracle in SQL Server Management Studio by using Object Explorer:

1. Start your Management Studio and choose your SQL Server instance.
2. In the **Object Explorer pane**, expand the **Server Objects**, right-click on **Linked Servers** and then click on **New Linked Server**.
3. Configure your linked server in the dialog box:
   o Give a name for your server in the **Linked server** field.
   o Under **Server type**, select **Other data source** .
   o Choose **Microsoft OLE DB Provider for ODBC Drivers** in the **Provider** drop-down list.
   o In the **Data source** field, enter the name of your DSN, e.g. Devart ODBC Driver for Oracle . Alternatively, you can input the ODBC Driver connection string in the **Provider** field.

The linked server will appear under the Linked Servers in the Object Explorer Pane. You can

now issue distributed queries and access Oracle databases through SQL Server.

Retrieving Data From Oracle

Disable the **Allow inprocess option** of MSDASQL OLE DB Provider for ODBC Drivers. For this, find the **MSDASQL** provider in the list of Linked Servers and double-click on it
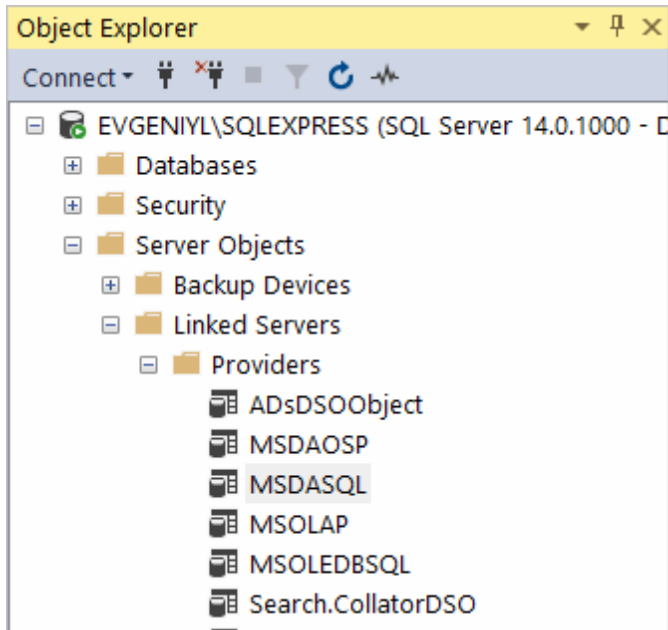
In the appeared **Provider Options** window, clear the **Allow inprocess** checkbox:
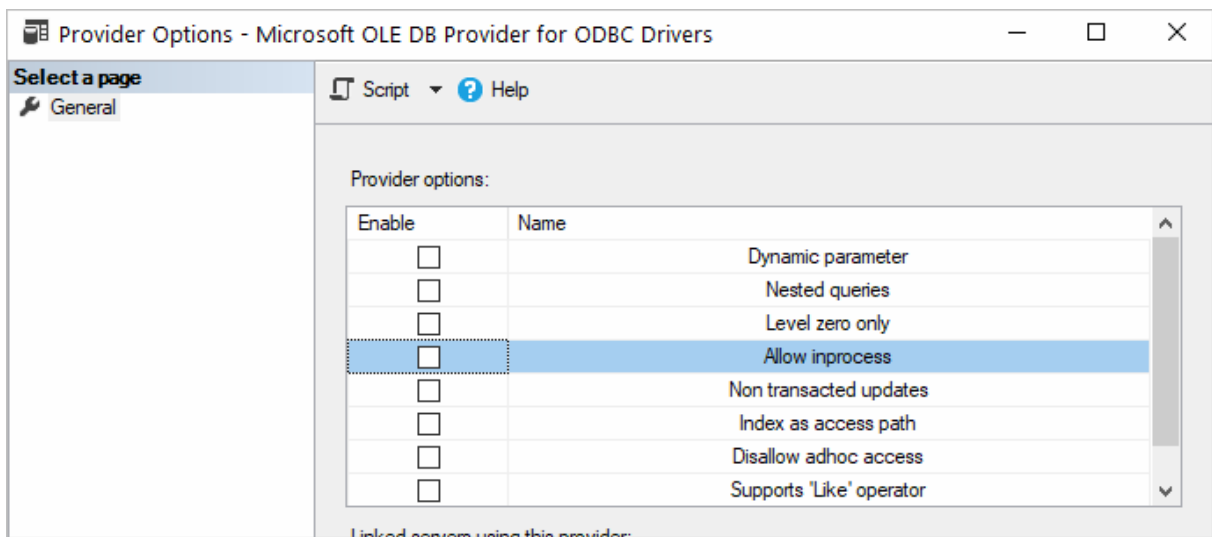
Create a new **Linked Server**
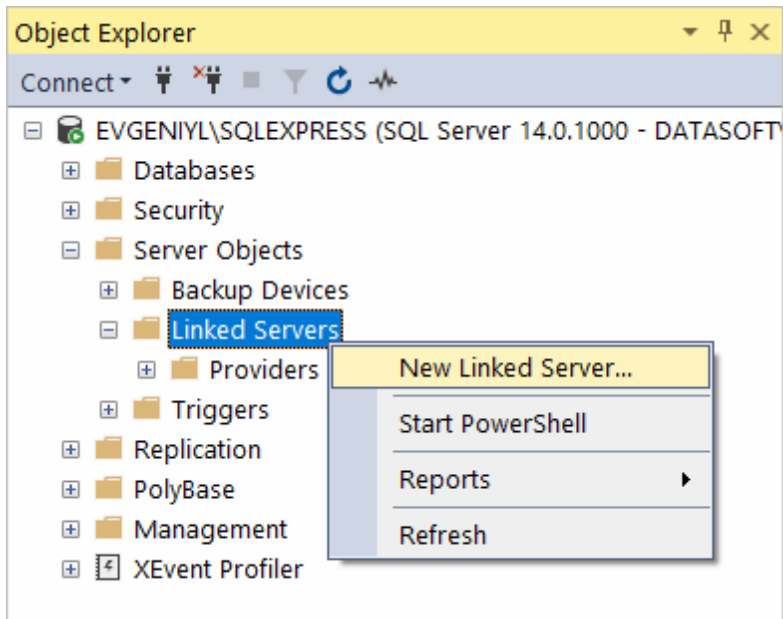
Make sure to select **Microsoft OLE DB Provider for ODBC Drivers** and specify the following parameters:
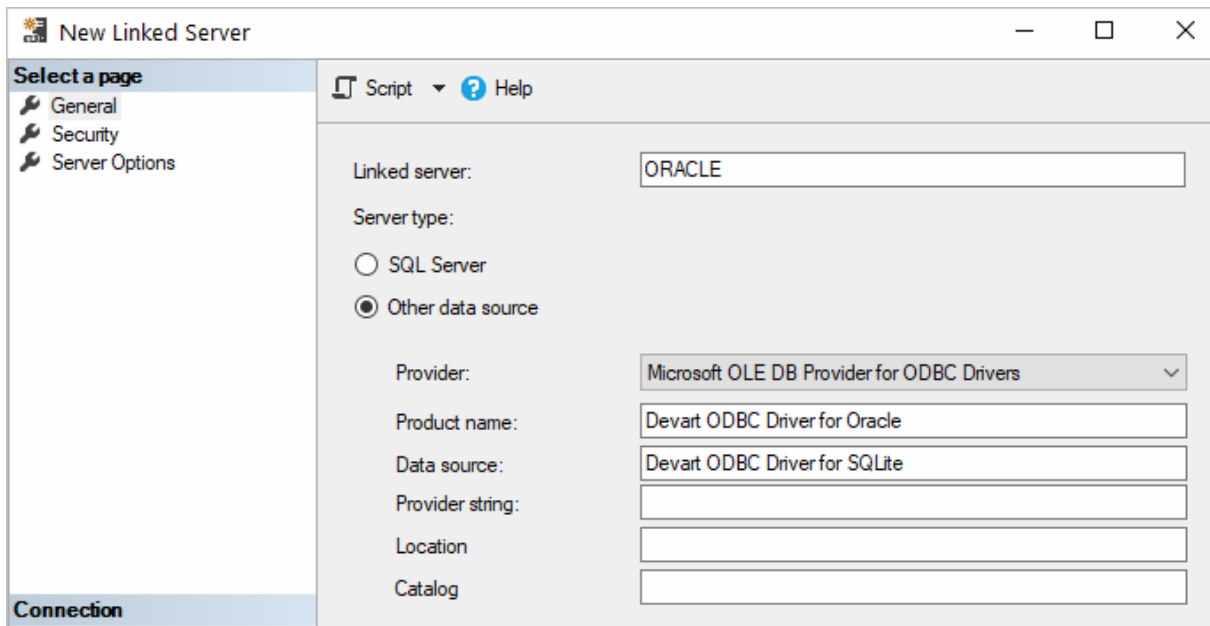
The Oracle tables are already available to be fetched. To query the linked server, click **New Query** in the toolbar:
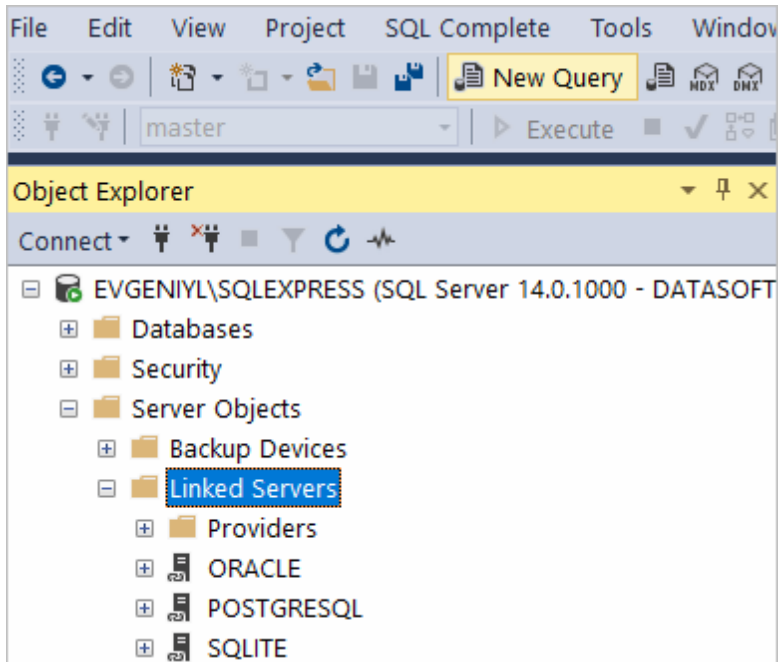
Enter your SQL query in the editor window and click **Execute** to run the query:
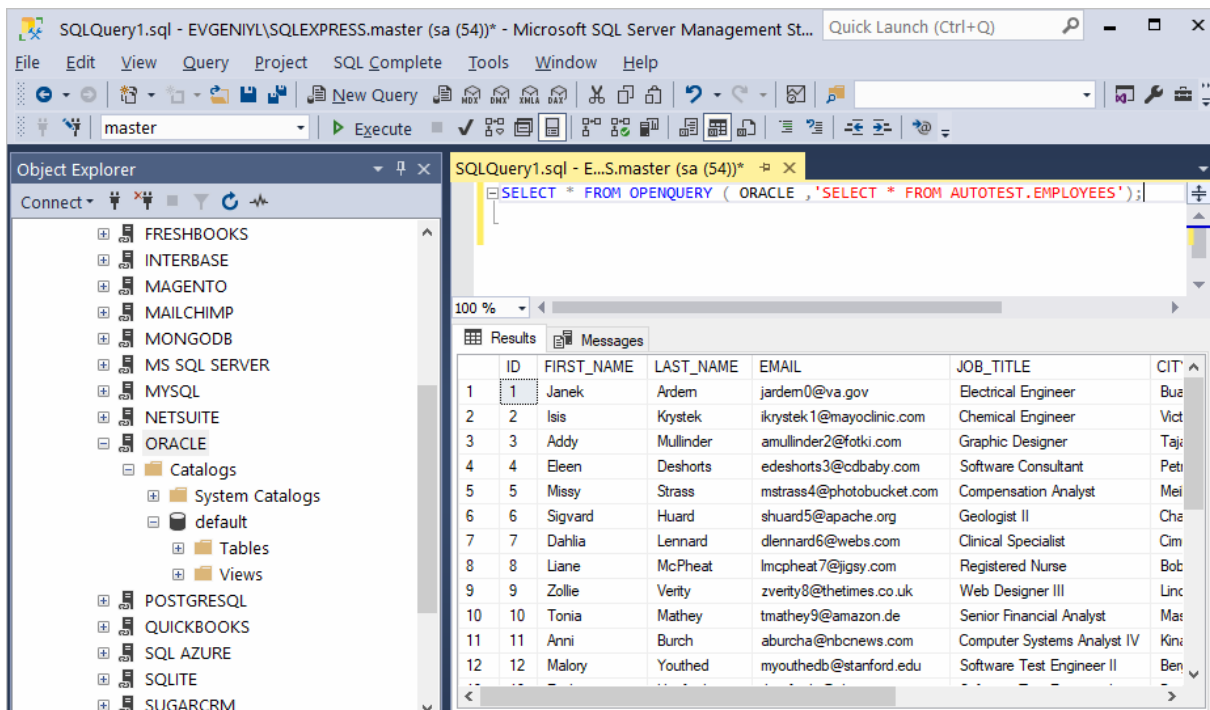
As a result, you can see the contents of the selected table retrieved directly from the Oracle account you are connected to.

