Table of Contents

1 What's New

New features in ODBC Driver for PostgreSQL 5.0

- Added a graphical interface for configuring the driver on macOS and Linux
- Added support for PostgreSQL 18
- Added support for the Bearer Token authentication when using an HTTP tunnel
- Improved performance of opening connections
- Improved compatibility with Node.js
- Improved compatibility with Tableau
- Improved compatibility with Vectorworks

New features in ODBC Driver for PostgreSQL 4.7

- Added support for password-protected private keys for the SSL protocol
- Improved compatibility with Microsoft Query

New features in ODBC Driver for PostgreSQL 4.6

- Added support for PostgreSQL 17
- Added support for the pgvector extension
- Added Read Only mode for connection

New features in ODBC Driver for PostgreSQL 4.5

- Fixed connection timeout setting before opening connection
- Improved compatibility with FileMaker Server for Ubuntu
- Improved compatibility with FileMaker Server for MacOS
- Added support for PostGIS
- Added support for SQL_ATTR_MAX_ROWS attribute
- Improved compatibility with Visual Basic in Visual Studio
- Improved compatibility with 4D in macOS

New features in ODBC Driver for PostgreSQL 4.3

- Added support for PostgreSQL 15
- Added support for macOS 13 Ventura
- Added support for the SCRAM-SHA-256-PLUS authentication mechanism
- Improved compatibility with Tableau Prep Builder
- Improved compatibility with Crystal Reports
- Improved the SSH connection establishment

New features in ODBC Driver for PostgreSQL 4.2

- Added support for Windows 11
- Added support for PostgreSQL 14
- Added support for OUT parameters in stored procedures in PostgreSQL 14
- Improved compatibility with PostgreSQL 8.1 and 8.2
- Improved compatibility with FICO Mosel
- Improved compatibility with FileMaker
- Improved compatibility with JMP on macOS
- Improved support for an ODBC installer on Windows 2000

New features in ODBC Driver for PostgreSQL 4.1

- MSI installer for deploying through GPO is added

New features in ODBC Driver for PostgreSQL 4.0

- Apple Silicon M1 is supported
- Compatibility with macOS Big Sur is improved

New features in ODBC Driver for PostgreSQL 3.4

- PostgreSQL 13 is supported
- Compatibility with Clarion Enterprise is improved

New features in ODBC Driver for PostgreSQL 3.3

- Materialized views are supported
- Compatibility with Power Pivot is improved
- Compatibility with Visual Basic is improved
- Compatibility with RStudio is improved

New features in ODBC Driver for PostgreSQL 3.2

- PostgreSQL 12 is supported

New features in ODBC Driver for PostgreSQL 3.1

- Amazon RDS for PostgreSQL is supported
- Work with password-protected private key for SSH protocol is supported
- Now ODBC driver activation does not require administrator privileges
- Improved compatibility with sandboxed applications for macOS
- Support for connection pooling is improved

New features in ODBC Driver for PostgreSQL 3.0

- PostgreSQL 11 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

New features in ODBC Driver for PostgreSQL 2.3

- Performance of batch operations is significantly improved
- The SSHStoragePath connection parameter is added
- Support for the use of "?", "?|" and "?&" JSON operators in SQL statements by escaping the question mark character ("\?", "\?|" and "\?&" respectively) is added

New features in ODBC Driver for PostgreSQL 2.2

- PostgreSQL 10 is supported
- Azure Database for PostgreSQL is supported
- 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

New features in ODBC Driver for PostgreSQL 2.1

- 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
- The SELECT ... FOR UPDATE statement is supported

New features in ODBC Driver for PostgreSQL 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
- Support for server versions 7.3-8.3 is improved
- Bug with fetching big data amount is fixed

New features in ODBC Driver for PostgreSQL 1.3

- Connection via SSL protocol is supported
- Connection via SSH protocol is supported
- Connection via HTTP tunnel is supported
- Compatibility with Power BI Desktop is improved
- Compatibility with Microsoft Visual FoxPro is improved

New features in ODBC Driver for PostgreSQL 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

New features in ODBC Driver for PostgreSQL 1.1

- Performance is improved

New features in ODBC Driver for PostgreSQL 1.0

- First release of ODBC Driver for PostgreSQL
- Windows 32-bit is supported
- Windows 64-bit is supported

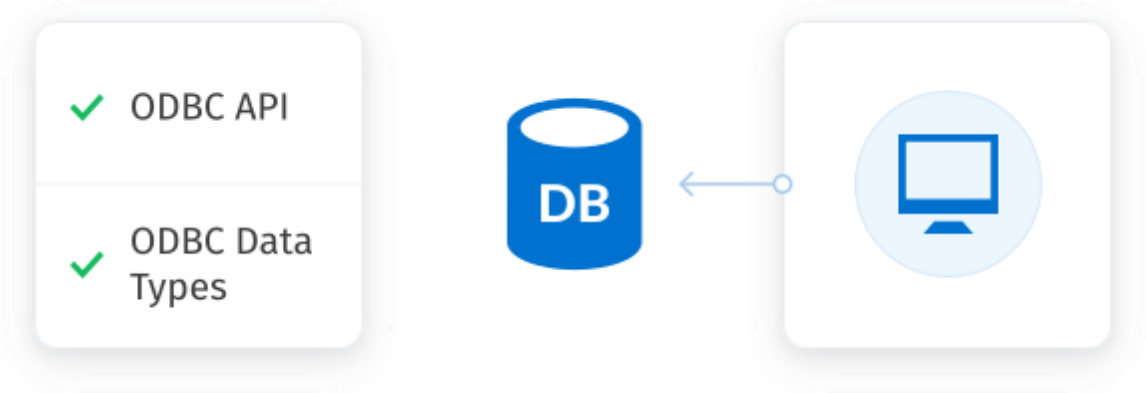
2 General Information

6. Getting Support

2.1 Overview

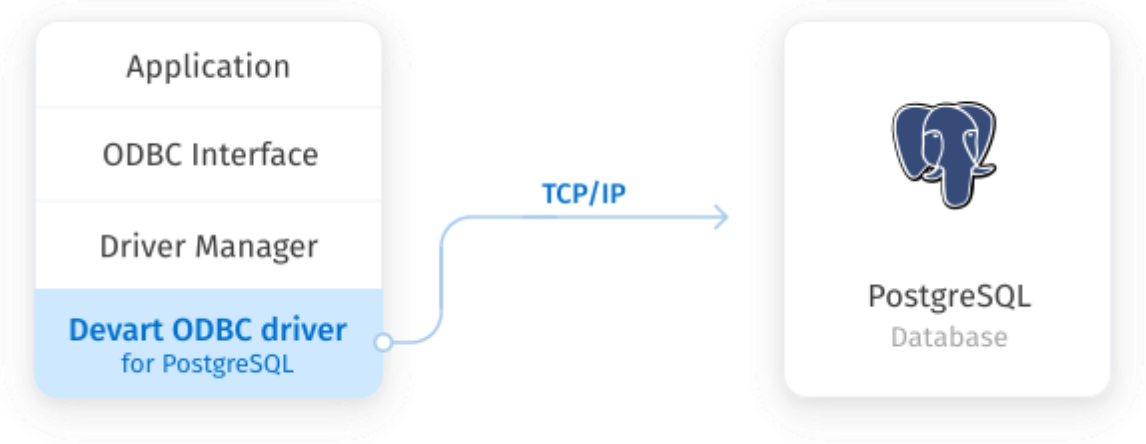
Overview

ODBC Driver for PostgreSQL is a high-performance connectivity solution with enterprise-level features for accessing PostgreSQL databases from ODBC-compliant BI, ETL, reporting, and analytics 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 PostgreSQL data from anywhere.



Direct Connection

Our data connector enables various ODBC-aware applications to establish a direct connection to PostgreSQL via TCP/IP to eliminate the need for PostgreSQL libraries or clients. A direct connection increases the speed of data transmission between an external application and PostgreSQL for real-time analytics. It also streamlines the deployment process, since there is no need to distribute any additional libraries or client tools with the driver.

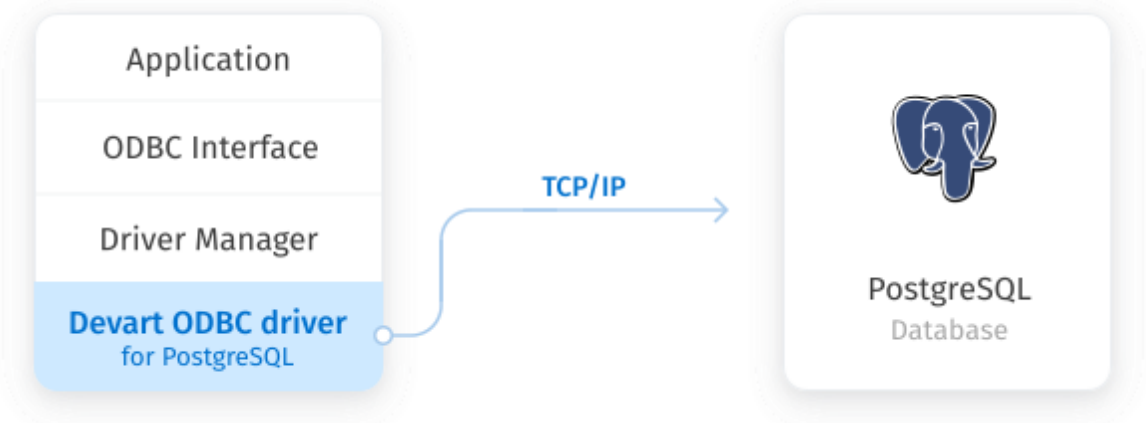


Compatibility

The following versions of PostgreSQL are supported:

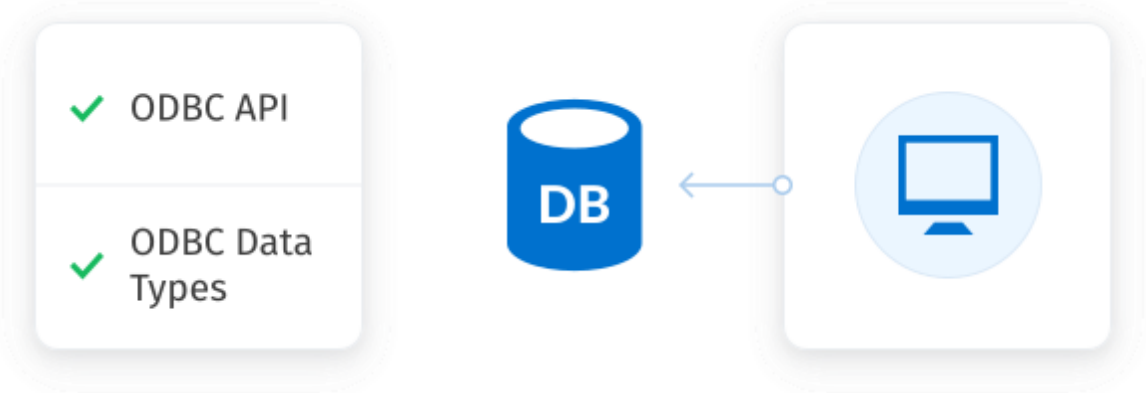
- PostgreSQL server from 8.0 to 17
- Microsoft Azure Database for PostgreSQL
- Amazon RDS for PostgreSQL
- Amazon Aurora
- Google Cloud for PostgreSQL

2.2 Features



Direct Connection

Database applications based on our solution get an opportunity to establish connection to PostgreSQL in directly. Direct Connection gives your applications an unrivaled advantage - connection to PostgreSQL databases directly via TCP/IP . 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.