Attention

If the Linked Server was created with the Allow inprocess option enabled, then you should delete this Linked Server and create it again with the Allow inprocess option disabled.

See also

- Troubleshooting SSMS

Request Support   ODBC Forum   Provide Feedback

4.6.2   Troubleshooting in SSMS

When creating a linked server in SSMS, most errors happen due to security issues with DCOM class MSDAINITIALIZE. We need to alter the DCOM Class MSDAINITIALIZE security settings to make it work.

Following are the steps:

1. Open Component Services (Start>Run>DCOMCNFG)

2. Expand Component Services>Computers>My Computer>DCOM Config

3. From the list of DCOM components on the right side, select **MSDAINITIALIZE** and go to its properties:
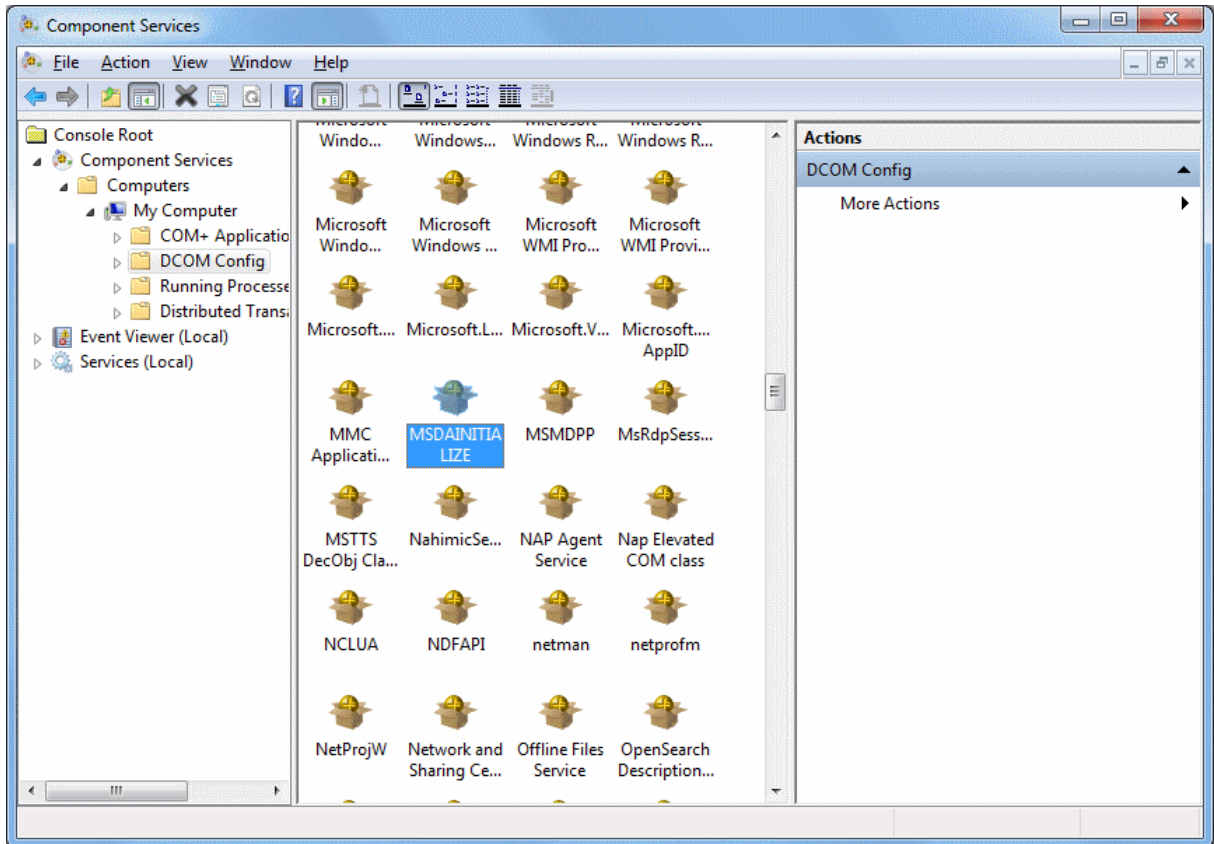
4. Go to the Security Tab, Choose 'Customize' and click on the 'Edit' Button:
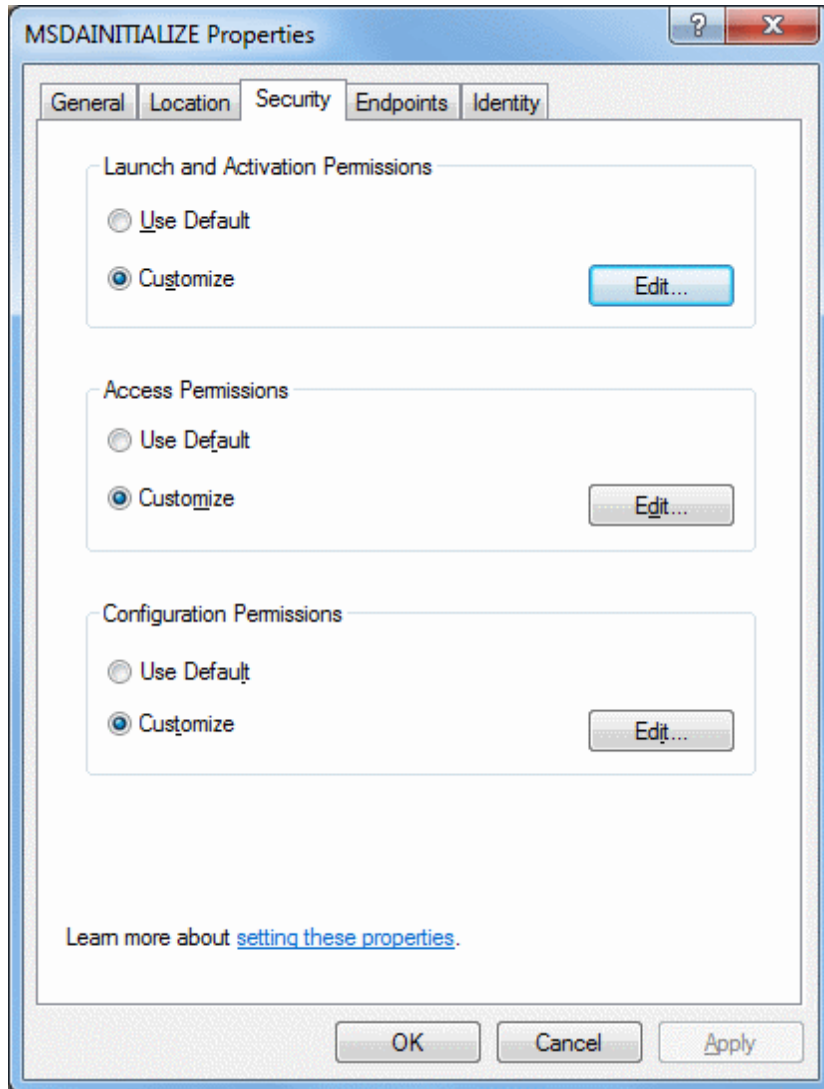
5. Add the Domain User who is accessing the linked server and 'Allow' all the permissions available (Local Launch, Remote Launch, Local Activation, Remote Activation). If you are connecting to SQL server using SQL account, you need to provide this permission to the account under which the SQL service is running.