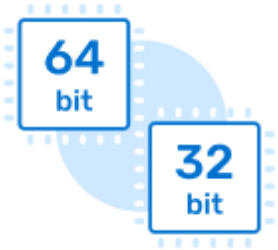


ODBC Conformance

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 PostgreSQL from various environments and platforms, that support ODBC.



Development Platforms Variety

ODBC Driver for PostgreSQL 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 PostgreSQL supports PostgreSQL server versions since 8.0 up to 14.0.

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.

2.3 Compatibility

PostgreSQL Compatibility

ODBC Driver for PostgreSQL supports PostgreSQL server versions from 8.0 to 17.

Microsoft Azure Database for PostgreSQL Compatibility

ODBC Driver for PostgreSQL supports Microsoft Azure Database for PostgreSQL.

Amazon RDS for PostgreSQL Compatibility

ODBC Driver for PostgreSQL supports Amazon RDS for PostgreSQL and Amazon Aurora.

Google Cloud for PostgreSQL Compatibility

ODBC Driver for PostgreSQL supports Google Cloud for PostgreSQL.

AlloyDB Compatibility

ODBC Driver for PostgreSQL supports AlloyDB in Google Cloud.

Heroku Postgres Compatibility

ODBC Driver for PostgreSQL supports Heroku Postgres.

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	✓

2.4 Requirements

The following requirement must be met for ODBC Driver for PostgreSQL:

- Only one version of ODBC Driver for PostgreSQL is installed on your system.

No additional client software is required on your system.

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 Software Edition specified in the registration letter you have received from Devart upon purchase of the Software:

- the "Desktop Edition" 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 Edition" 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, depending on the License Type specified in the registration letter you have received from Devart upon purchase of the Software:

- the "Subscription-based License" allows you to install and use the Software on a single computer only during the subscription term specified at purchase. An Internet connection is required to activate the license and check the license status when the Software is used. Once the subscription term is over, you will be able to either stop using the Software or renew the license for a new subscription term;

- the "Perpetual License" allows you to install and use the specific Software product version on a single computer without an active subscription. A subscription provides access to new product releases, regular upgrades, and support for new server versions provided during the subscription term;

- the "Site License" allows you to install and use the Software on one or more computers in a single company in accordance with this Agreement;

- the "OEM License" allows you to install and use the Software on one or more computers in a single company as well as deploy the Software as part of a licensee's application to web servers, application servers, batch servers, desktops, and other end-user devices. This definition includes the ability to install and use the application containing the Software without

any additional fees in favor of the licensor.

1.3. 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.

1.4. If you have the "OEM License", you are also entitled to:

- make any number of copies of the Software to deploy it to your end-user.

- deploy the Software to your end-user as a Software installation package or integrate it into your Applications.

1.5. 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 unless you are an "OEM License" holder. 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" on 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 Database 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.

2.6 Getting Support

This document lists several ways you can find help with using ODBC Driver for PostgreSQL describes the Priority Support program.

Support Options

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

- You can find out more about ODBC Driver for PostgreSQL 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 PostgreSQL developers through the ODBC Driver for PostgreSQL Priority Support program.

Subscriptions

The ODBC Driver for PostgreSQL Subscription program is an annual maintenance and

support service for ODBC Driver for PostgreSQL users.

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

- Product support through the ODBC Driver for PostgreSQL Priority Support program
- Access to new versions of ODBC Driver for PostgreSQL when they are released
- Access to all ODBC Driver for PostgreSQL updates and bug fixes
- Notifications about new product versions

Priority Support

ODBC Driver for PostgreSQL Priority Support is an advanced product support service for getting expedited individual assistance with ODBC Driver for PostgreSQL-related questions from the ODBC Driver for PostgreSQL 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 PostgreSQL Subscription.

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

Your ODBC Driver for PostgreSQL Registration number.

- Full ODBC Driver for PostgreSQL edition name and version number. You can find the version number in DLL version information.
- Versions of the PostgreSQL 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 PostgreSQL.

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

3 Using ODBC Driver

1. Installation

2. Product Activation
3. Connecting to PostgreSQL
4. Connection String Parameters
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

3.1 Installation

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

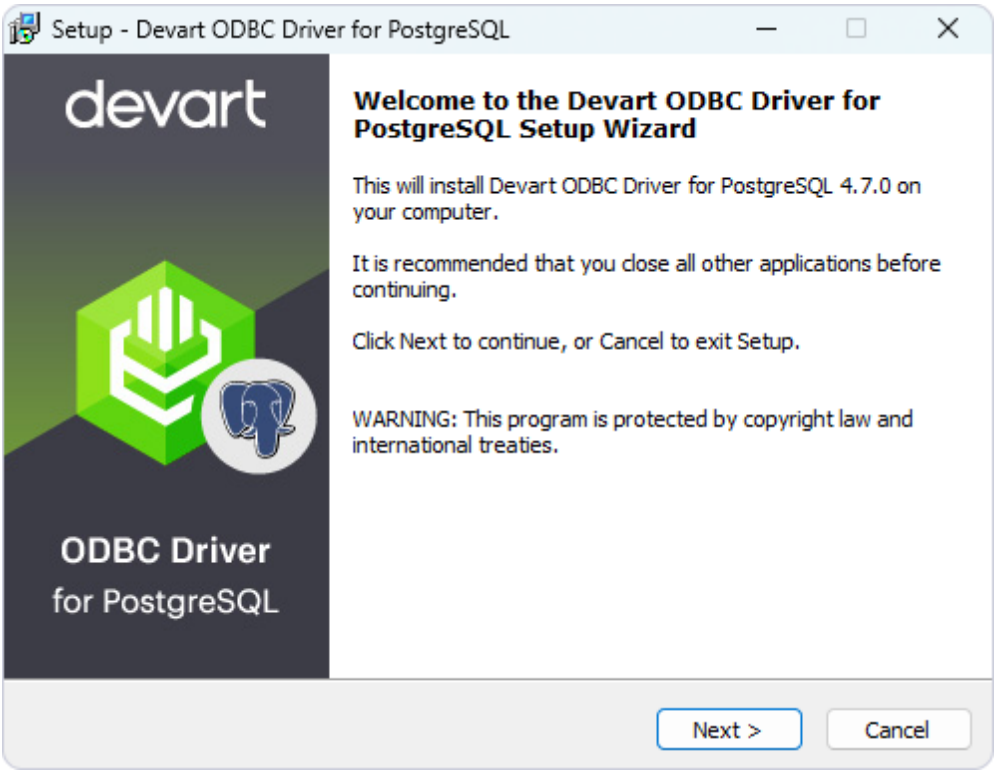
See how to install Devart ODBC Driver for PostgreSQL:

- Windows
- Windows Silent
- macOS
- Linux DEB
- Linux RPM

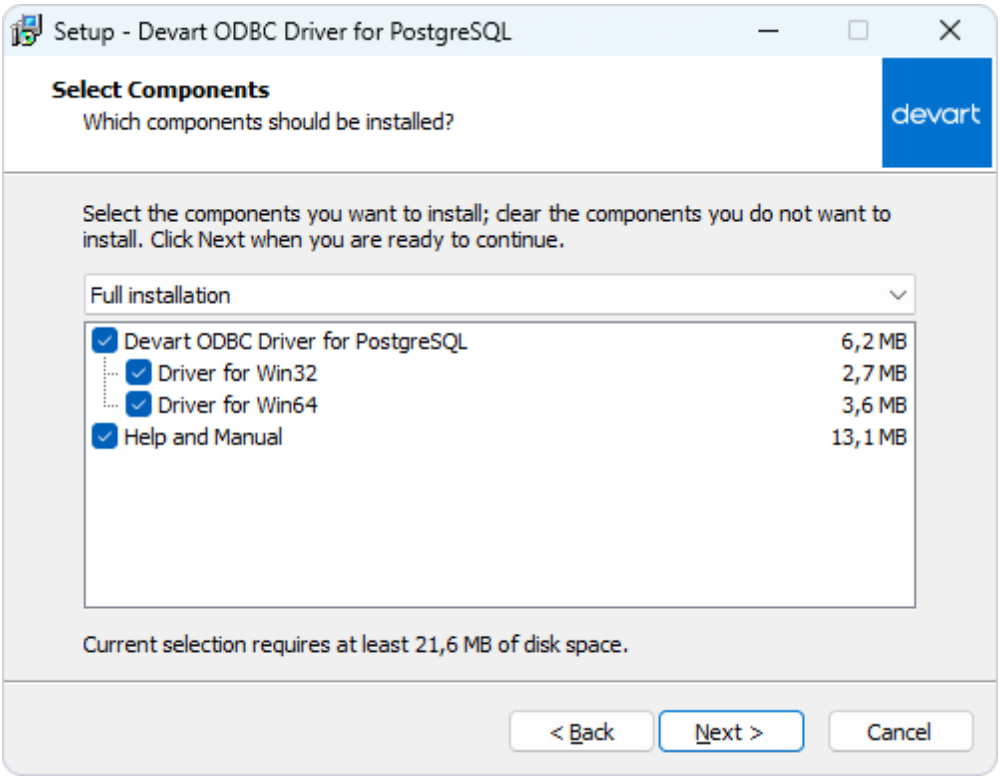
3.1.1 Windows

Installation

1. Download and run the installer.
2. Follow the instructions in the wizard.



3. 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 choose whether to install the **64-bit** version of the driver. Clear the checkbox if you do not need a 64-bit installation. There is also a checkbox on this page that allows you to choose whether to install Help and Manual.



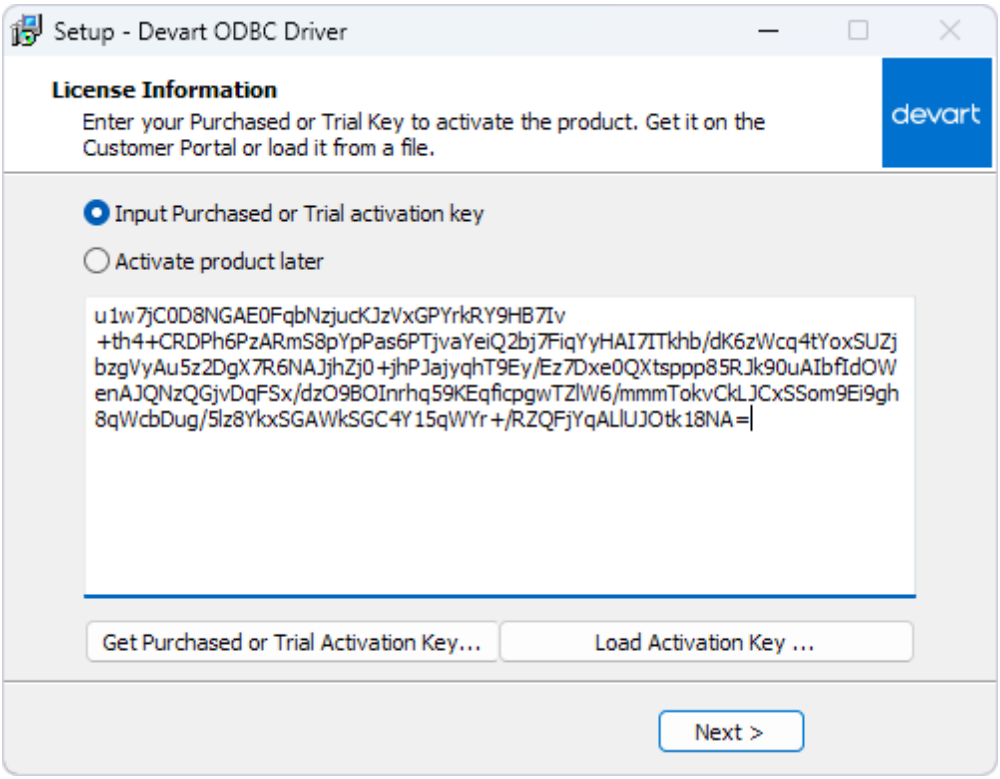
5. On the **License Information** page, select when you want to activate the driver:

- **Immediately after installation:** Select **Input Purchased or Trial activation key** and enter your key in the provided box, or click **Load Activation Key** and select the file containing your key.
- **Any other time:** Select **Activate product later**.

You need to activate the driver even for the trial version.

You can find your activation key in the registration email or your Customer Portal account.

To open the Customer Portal, click **Get Purchased or Trial Activation Key**.



6. Click **Next** to complete the installation.

7. Click **Finish** to exit Setup.

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

See also:

- Installation on macOS
- Install Linux DEB package
- Install Linux RPM package

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:

```
DevartODBCPostgreSQL.exe /SILENT /ActivationKey=y1c7nmgdu234laszxcvONGurjfhx
```

```
DevartODBCPostgreSQL.exe /VERYSILENT /ActivationKey=ekhdh765mh09ukr237gfHRtr
```