6. Do this for all the 3 sections in the above screenshot.

To edit the Security settings, we followed the below steps:

1. Start > Run > Regedit

2. Find the Key: HKEY_LOCAL_MACHINE\SOFTWARE\Classes\AppID\{2206CDB0-19C1-11D1-89E0-00C04FD7A829}

3. Right Click>Permissions>Advanced>Owner Tab:
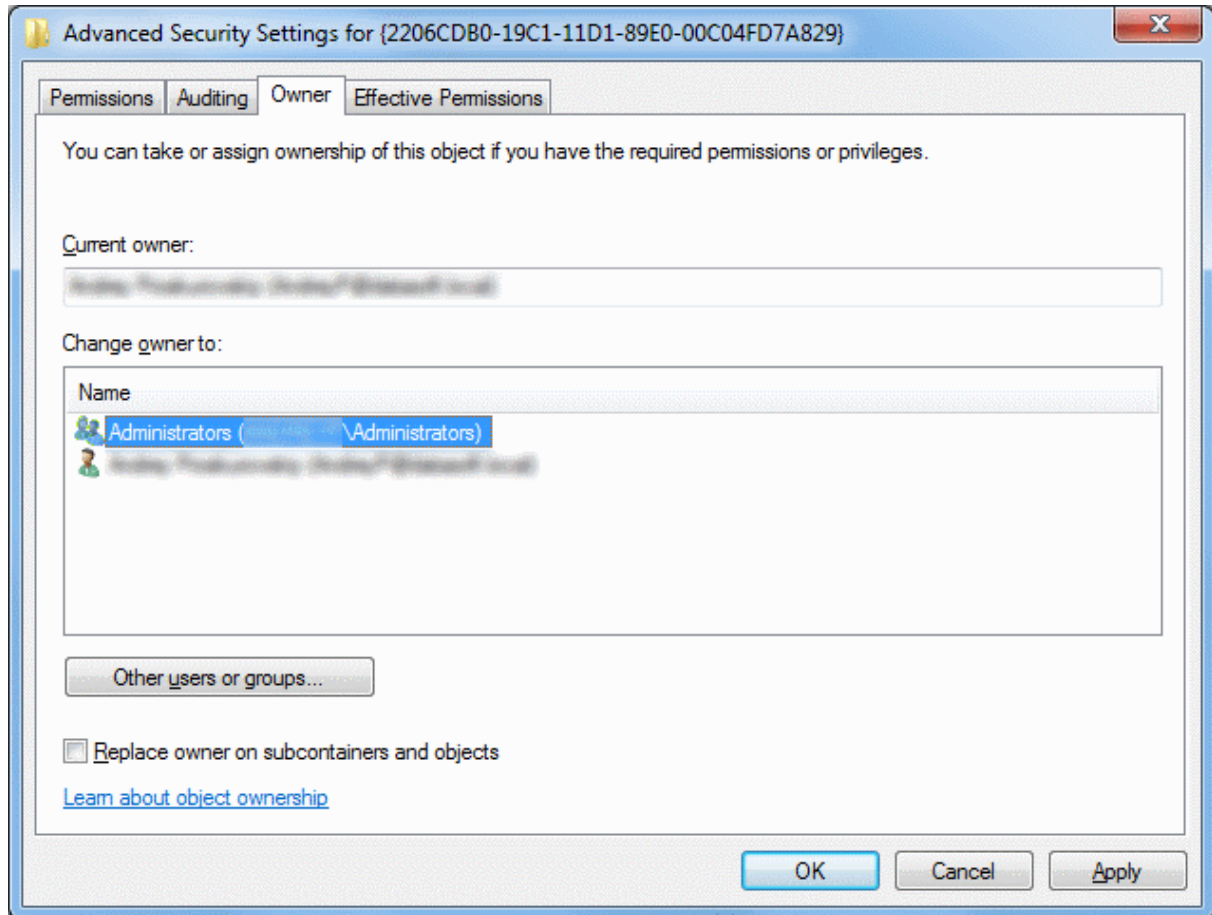
4. Change the owner to Administrators.

5. Now, grant 'Full Control' to Administrators:
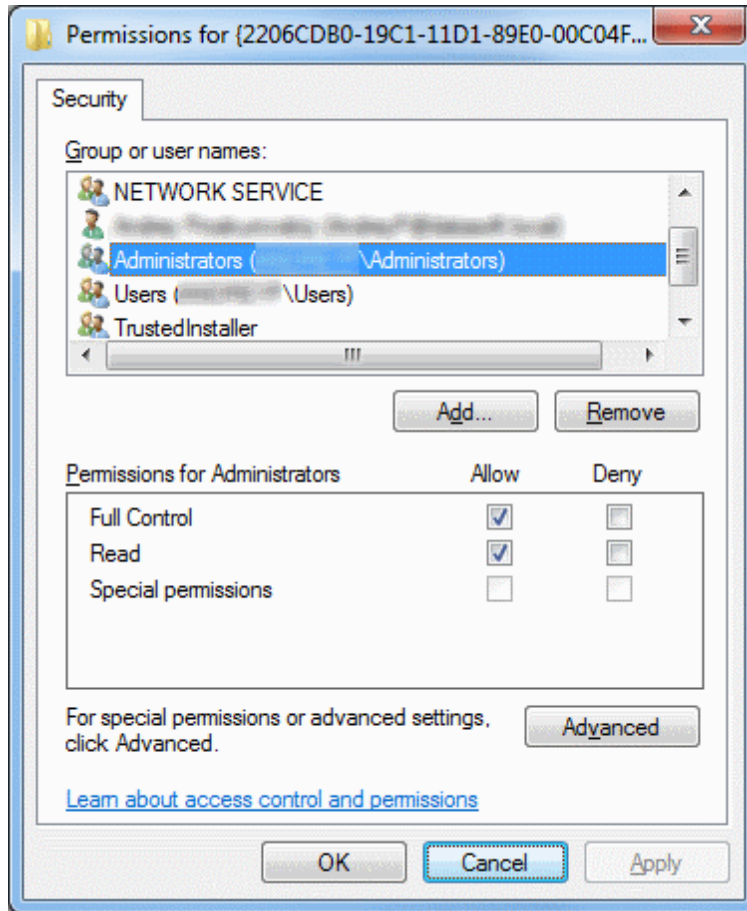
After this you should be able to edit MSDAINITIALIZE security settings .

See also

- Error message when you try to create an instance of an OLE DB provider in SQL Server: "Cannot create an instance of OLE DB provider"

Request Support   ODBC Forum   Provide Feedback

4.7   Using in OpenOffice and LibreOffice

Connecting to Oracle from OpenOffice and LibreOffice using ODBC Driver for Oracle

The article describes how to use Apache OpenOffice and LibreOffice to access ODBC data sources using the respective driver. You can access Oracle data from Open Office Base or LibreOffice Base — desktop database management systems. Note that the Windows version of OpenOffice is 32-bit, and you may get the error "The specified DSN contains an architecture mismatch between the Driver and Application" when trying to access a data source through a 64-bit ODBC driver. To get rid of the error message, set up the 32-bit version of the driver.

To connect to an ODBC data source from OpenOffice or LibreOffice using our driver for Oracle, perform the steps below:

1. Start OpenOffice or LibreOffice, click **Database** to open the **Database Wizard**.
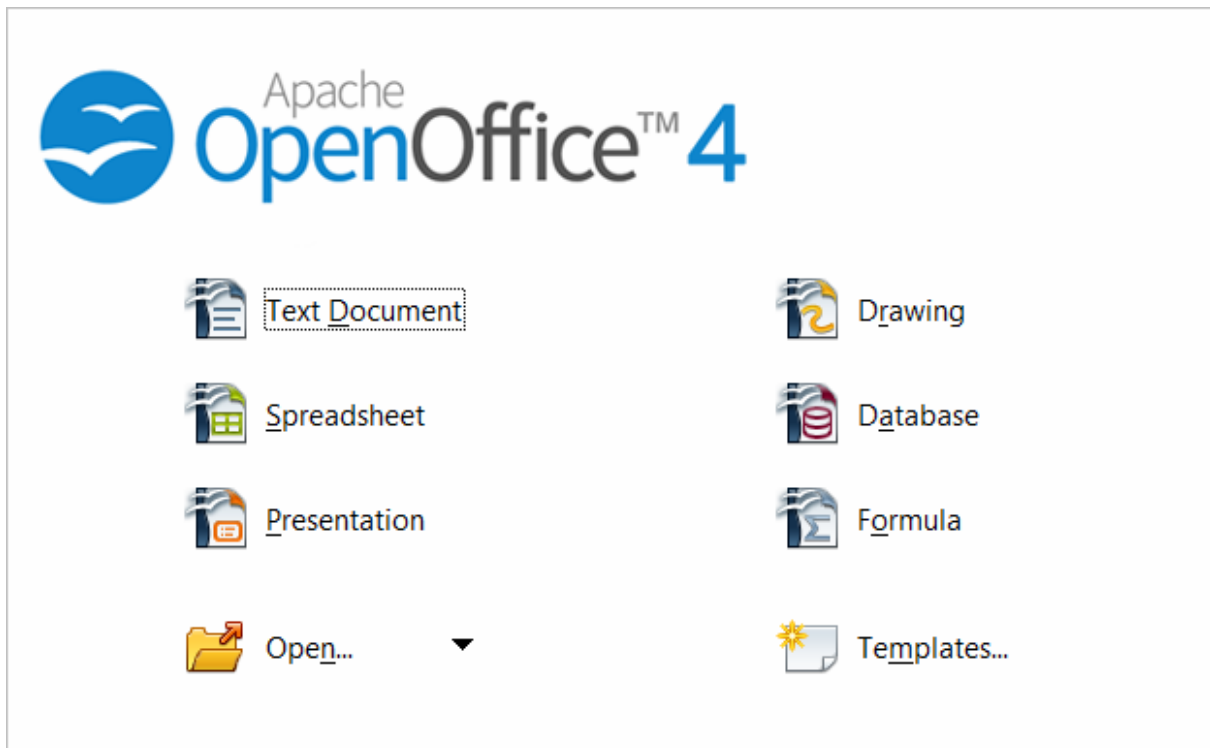
Alternatively, you can launch the **Database Wizard** from OpenOffice or LibreOffice Calc, Writer or any other tool by choosing **File > New > Database**.
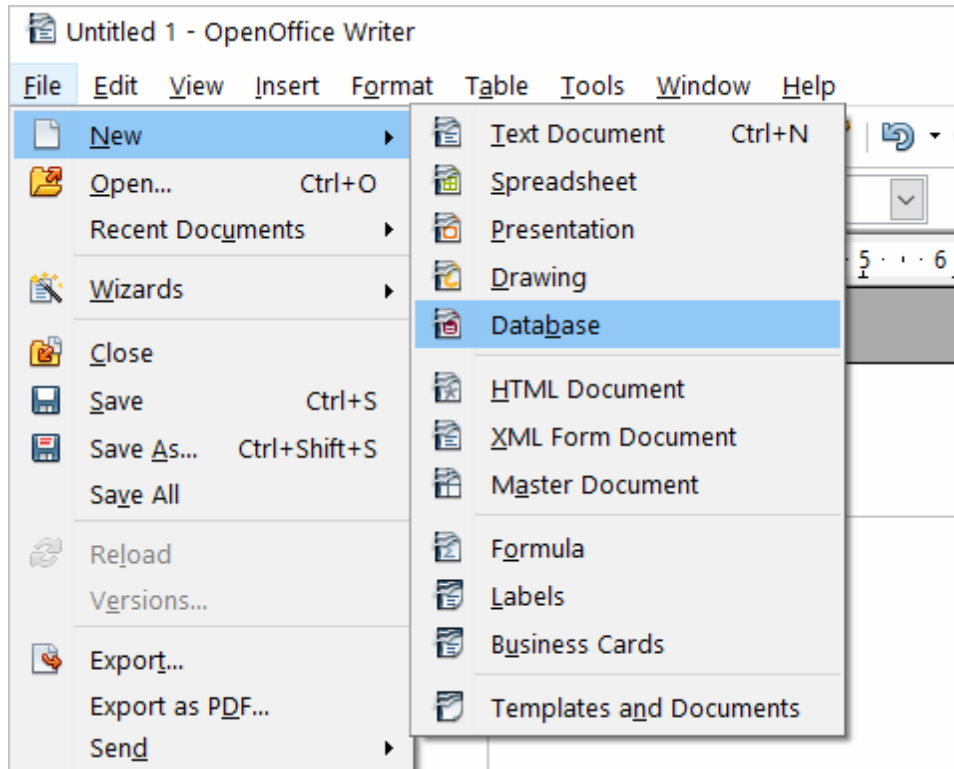
2. In the **Database Wizard dialog box**, click **Connect to an existing database**, select **ODBC** from the drop-down list, and click **Next**.
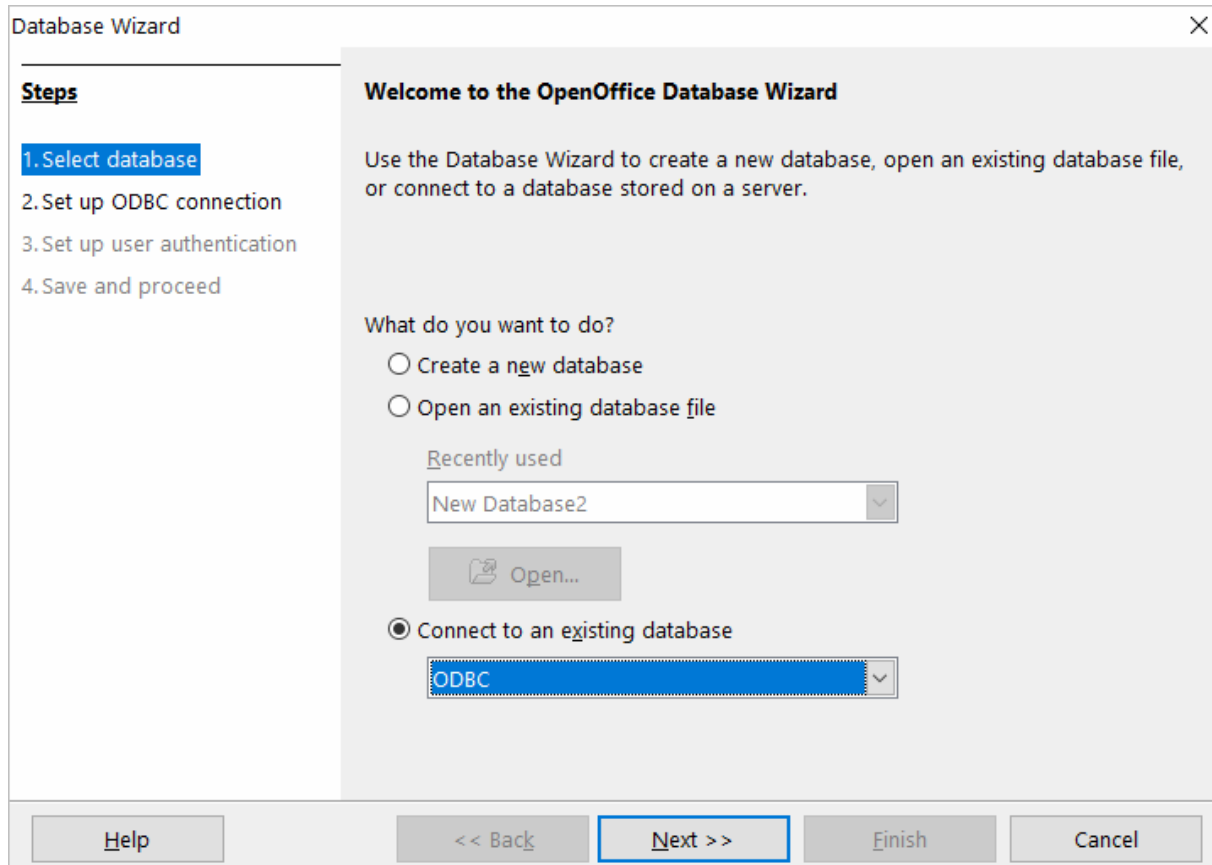
3. Specify the name of the data source you want to connect to. You can either type the name of your data source into the field, e.g. **ODBC Driver for Oracle**, or you can click **Browse**, double-click the data source you need, and then click **Next**.
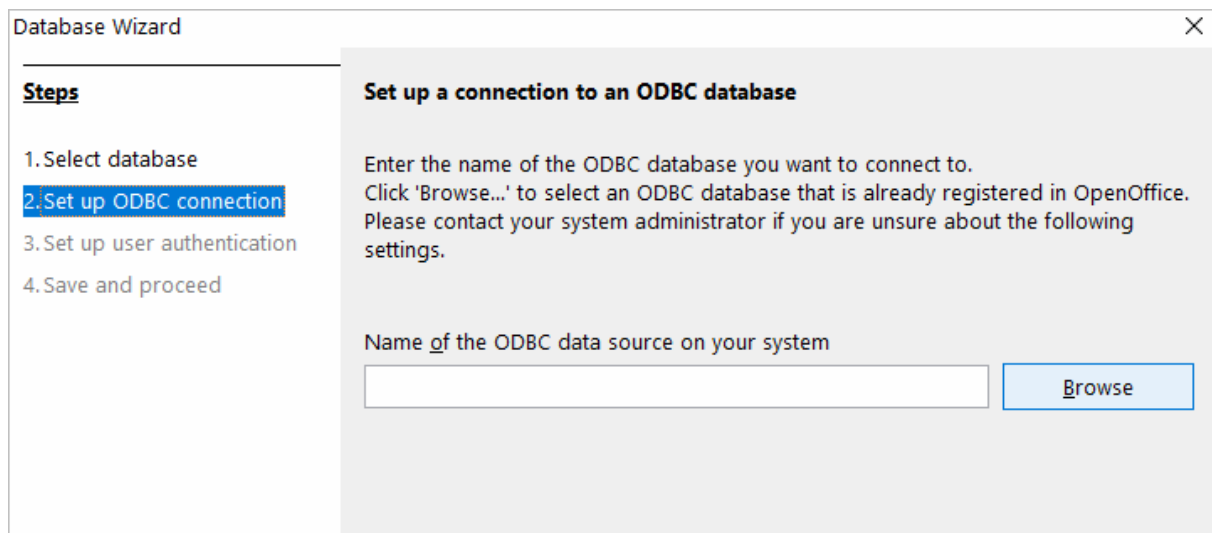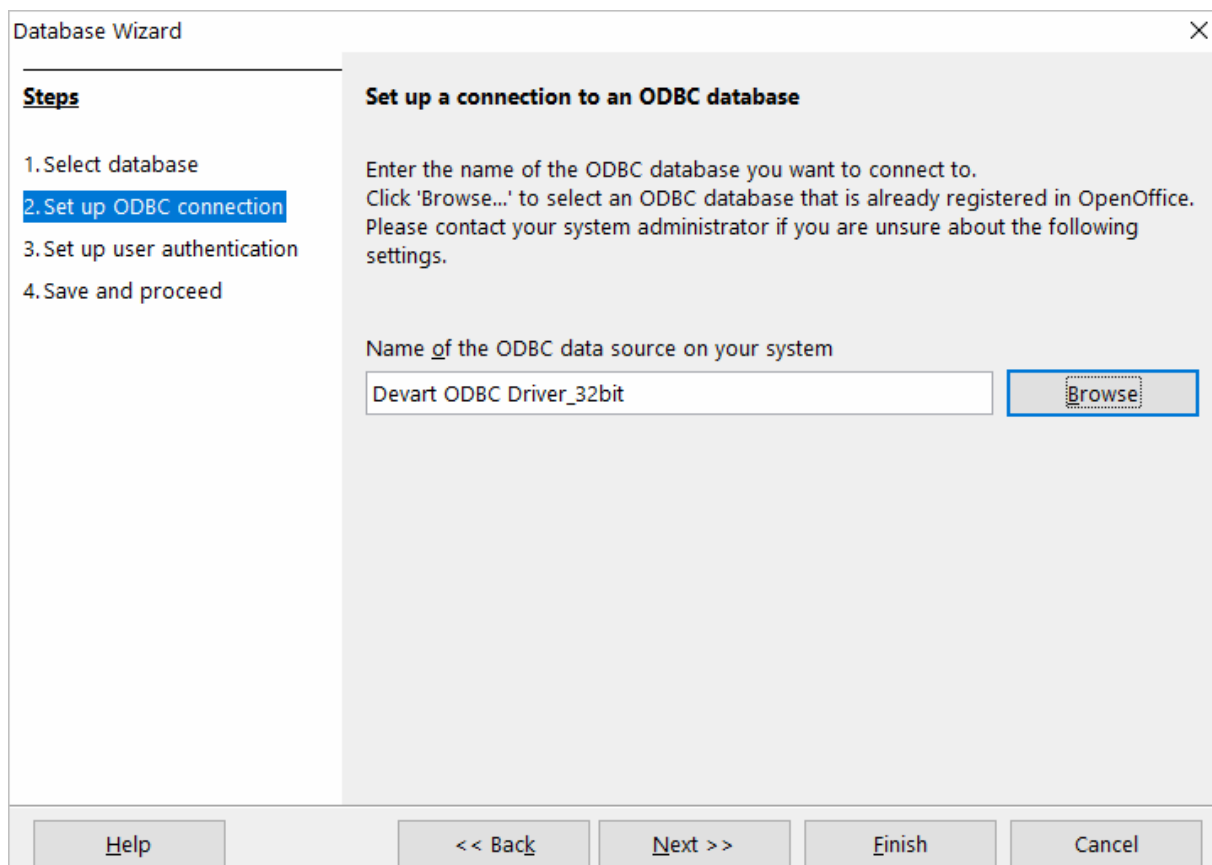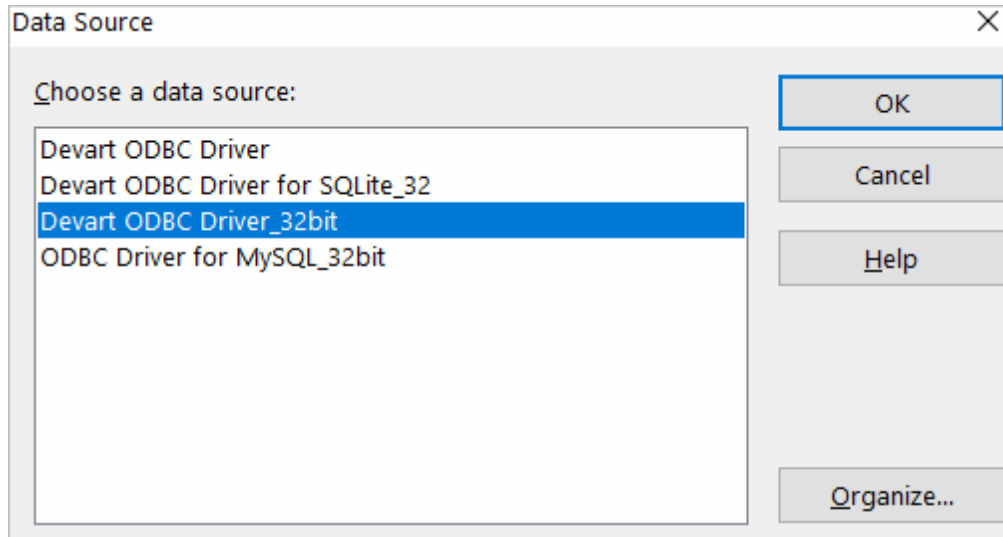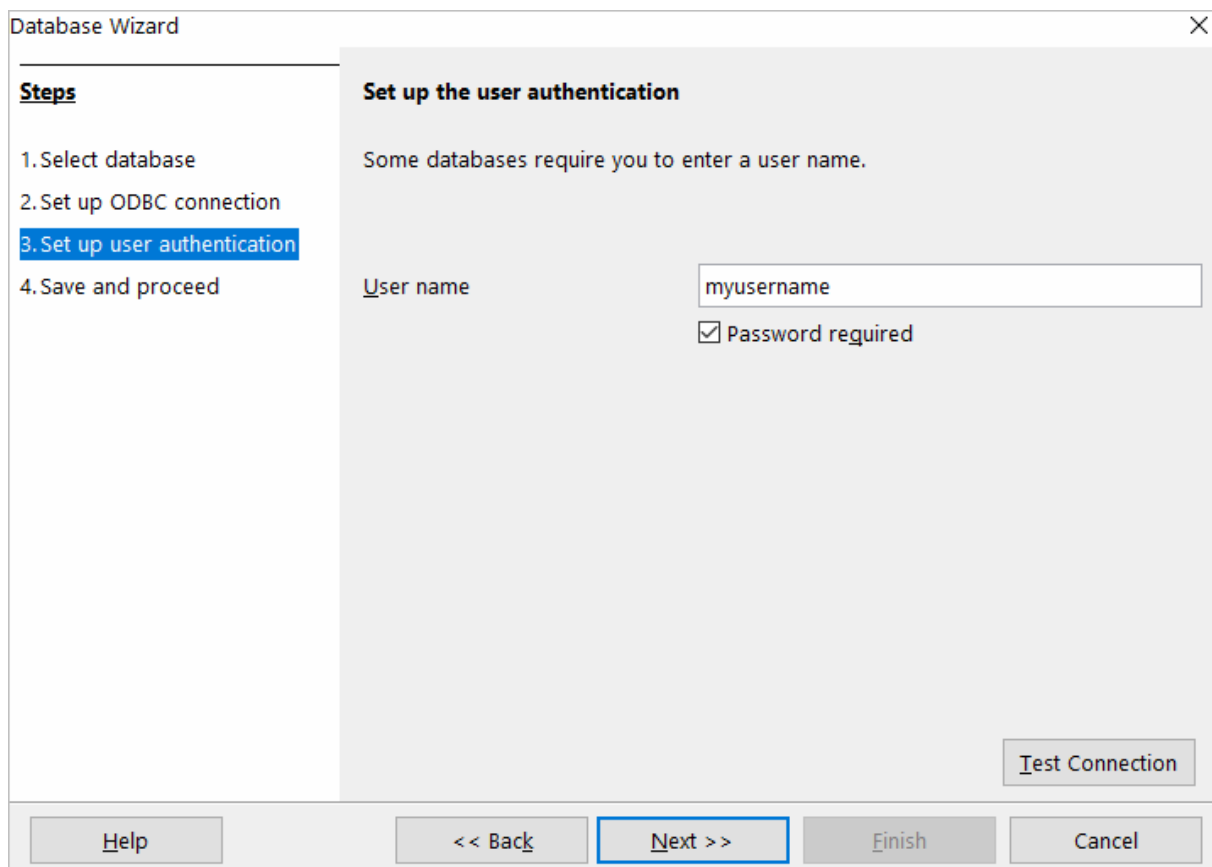
4. If your database requires a user name, type it into the **User name** field. If you are