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

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

```
DevartODBCPostgreSQL.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.

3.1.3 macOS

Prerequisites

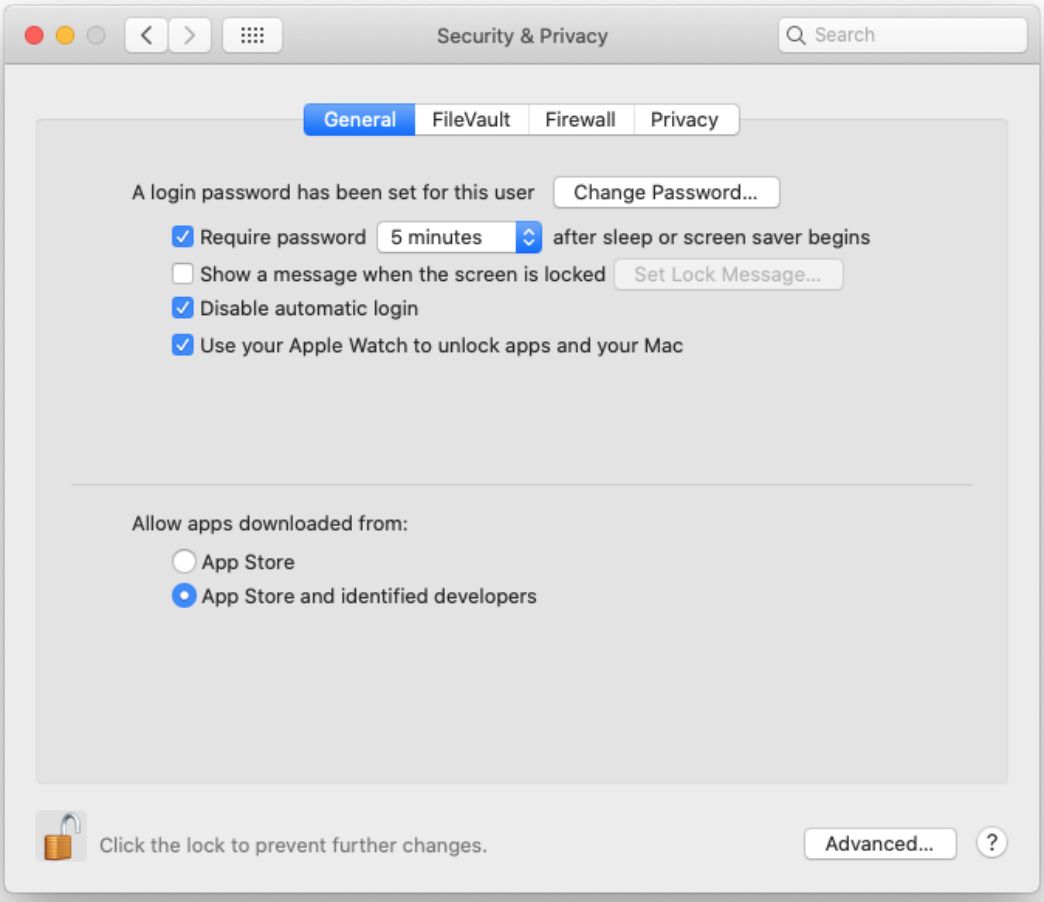
ODBC Driver for PostgreSQL works under the control of an ODBC driver manager. ODBC driver manager is not distributed along with our driver and must be installed separately.

ODBC Driver for PostgreSQL is compatible with iODBC driver manager.

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

Installing ODBC Driver for PostgreSQL

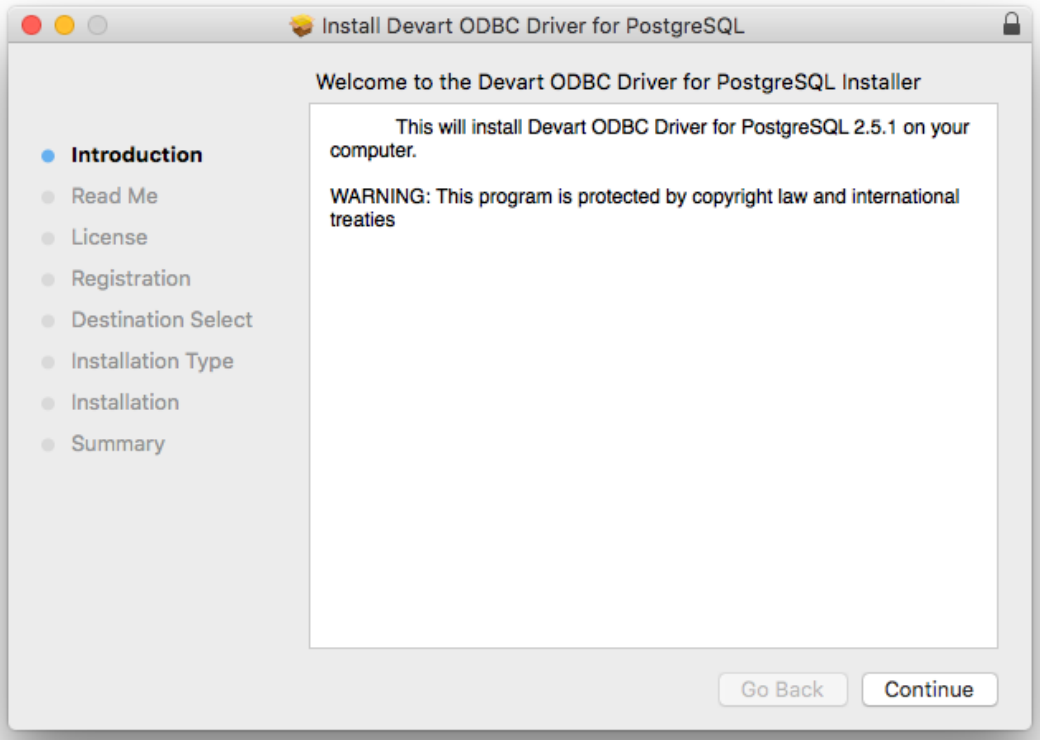
1. Go to Security & Privacy settings in the System Preferences.
2. Enable the *App Store and identified developers* option in the **Allows apps downloaded from** section.



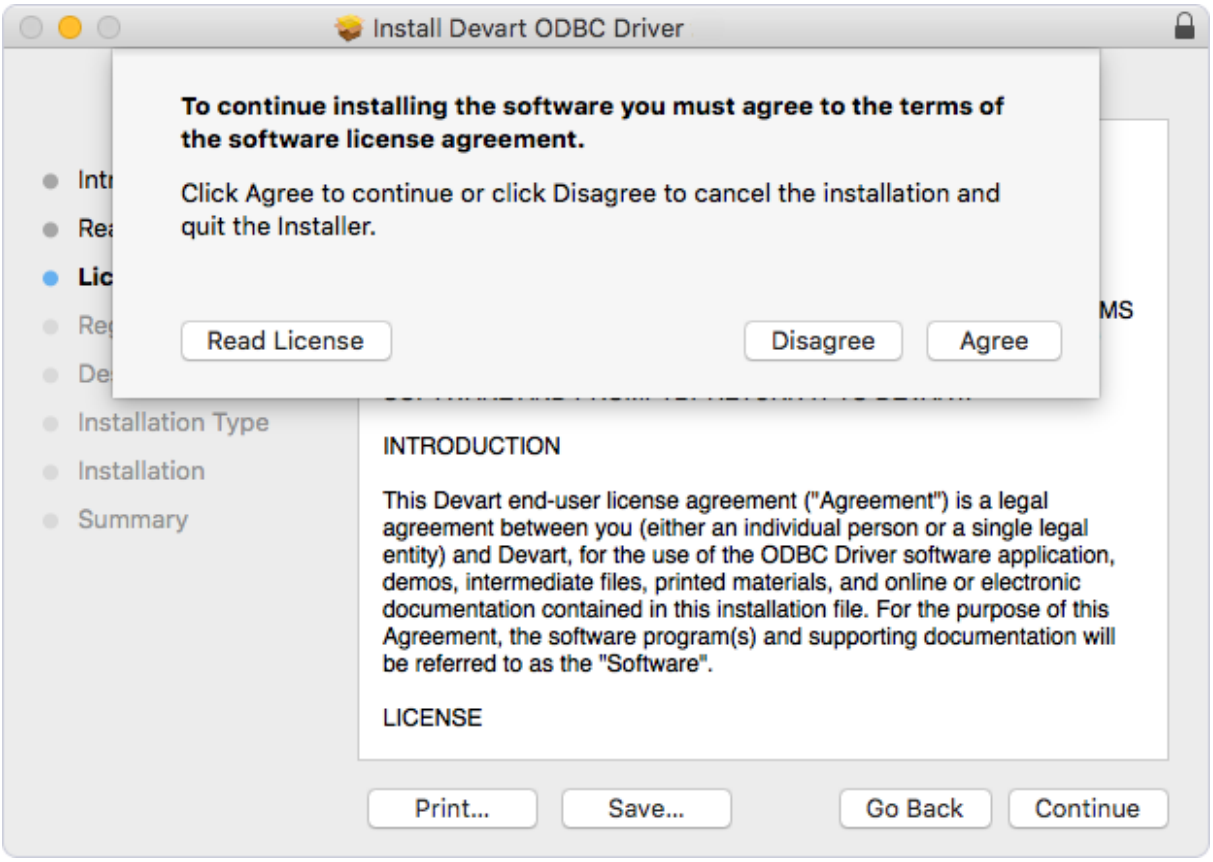
If the options in **Allow apps downloaded from** section are grayed out, click the lock icon and enter your administrator password to proceed with the installation.

3. Download the PKG file from the Devart website.

4. Run the downloaded file, click **Allow** to proceed with the installation.



5. After reading the license agreement, click **Agree**.

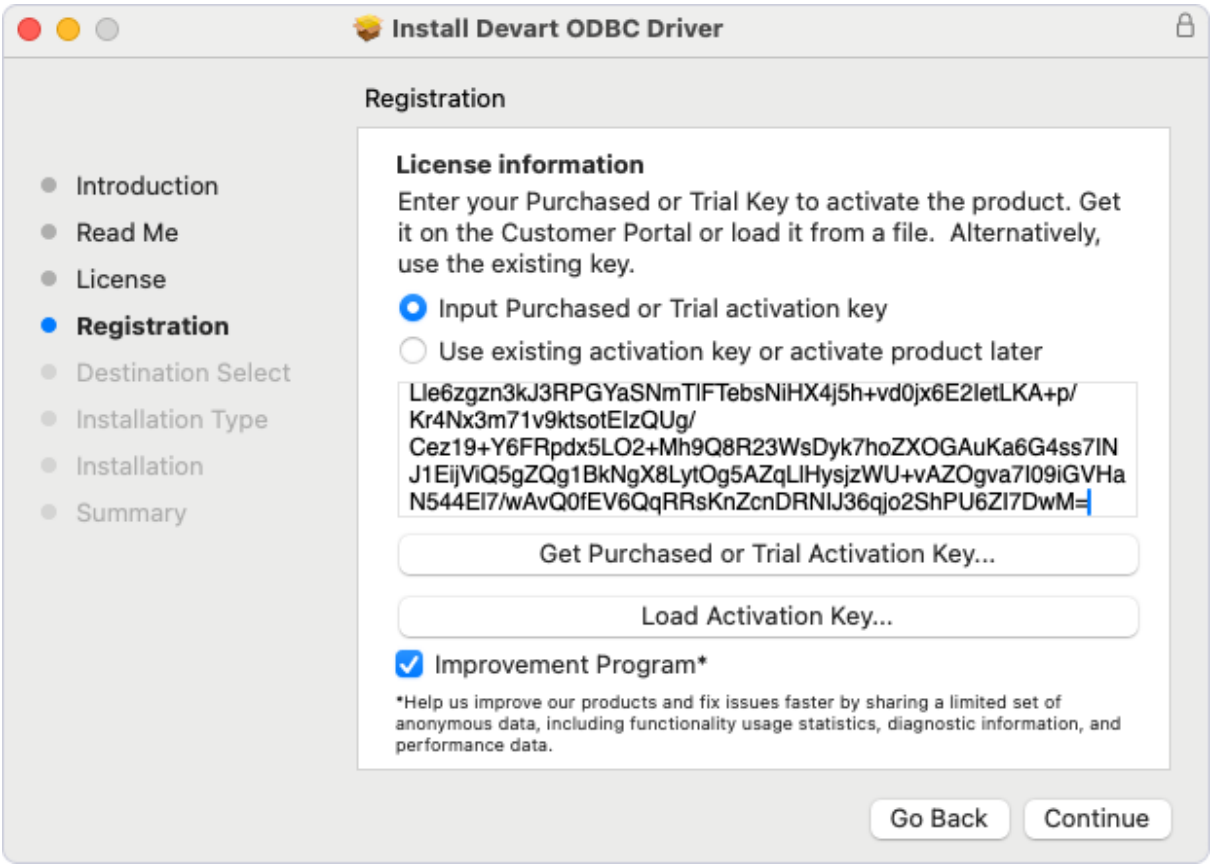


6. On the **Registration** page, specify your activation key using one of the following methods:

- Enter an activation key:

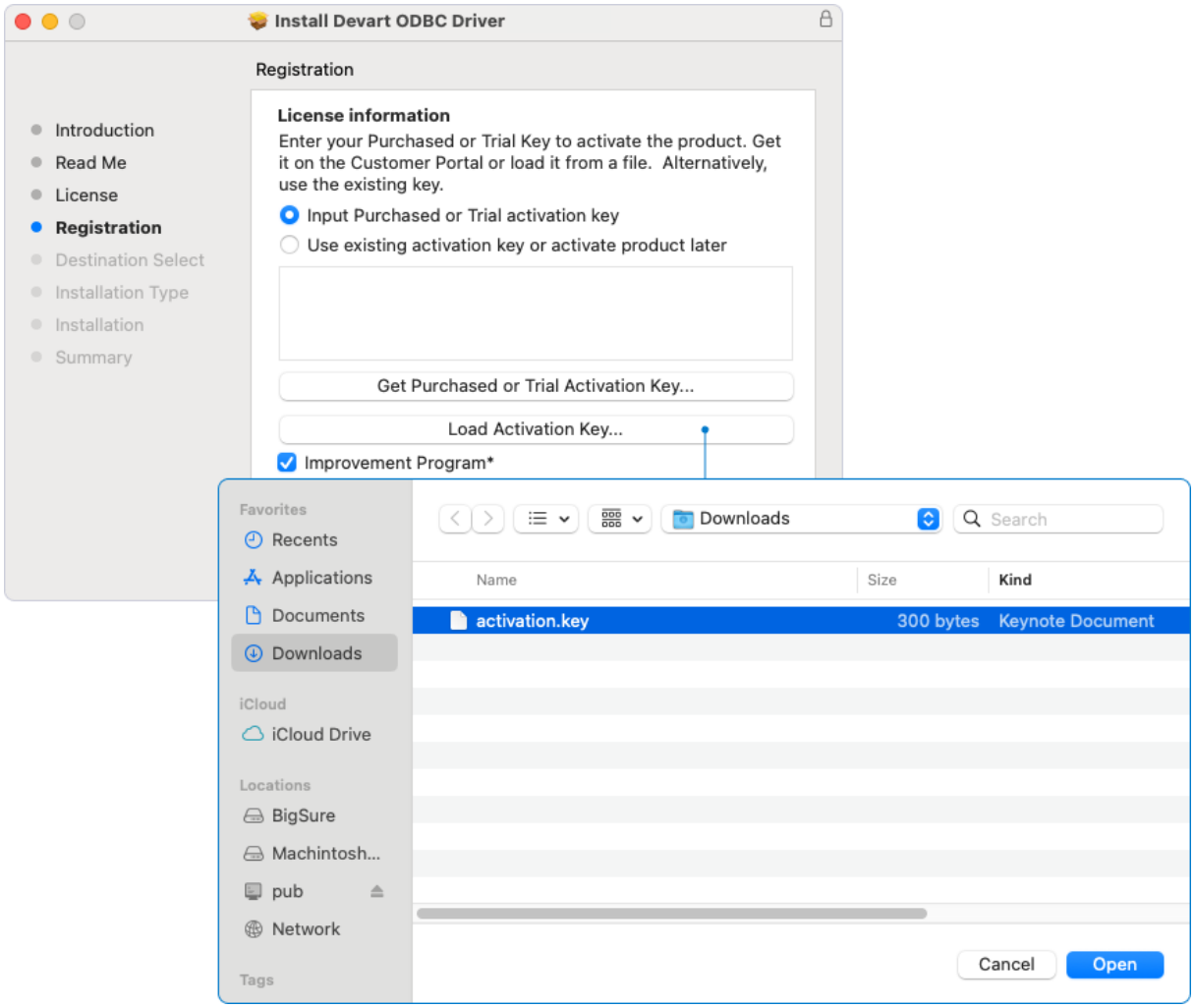
  1. Select **Input Purchased or Trial activation key**.

  2. Enter your activation key.



- Load an activation key file:

  1. Click **Load Activation Key**.
  2. Navigate to the location of the activation file.
  3. Click **Open**.

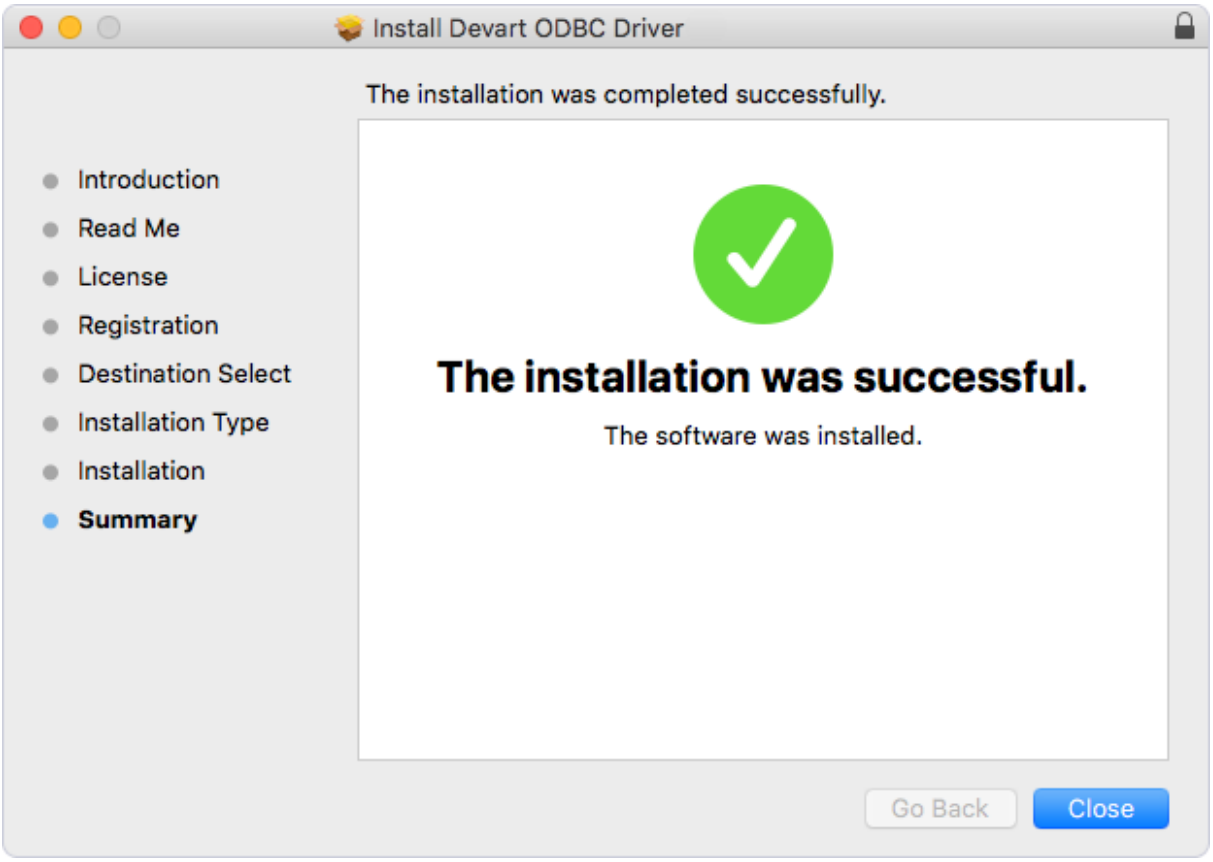


You need to activate the driver even for the trial version.

You can find your activation key in the registration email or your Customer Portal account. To open the Customer Portal, click **Get Purchased or Trial Activation Key**.

If you're reinstalling the driver or don't want to activate the driver right now, select **Use existing activation key or activate product later**.

7. To complete the installation click **Continue**, then click **Install**.



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

See also:

- Installation on Windows
- Install Linux DEB package
- Install Linux RPM package

3.1.4 Linux DEB

Prerequisites

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

ODBC Driver for PostgreSQL is compatible with unixODBC driver manager. Depending on

your Linux distribution, you can install the unixODBC driver manager using one of the following commands:

- For Ubuntu 23 and later versions:

```
sudo apt-get install libodbcinst2 libodbc2 odbcinst unixodbc
```

- For other distributions, including Ubuntu 22 and earlier versions:

```
sudo apt-get install odbcinst1debian2 libodbc1 odbcinst unixodbc
```

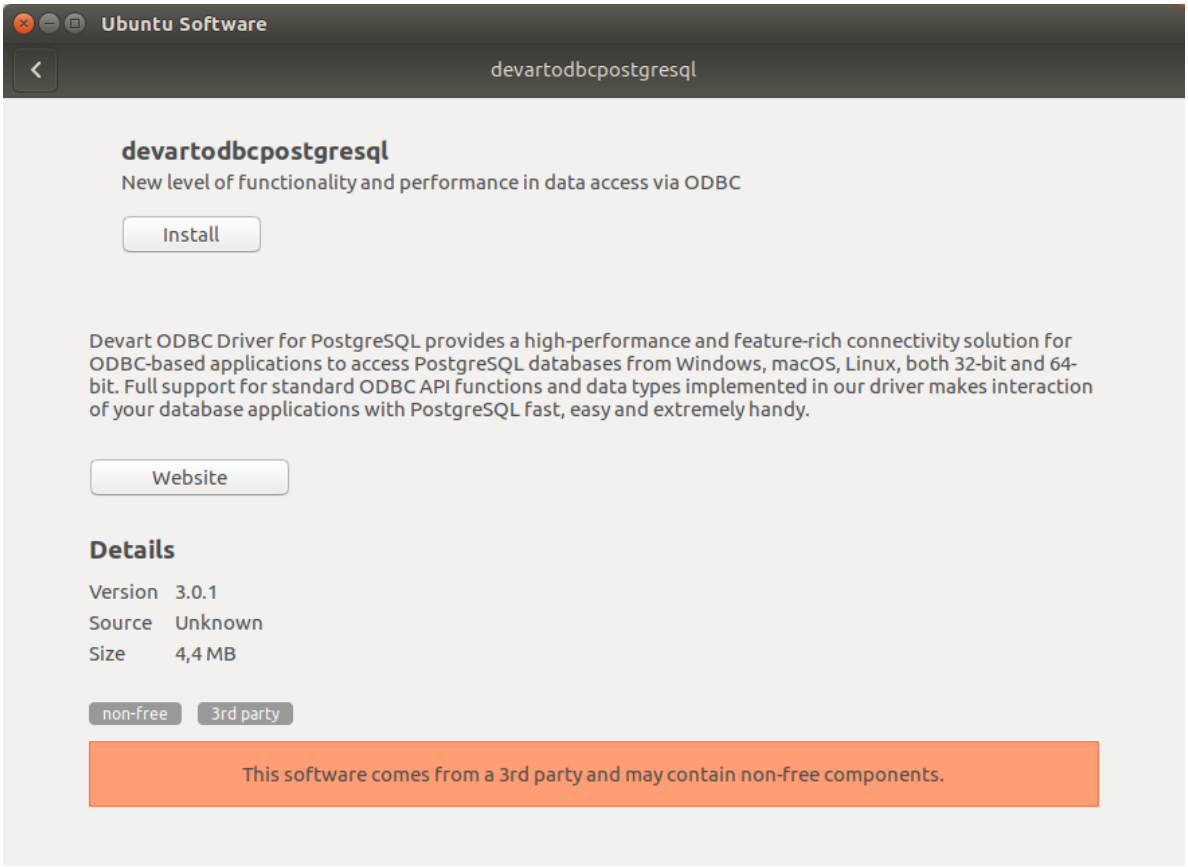
If you are using other ODBC driver managers, ODBC Driver for PostgreSQL will be installed, but it will require manual modification of the configuration files of these managers.

Installation

Let's consider how to install Devart ODBC Driver on Linux from a DEB package, for example, on Ubuntu. There are two ways to install the driver: manually using the GUI 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 **Install**.



If the installation is successfully completed, the **Install** button changes to **Remove**.

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

You need to activate the driver even for the trial version.

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).

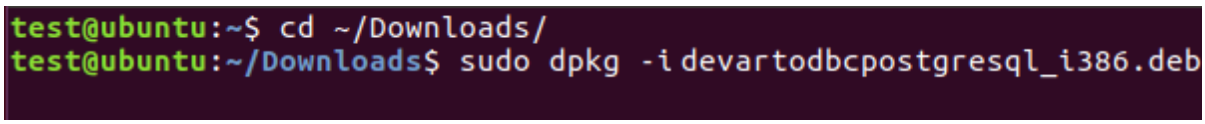
2. Run the 'Terminal' program.

3. Navigate to the folder with the downloaded package (if you downloaded the package to a folder other than ~/Downloads, specify the path to that folder as the cd command parameter):

```
cd ~/Downloads/
```

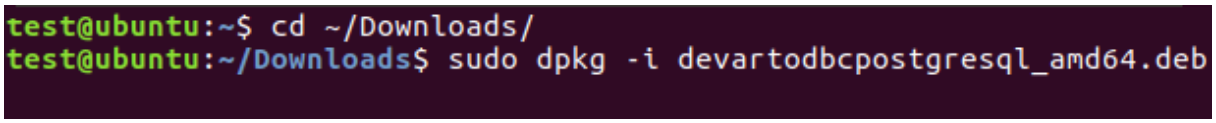
4. To install the devartodbcpostgresql_i386.deb on a 32-bit system, use the following command:

```
sudo dpkg -i devartodbcpostgresql_i386.deb
```

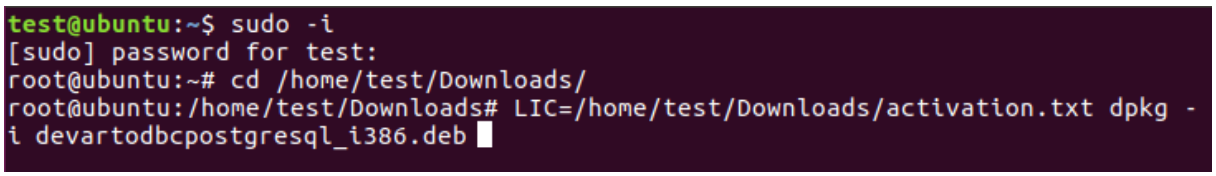


5. To install the devartodbcpostgresql_amd64.deb on a 64-bit system, use the following command:

```
sudo dpkg -i devartodbcpostgresql_amd64.deb
```



The driver is installed successfully.



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

You need to activate the driver even for the trial version.

See also:

- Install Linux RPM package
- Installation on Windows

- Installation on macOS

3.1.5 Linux RPM

Prerequisites

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

ODBC Driver for PostgreSQL is compatible with unixODBC driver manager.

If you are using other ODBC driver managers, ODBC Driver for PostgreSQL will be installed, but it will require manual modification of the configuration files of these managers.

Installation

Let's consider how to install Devart ODBC Driver on Linux from an RPM package, for example, on CentOS. To install the driver, 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 to the ~/Downloads folder (or the selected one).

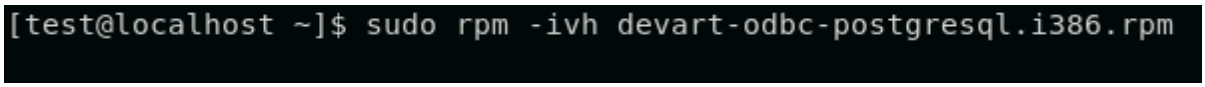
2. Run the 'Konsole' program.

3. Navigate to the folder with the downloaded RPM package (if you downloaded the package to a folder other than ~/Downloads, you need to specify the path to that folder as the cd command parameter):