connecting to a password protected database, check the **Password required** field.

Alternatively, you can specify these parameters in the data source settings of your ODBC

Driver for Oracle and leave these fields empty in **Database Wizard**.

To test the connection to your data source, click **Test Connection**, input your credentials and click **OK**.
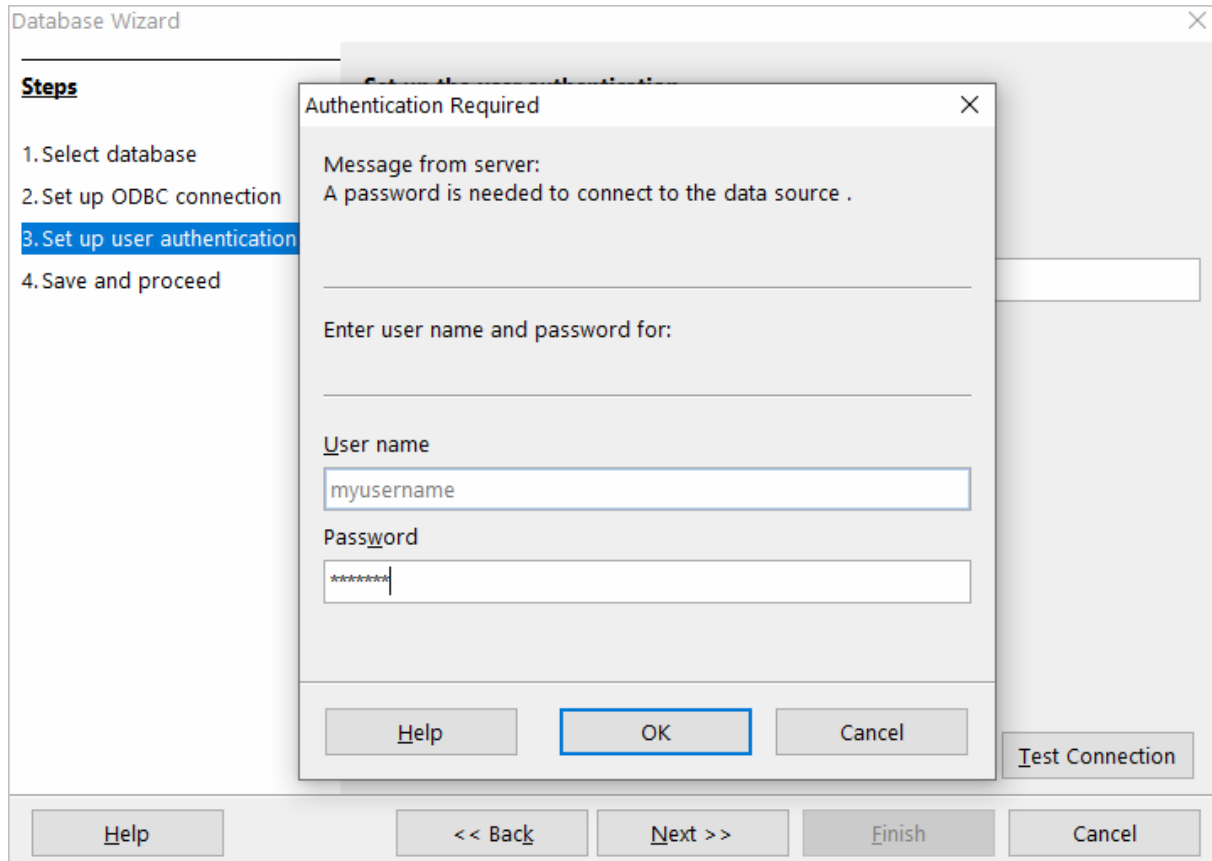
If you have entered valid credentials, you will see a success message. Click **Next** to proceed to the final step.

5. You can keep the default selection in this dialog box and click **Finish**.

Database Wizard ✕

Steps

1. Select database
2. Set up ODBC connection
3. Set up user authentication
4. Save and proceed

Decide how to proceed after saving the database

Do you want the wizard to register the database in OpenOffice?

◉ Yes, register the database for me

◯ No, do not register the database

After the database file has been saved, what do you want to do?

☑ Open the database for editing

☐ Create tables using the table wizard

Click 'Finish' to save the database.

Help		<< Back	Next >>	**Finish**	Cancel

You will be prompted to give a name to your new database and select the directory where you want to store it.
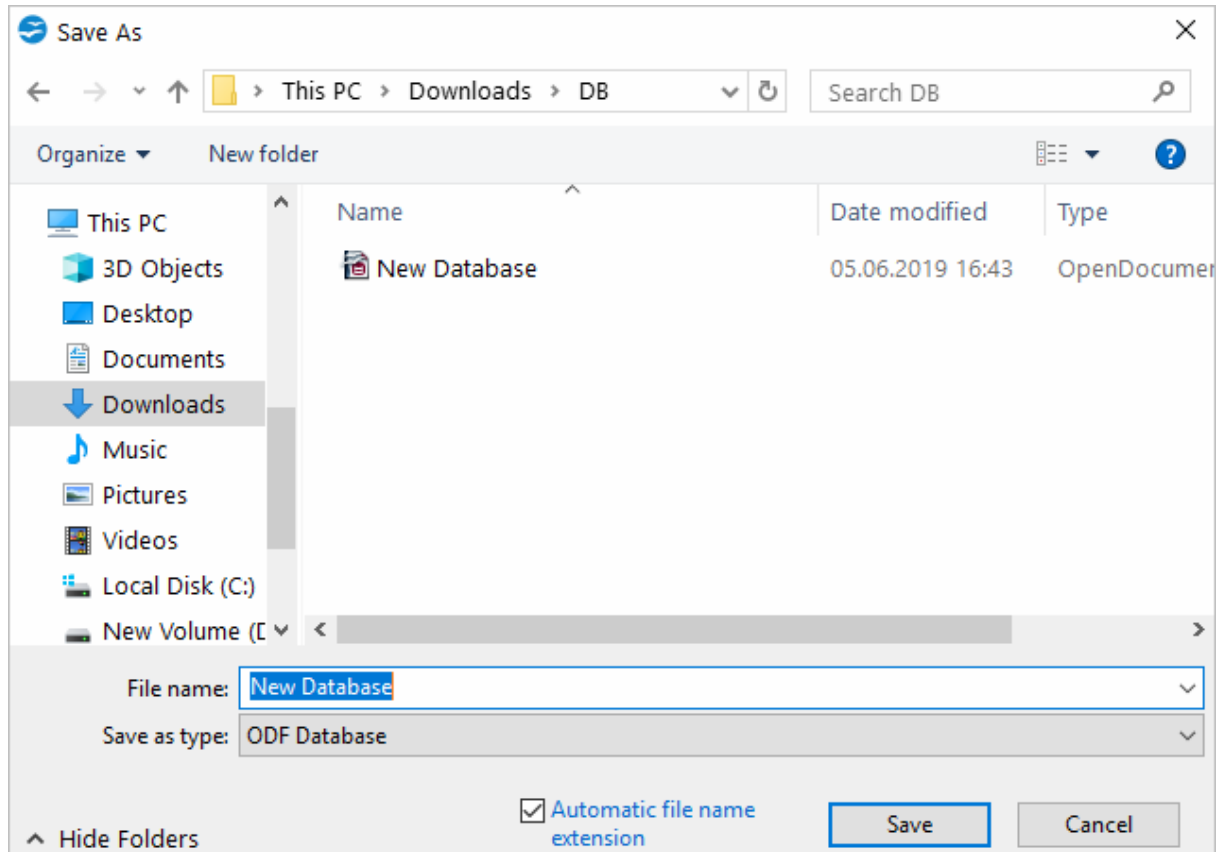
6. When the database opens, you will see the list of tables from your data source diplayed in OpenOffice or LibreOffice Base workspace. To view the data from a specific table, double-click the table name.