```
cd ~/Downloads/
```

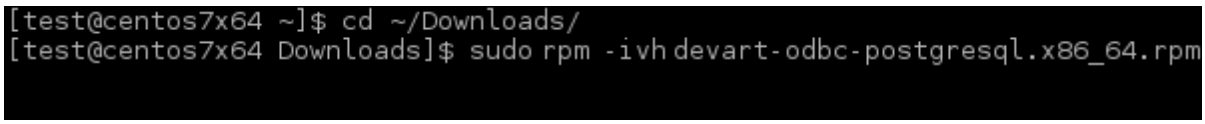
4. To install the devart-odbc-postgresql.i386.rpm on a 32-bit system, use the following command:

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



To install the devart-odbc-postgresql.x86_64.rpm on a 64-bit system, use the following command:

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



The driver is installed successfully.

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

You need to activate the driver even for the trial version.

See also:

- Install Linux DEB package
- Installation on Windows
- Installation on macOS

3.2 Remote Installation

One of the key advantages of Group Policy is the ability to deploy software remotely using MSI files. This section explains how to use Group Policy to remotely install the ODBC Driver for PostgreSQL on client computers.

The information is organized into the following sections:

- Creating the MST File Using Orca
- Remote Deployment and Activation
- Upgrading Driver Version and License Key

3.2.1 Package Transformation

Creating the MST File Using Orca

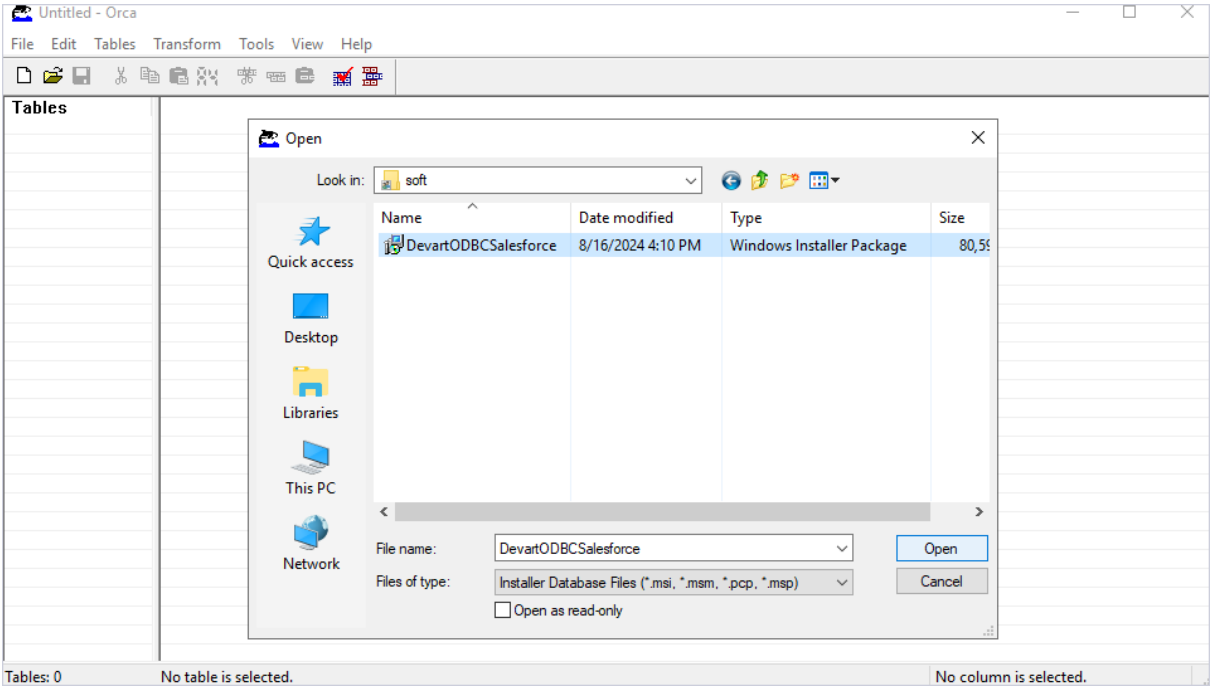
To customize the installation of the ODBC Driver for PostgreSQL, you first need to edit the Windows Installer Package (MSI) by creating an MST file. This will allow for customized installation of an original Windows Installer (MSI) Package.

An MST file, or Windows Installer Setup Transform file, contains program configuration settings. In our case, the MST file for the ODBC Driver for PostgreSQL will include the correct license information. This MST file is used together with the original MSI package in the Group Policy software distribution system.

There are many tools available for customizing MSI file settings, so you can choose the one that best suits your needs. In this example, we'll be using **Orca**, which is available as part of the Windows SDK Components for Windows Installer Developers. For more information about Orca, visit the official Microsoft website.

To start the process of MST file generation using the Orca editor, follow the steps below:

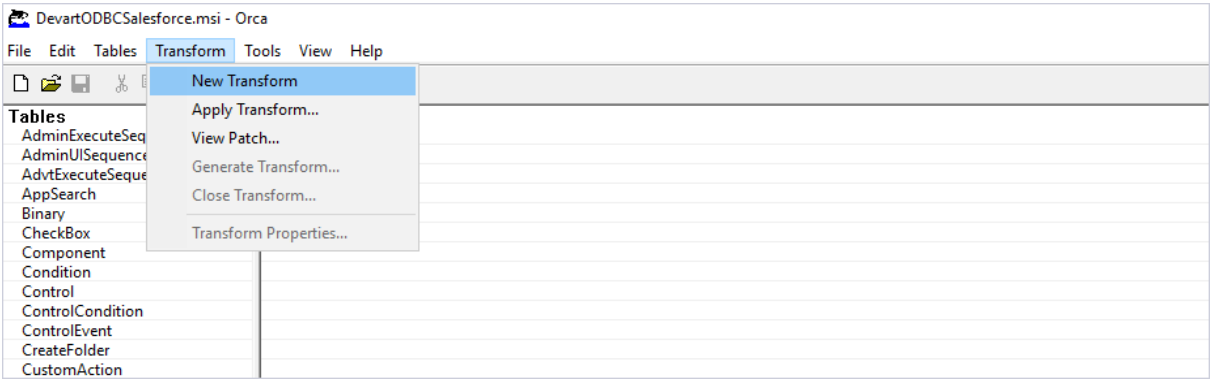
1. Launch the Orca application, then open the required MSI file by selecting **Open** in the **File** menu or click the **Open** icon on the toolbar below.



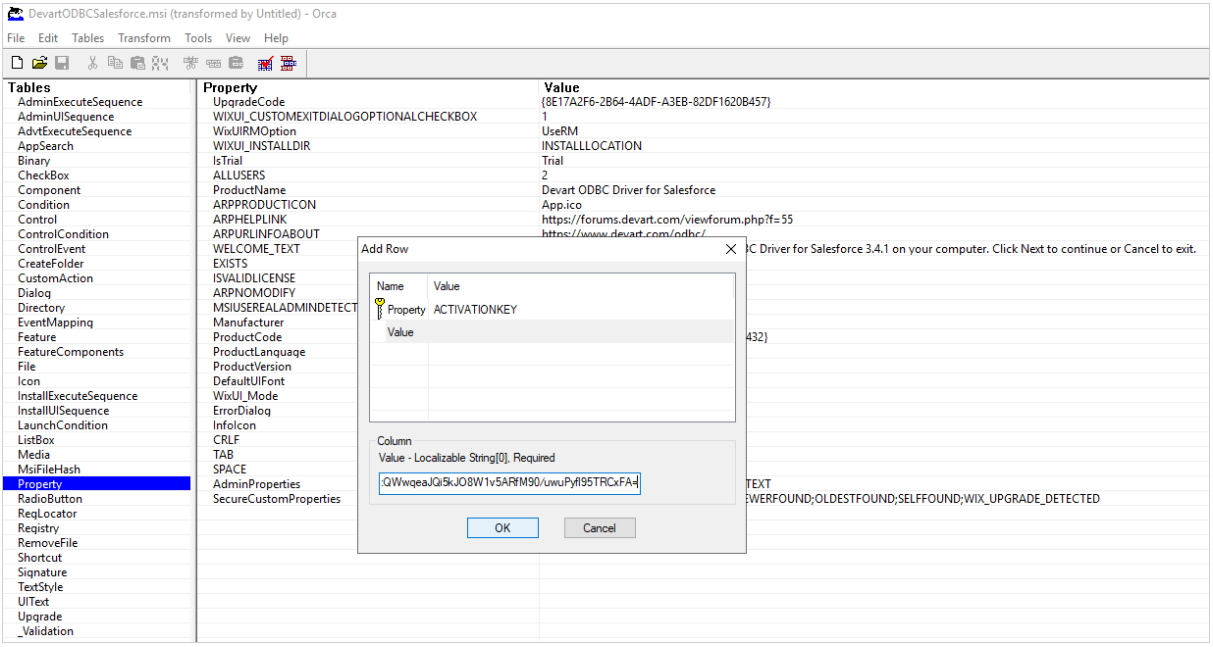
The MSI file for the ODBC Driver for Salesforce is taken as an example to illustrate the Group Policy installation process. Use the same steps described in this section when installing the ODBC Driver for PostgreSQL.

2. As a result, the **Tables** menu on the left side of the main application window will display the properties of the selected MSI file.

3. Next, navigate **Transform** -> **New Transform**.



4. To proceed, select **Property** from the **Tables** menu, then double-click any empty row on the right side of the application window.

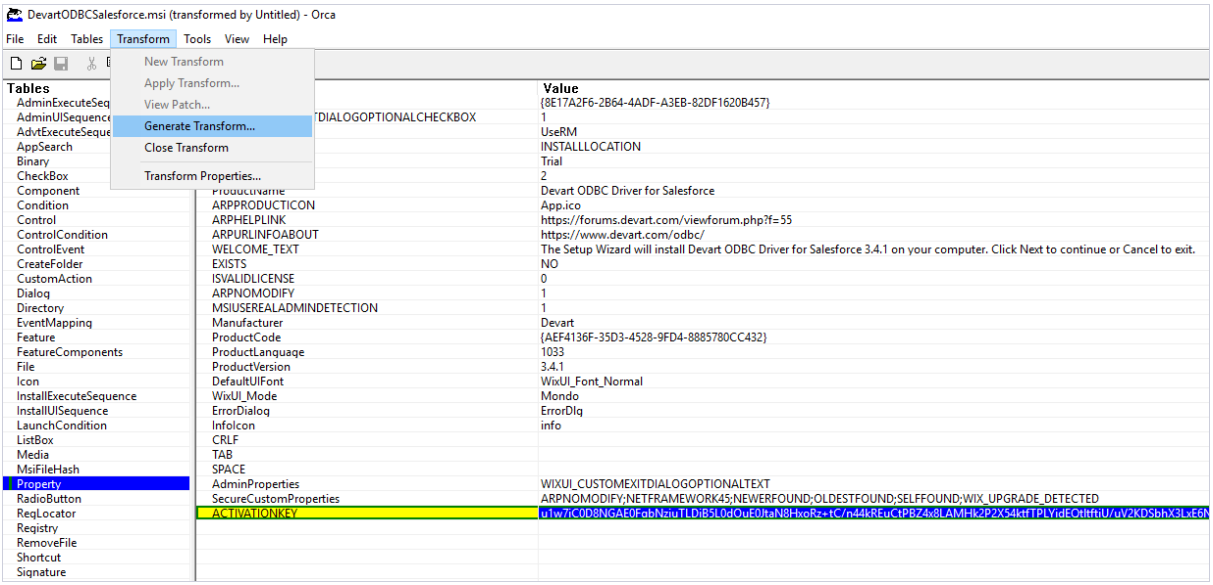


In the **Add Row** dialog that opens, make the following settings and press **OK** to apply the changes:

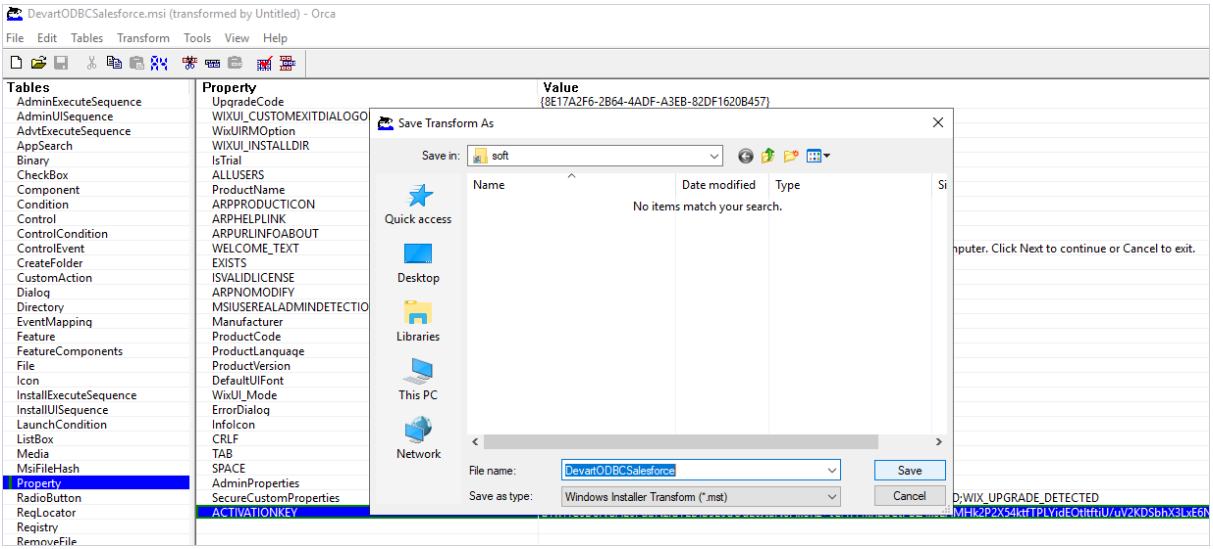
- **Property** - enter **ACTIVATIONKEY** with capital letters only.
- **Value** - enter the valid OEM license key for the ODBC Driver for PostgreSQL.