7. To create an SQL query, click **Queries** in the **Database** pane, then click **Create Query in SQL View…**
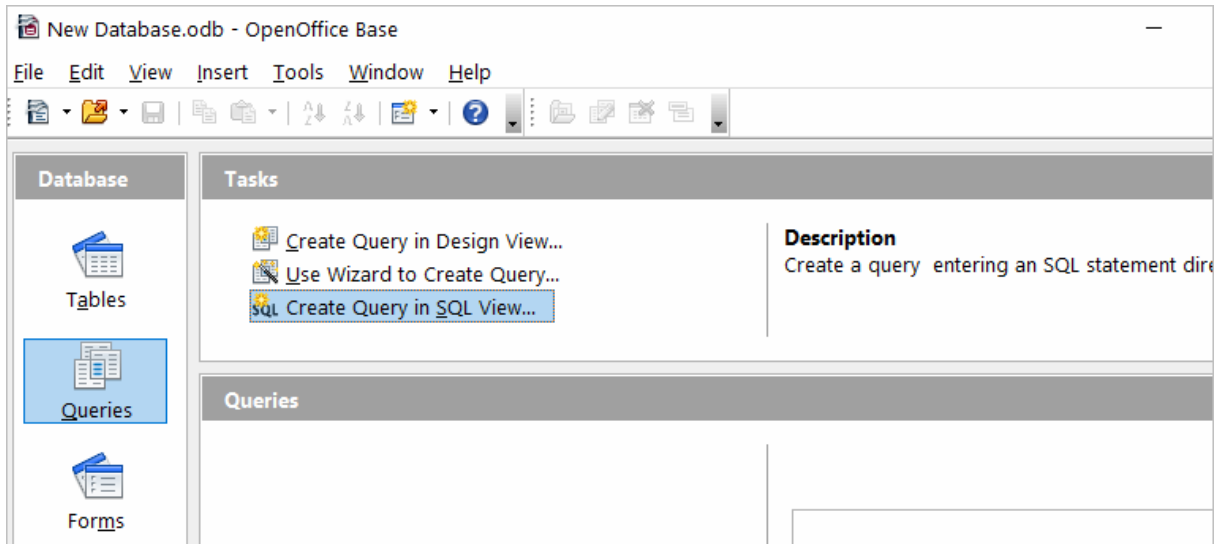
Enter your query in the query text box and click **Run Query (F5)**. The date will be fetched from the database and displayed in Open Office or LibreOffice, respectively.
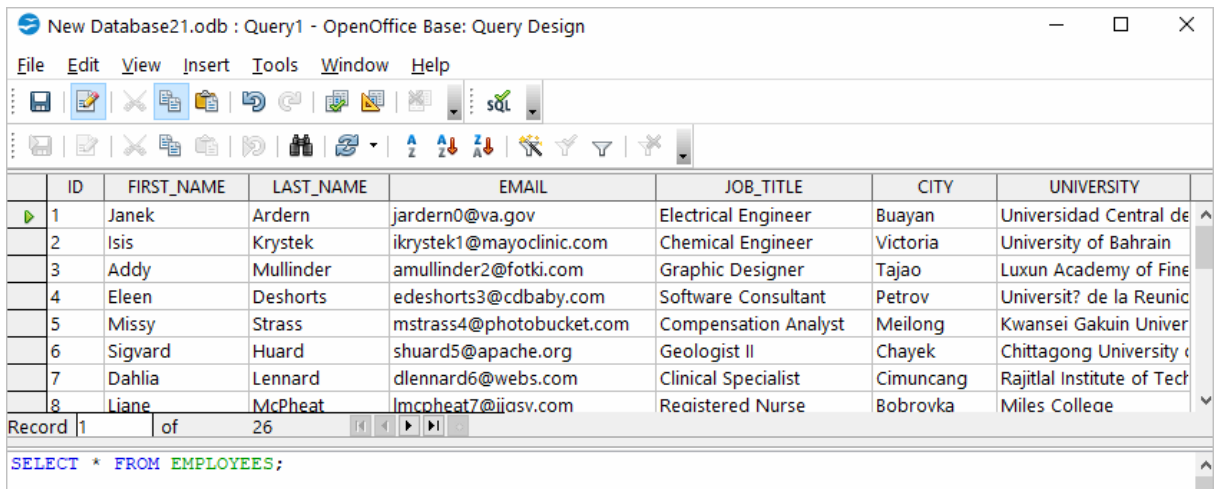
4.8   Using in PHP

Connecting to Oracle from PHP using ODBC Driver for Oracle

PHP is one of the most popular programming languages for website development. ODBC drivers are connectors that make PHP development database agnostic — your software written in PHP will function with any vendor's database management system. You can use functions like odbc_exec() to prepare and execute SQL statements against any databases like MySQL, SQLite, PostgreSQL, etc.

PHP-based projects usually require a data storage, whether a traditional database or a cloud-based database. You can establish a connection to them using ODBC interface. With our ODBC drivers, you can access various data sources and retrieve tables and fields from a database.

Below is a sample PHP script for accessing Oracle via ODBC. The script connects to Oracle database and fetchs all records from a table:

Step 1: Connect to ODBC data source

The *odbc_connect()* function is used to connect to an ODBC data source. Note that the function takes three mandatory parameters: the data source name, username and password. If your database is not password-protected or doesn't require a username, leave these parameters empty. In the following example, a connection is established using the *odbc_connect()* function in PHP.

```php
<?php
    $user = "myusername";
    $password = "mypassword";
    $ODBCConnection = odbc_connect("Driver={Devart ODBC driver for Oracle};D
```

Step 2: Execute an SQL statement

If connection is successful, the *odbc_exec()* function is used to execute a SELECT statement against the *dept* table in the *autotest* database.

```php
$SQLQuery = "SELECT * FROM autotest.dept";
    $RecordSet = odbc_exec($ODBCConnection, $SQLQuery);
```

Step 3: Print the result set

The *odbc_fetch_row()* function is used to return records from the result set. While *odbc_fetch_row()* returns rows, the *odbc_result_set()* function prints a set of result in HTML table. After all rows from the result set have been printed, the *odbc_close()* function closes the connection.

```php
    while (odbc_fetch_row($RecordSet)) {
        $result = odbc_result_all($RecordSet, "border=1");
```

```
    }
    odbc_close($ODBCConnection);
?>
```

You can modify this script by specifying general settings for each Devart ODBC driver to use any of them with your PHP projects.

    [Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

4.9   Using in Power BI

Importing Oracle Data into Power BI Through an ODBC Connection

Power BI is a popular business intelligence solution that is comprised of services, apps, and connectors that allow you to pull raw data from various sources and create meaningful reports. To connect Power BI to a data source such as Oracle, you can use a corresponding ODBC driver.

This tutorial explores how to connect to Oracle and import data into Power BI Desktop using an ODBC driver. It is assumed that you have already installed and configured a DSN for ODBC driver for Oracle.

1. Run Power BI Desktop and click **Get Data**.

2. Select the **Other** category in the **Get Data** dialog box, then select **ODBC**. Click **Connect** to confirm the choice.

3. In the **From ODBC** dialog box, expand the **Data Source Name (DSN)** drop-down list and select the previously configured DSN for Oracle

4. If you would like to enter a SQL statement to narrow down the returned results, click the **Advanced options** arrow, which expands the dialog box, and type or paste your SQL statement.

5. Click **OK**. If your data source is password-protected, Power BI will prompt you for user credentials. Type your **Username** and **Password** in the respective fields and click.

6. Now you should see the data structures in your data source. You can preview the contents of the database objects by clicking on them.

7. To load the Oracle data into Power BI for analysis, select the needed table and click **Load**.

    [Request Support](#)   [ODBC Forum](#)   [Provide Feedback](#)

Reserved.

4.10   Using in Python

Python module pyodbc and Oracle

Python is a popular general purpose scipting language that is also becoming popular among web developers. If you want to use an Oracle database as a data storage for your Pyhon app, this tutorial teaches how to connect Python to an Oracle database using ODBC driver, code samples included. There are many ways to connect to Oracle database from Python, including cx_Oracle, a Python extension module that enables access to Oracle database, msot of the applications though use either ODBC or JDBC driver to connect to Oracle. One of the most convenient methods to connect to an external database or access cloud data from Python is using the pyodbc Python module that implements the Python DB specification and allows you to easily connect Python appplications to data sources with the Devart ODBC driver for Oracle.

Installing pyodbc module and Oracle ODBC

If you don't have Python installed on your machine, go to the Python official website, download the installer for your system and run it. You will also need to install the **pyodbc** module — the easiest way to do that is by using the *pip install pyodbc* command in the Python interactive mode. Oracle database storage comes with ODBC support, but in order to connect to Oracle from Windows, macOS, or Linux, you must install ODBC driver for Oracle, otherwise the pyodbc module will not be able to establish a connection to the database.

Next, you need to download the ODBC Driver for Oracle. To use the ODBC driver as a translation layer between the application and the database, you need to configure it by following the installation instructions.

Connecting Python to Oracle database via ODBC Driver

Below is a Python code example that connects to Oracle using ODBC driver. First we import the pyodbc module, then create a connection to the database, insert a new row, and read the contents of the EMP table while printing each row to the Python interactive console. If you have ever connected to any relational database from Python, you will easily understand the code as we describe each line. To execute the script, you can type this code directly in the

interactive console or add the code to a file with the .py extension and run the file from the command prompt.

Step 1: Connect

The pyodbc module is imported to provide the API for accessing Oracle database. The code uses the driver named "Devart ODBC Driver for Oracle" to connect to the remote database. Once a connection is established, you can perform CRUD operations on the database.

```
import pyodbc
cnxn = pyodbc.connect('DRIVER={Devart ODBC Driver for Oracle};Direct=True;Ho
```

Step 2: Insert a row into Oracle table

Here's a simple example of how to execute an *insert* statement to test the connection to the database. The script inserts a new record to the EMP table.

```
cursor = cnxn.cursor()
cursor.execute("INSERT INTO EMP (EMPNO, ENAME, JOB, MGR) VALUES (535, 'Scott
```

Step 3: Retrieve data from Oracle table

The *cursor.execute()* function retrieves rows from the *select* query on a dataset. The *cursor.fetchone()* function iterates over the result set returned by *cursor.execute()* while the *print()* function prints out all records from the table to the console.

```
cursor = cnxn.cursor()
cursor.execute("SELECT * FROM EMP")
row = cursor.fetchone()
while row:
 print (row)
 row = cursor.fetchone()
```

Conclusion

We have shown you how to access Oracle database using Python, pyodbc and Devart ODBC driver. This tutorial uses only a handful of lines of code to establish a connection and demonstrate basic operations, but you can use the pyodbc module and our driver for Oracle in a enterprise application in the production environment. Feel free to download Devart ODBC driver for Oracle for free to evaluate its features. You can use the same steps to connect to any data source using a Devart ODBC driver: basically, all you have to do is modify the connection string, the pyodbc functions used in this tutorial work for any database.

Request Support   ODBC Forum   Provide Feedback

4.11 Using in QlikView

Connecting to Oracle from QlikView using ODBC Driver for Oracle

This tutorial describes how to connect and configure QlikView to retrieve data from Oracle for further analysis. QlikView is a data visualization tool that connects and pulls data from different popular databases like MySQL, MongoDB, Oracle, SQL Server, Postgres, etc. to present it in a single view. The business intelligence platform identifies relationships in your data and discovers patterns and opportunities to support your decision making.

QlikView supports the ODBC connectivity interface for communication with external data sources. An ODBC data source must be configured for the database you want to access. You can create an ODBC connection using a DSN during the ODBC driver installation or later.

To connect to an ODBC data source from QlikView using our driver for Oracle, perform the steps below:

1. Open the QlikView client application and click **File > New**. Close the **Getting Started** wizard and open **File > Edit Script (CTRL+E)**.
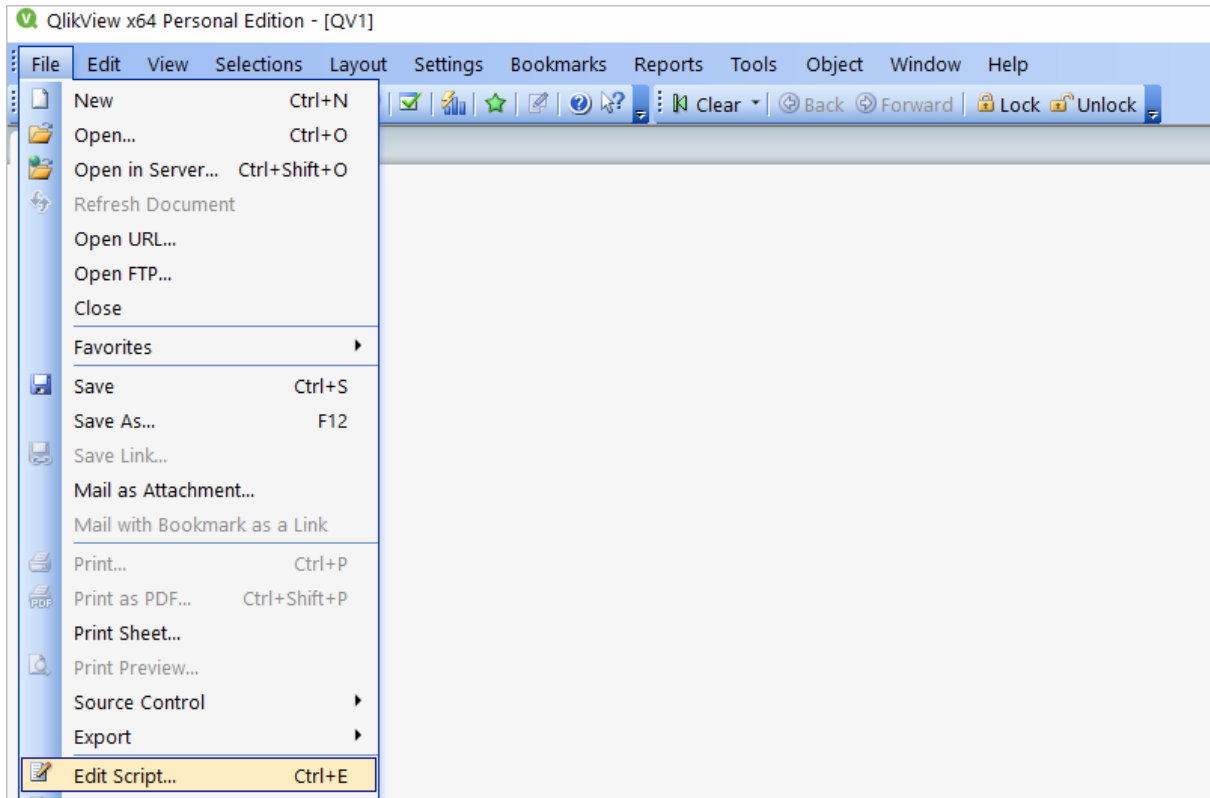
2. In the **Data** tab, choose **ODBC** from the **Database** drop-down and click **Connect**. Select the **Data Source** you created earlier, type in the **User ID** and **Password** if your database is password-protected. You can test the connection by choosing **Test Connection**. The **Connection Test succeeded** message should appear. Click **OK** to connect to your data source.
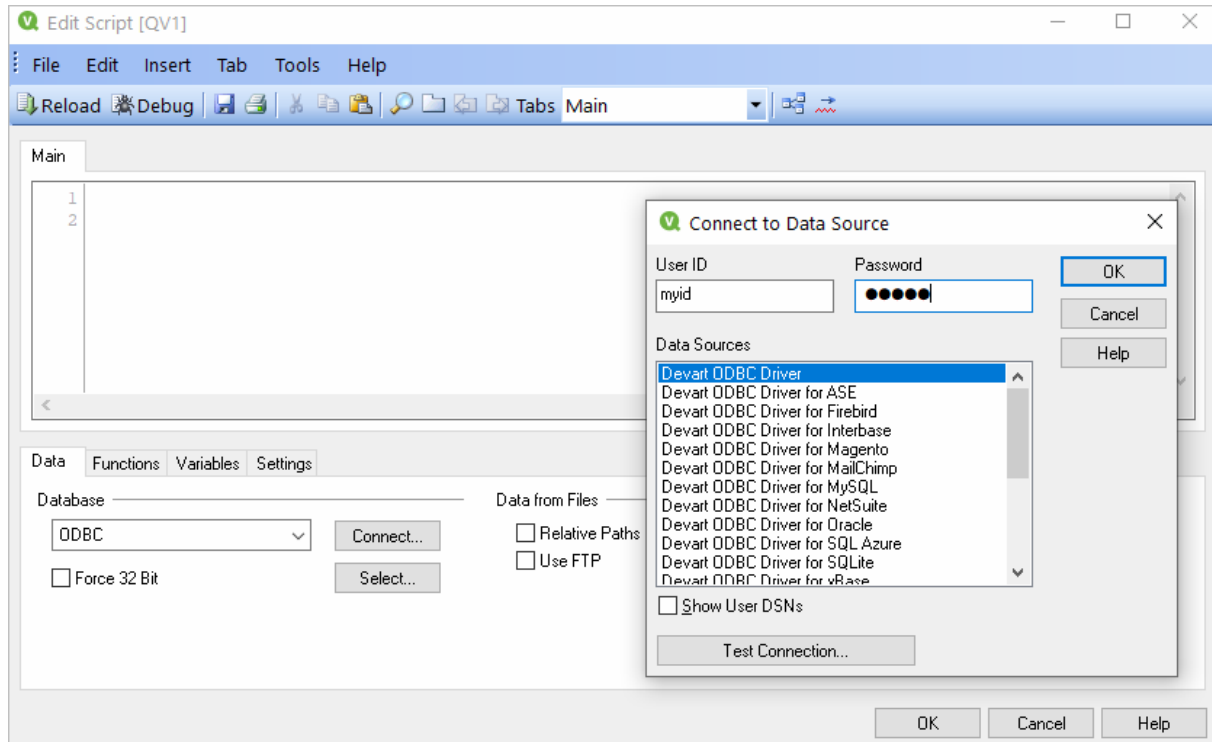
3. To retrieve the data from your data source, you can enter an SQL query and press **F5**. You will be suggested to choose fields to be displayed.
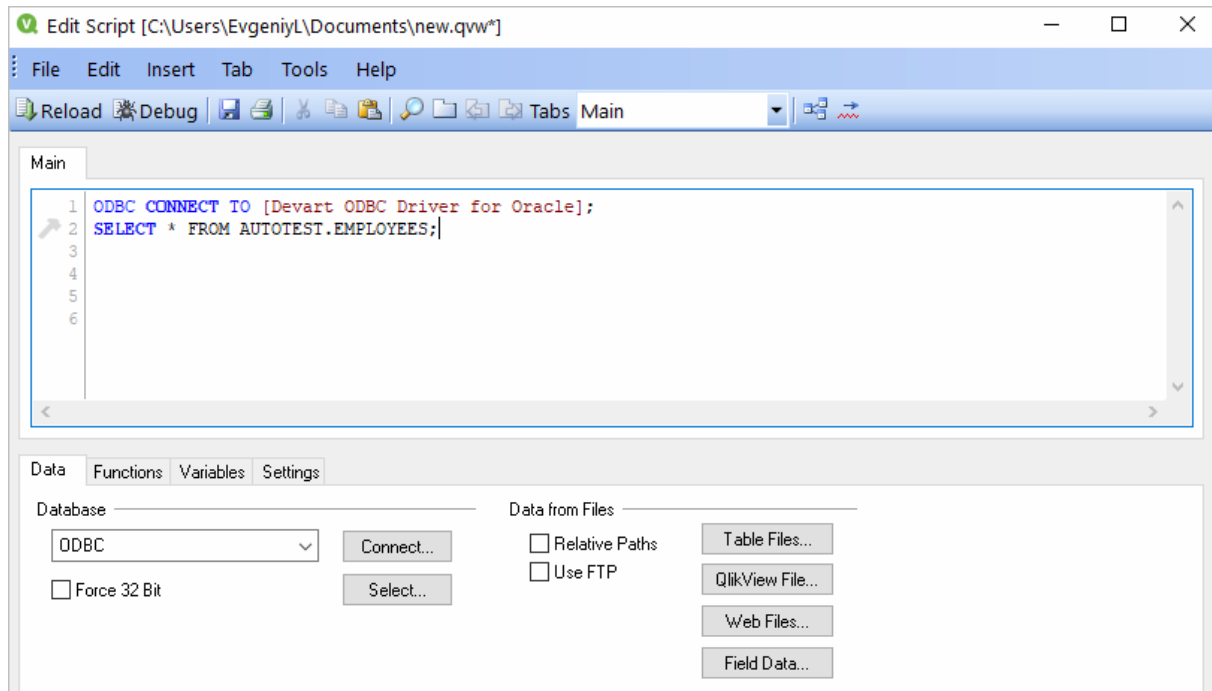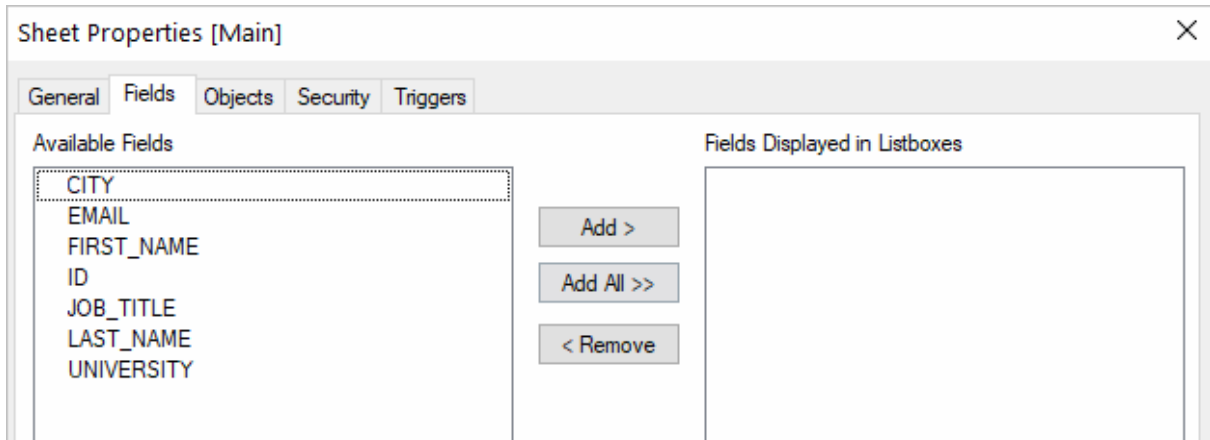
4. Alternatively, you can click **Select**, and QlikView will show you the database structure window where you can compose a SELECT statement for the data to be fetched. You can choose a different database from the database drop-down list. Select the necessary tables and fields. You can retrieve date from multiple tables and fields by selecting them and clicking **Add**. When you are ready with your SELECT statement, click **OK**. You will get back to the main script editor with your SQL statement. Press **F5** to execute the script and select the fields to be displayed in QlikView.
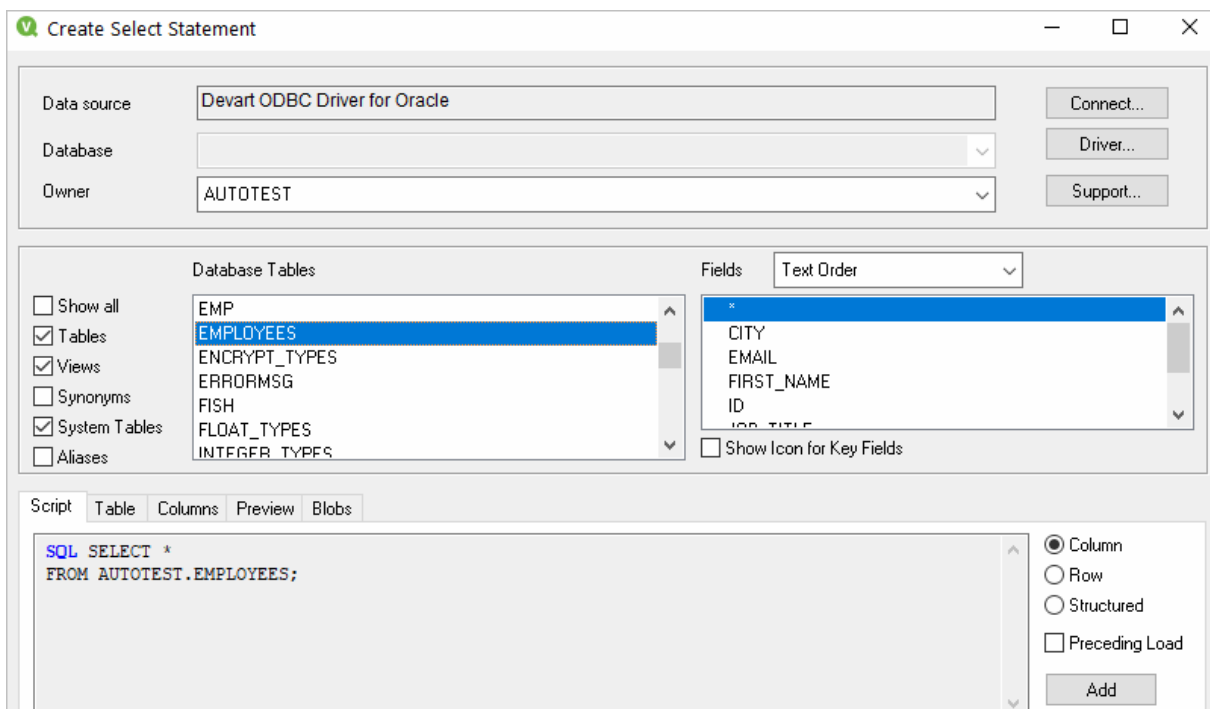
5. Once the data has been fetched, you can choose a table layout to present the data in a

table. Choose **Layout > New Sheet Object > Table Box**. Select the fields to be added to the tablebox and click **OK**.

Request Support   ODBC Forum   Provide Feedback

4.12   Using in SSIS

SQL Server Integration Services (SSIS) is a component of SQL Server that is designed to perform various data migration tasks. When using Devart ODBC Driver for Oracle as a translation layer between the data source and SSIS, the driver and SSIS communicate via Microsoft ODBC version 3.x.

Note that when you extract data from an ODBC data source using the `SQLExecDirect` function, an issue may occur: SSIS expects the ODBC 2.x behavior, while the ODBC driver continues to fetch data from a data source via ODBC version 3.x. To prevent any issues when using SQLExecDirect, you should force the ODBC 2.x behavior in the DSN settings: open the `Advanced Settings` tab and select `Ver 2.x` from the `ODBC Behavior` dropdown.
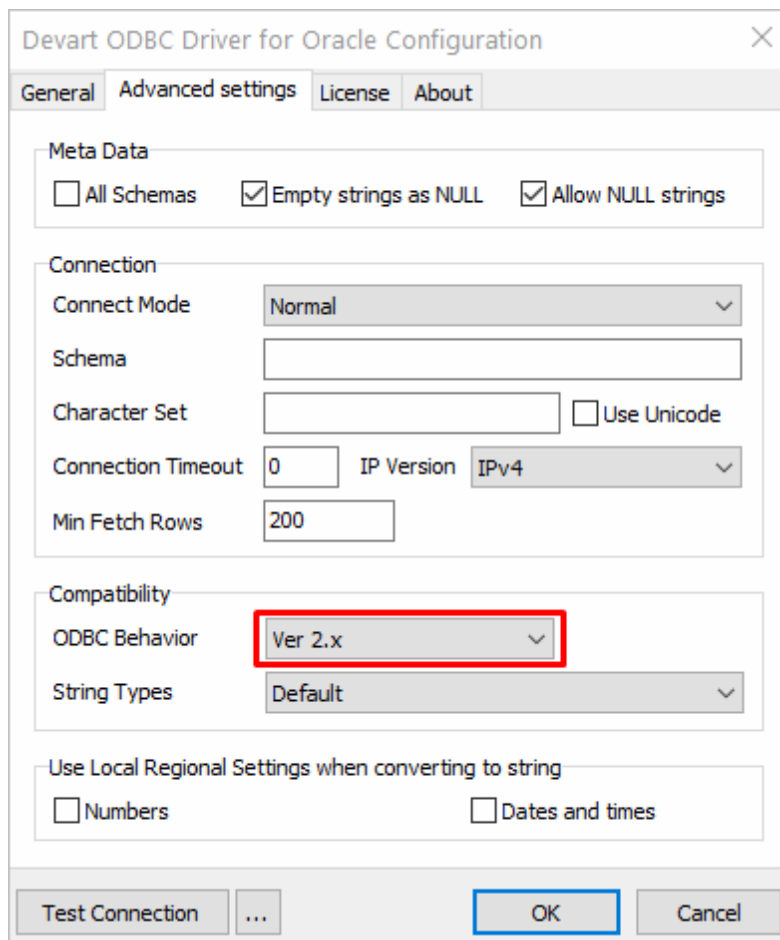
4.13   Using in Tableau

This section describes how to establish and troubleshoot a connection to Oracle from Tableau using ODBC Driver for Oracle.

- Using in Tableau

- Troubleshooting in Tableau

Request Support   ODBC Forum   Provide Feedback

4.13.1   Using in Tableau

Importing Oracle Data Into Tableau Through an ODBC Connection

This article explains to establish and ODBC connection to Oracle from Tableau Desktop. Tableau is a data visualization tool that allows you to pull in raw data, perform analysis on it, and create meaningful reports to get actionable insights. With Tableau Desktop and our suite of ODBC drivers, you can connect to various relational and non-relational databases, both cloud and on-premise.

1. Run Tableau Desktop.

2. On the start page, select **More...** in the **Connect** pane.

3. Choose **Other Databases (ODBC)**.

4. Expand the **DSN** drop-down list and select the DSN that you have created and configured for Oracle. Alternatively, if you have not created a DSN, you can choose the **Driver** option and select Devart ODBC Driver for Oracle from the drop-down.

5. Click **Connect**.

6. After a successful connection, click **Sign in**.

7. Select the needed database and schema in Oracle.

8. You should see the list of all tables you have access to in the connected data source.

9. Drag-and-drop the table name to the area where it says **Drag tables here** to retrieve the data, or click **New Custom SQL** to write a query that will select only specific data from the table.

10. Hit **Update Now** to retrieve and display the data.

Request Support   ODBC Forum   Provide Feedback

Reserved.

4.13.2 Troubleshooting in Tableau

Troubleshooting ODBC Connection in Tableau

The default Tableau settings limit the number of rows to be retrieved through an ODBC connection to 1,000. To overcome this limitation, create a Tableau Datasource Customization (`.tdc`) file in '*Users\\[your name]\\Documents\\My Tableau Repository\\Datasources*' — for example, *devart-oracle.tdc*, and add the following capabilities to the file:

```xml
<?xml version='1.0' encoding='utf-8' ?>
<connection-customization class='genericodbc' enabled='true' version='1.0'>
<vendor name='oracle' />
<driver name='Devart ODBC Driver for Oracle' />
    <customizations>
            <customization name='CAP_QUERY_TOPSTYLE_ROWNUM' value='no' />
            <customization name='CAP_QUERY_TOP_N' value='no' />
            <customization name='CAP_QUERY_TOPSTYLE_LIMIT' value='no' />
            <customization name='CAP_QUERY_TOPSTYLE_TOP' value='yes' />
            <customization name='CAP_SELECT_TOP_INTO' value='no' />
    </customizations>
</connection-customization>
```

Troubleshooting ODBC Connection in Tableau on macOS

The iODBC driver manager incorrectly handles the SQL_WCHAR and SQL_WVARCHAR ODBC data types. To work with these data types in Tableau, define an additional capability:

```xml
<customization name='CAP_ODBC_BIND_SUPPRESS_WIDE_CHAR' value='yes' />
```

Request Support   ODBC Forum   Provide Feedback