As shown in the following screen, a new property, **ACTIVATIONKEY**, has been added, with the license key displayed in the value column next to it.

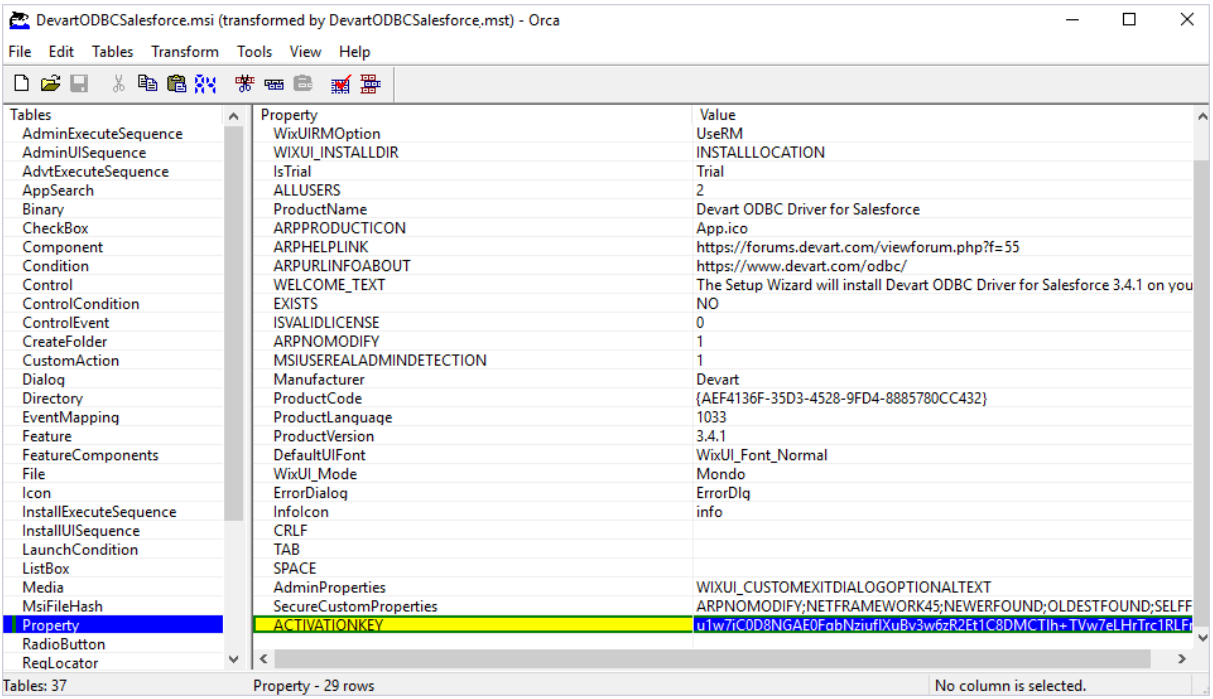
5. Once the configuration changes have been made, select **Transform** -> **Generate Transform**.



6. In the **Save Transform As** dialog that appears, enter a suitable name for the new MST file and click **Save** to apply your settings.



7. If successful, the encryption message *DevartODBCSalesforce.msi (transformed by DevartODBCSalesforce.mst) - Orca* will be displayed at the top of the Orca application window.



In case of a positive outcome, the newly created MST file will be located in the folder you specified, alongside the MSI file.

3.2.2 Deployment and Activation

Installing and Activating Software Remotely

Group Policy automated-program installation is specifically designed for deploying Windows Installer packages (MSI files). Therefore, when deploying the ODBC Driver for PostgreSQL using Group Policy, be sure to use the corresponding MSI file for the ODBC Driver for PostgreSQL.

Prerequisites: Locating the MSI Installation File

Prior to making configuration settings in the Group Policy, you'll need to create a distribution folder:

1. Create a shared network folder on the publishing server.
2. Set the appropriate sharing permissions on this folder to allow read access to the driver installation package for all domain users.

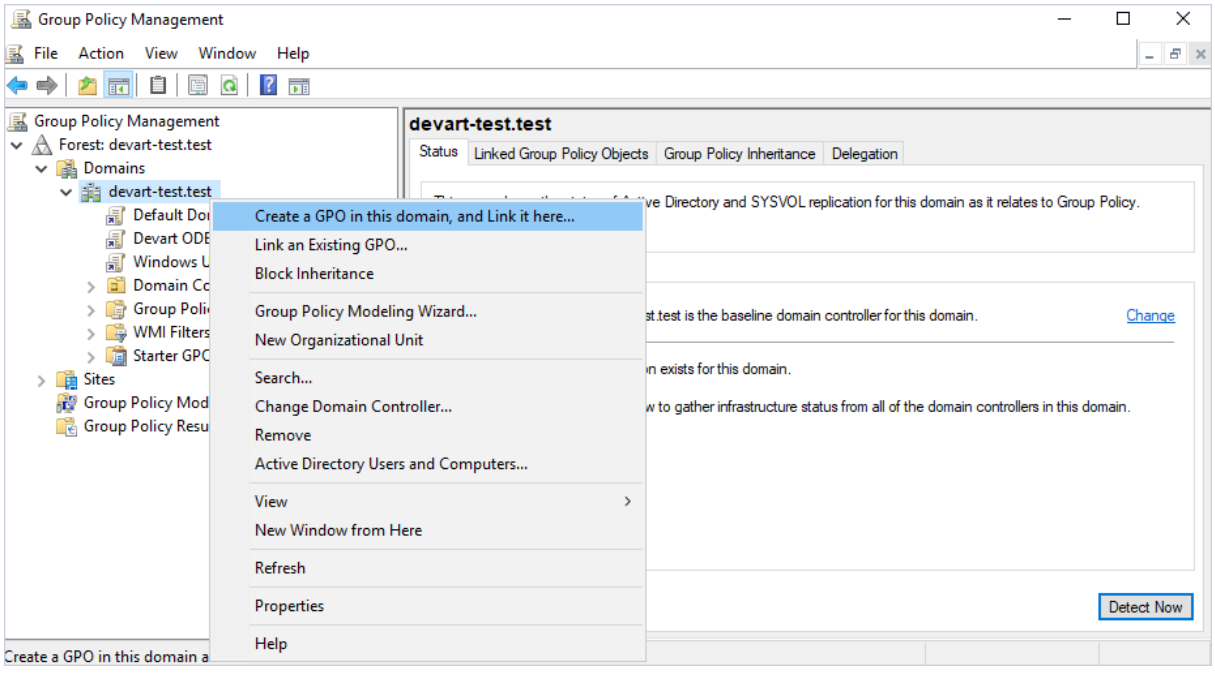
3. Download the ODBC Driver for PostgreSQL MSI file, and place it in the network folder.

> The MSI file for the ODBC Driver for Salesforce is taken as an example to illustrate the Group Policy installation process. Use the same steps described in this section when installing the ODBC Driver for PostgreSQL.

Further in this section, you'll find more detailed information on how to deploy and activate the ODBC Driver for PostgreSQL on remote client computers using Group Policy.

Server-Side Actions

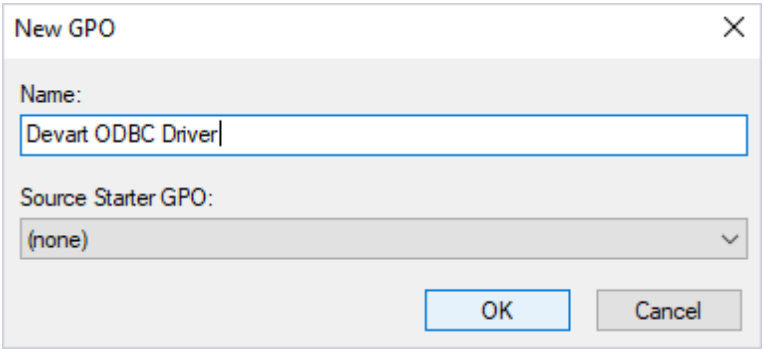
1. Open the **Group Policy Management** desktop application.
2. In the **Group Policy Management** window, navigate to the desired forest node, then expand the appropriate option under the **Domains** node. For this example, we'll select **devart-test.test**. Right-click the Domains node, and from the context menu, select **Create a GPO in this domain, and Link it here**.



3. You can now create a New Group Policy Object. In the **New GPO** dialog enter a name for

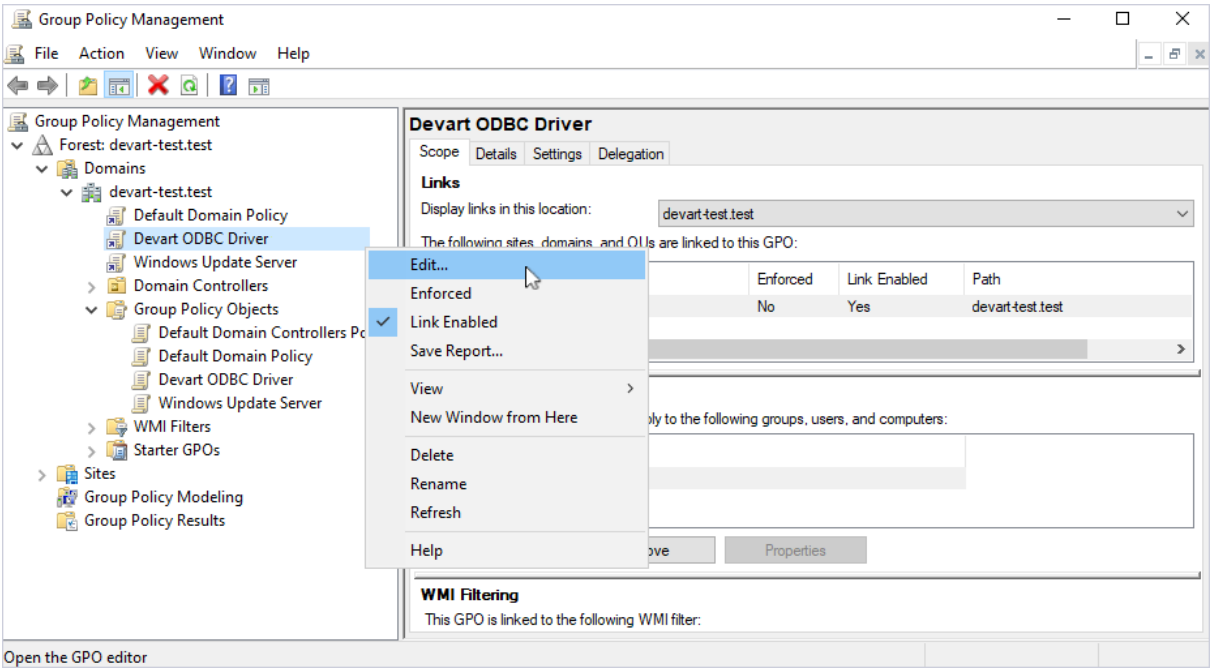
the new object and click **OK**. The new GPO will then appear within the **Group Policy Management** container.

For example, let's create a GPO named after the ODBC driver name.

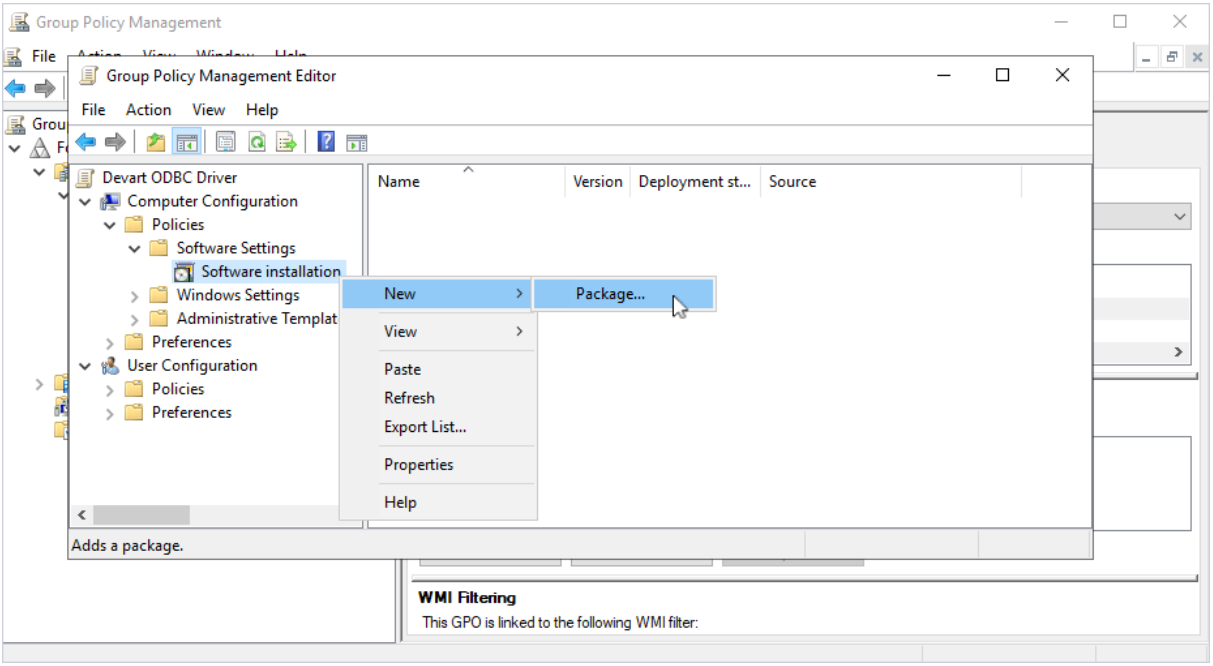


Keep in mind that each ODBC Driver for PostgreSQL Windows installation package corresponds to one Group Policy Object (GPO), which is important for managing future software upgrades. To install multiple drivers using Group Policy, you need to create a separate GPO for each driver you want to deploy.

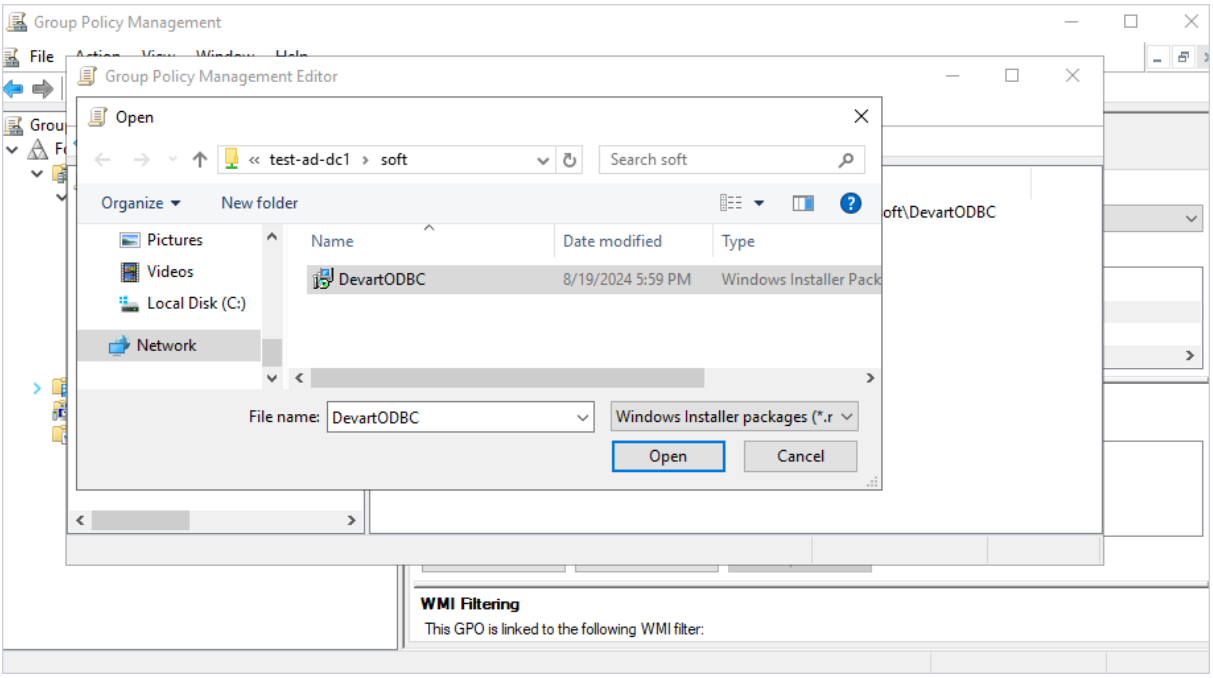
4. Right-click the new object and select **Edit** from the context menu.



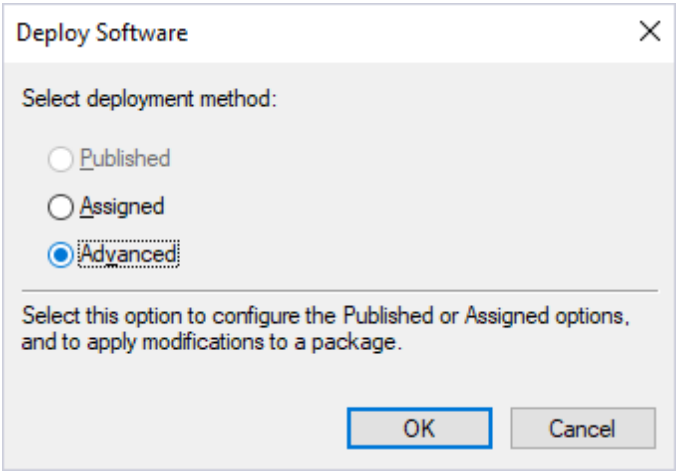
5. In the left pane of the **Group Policy Management Editor**, navigate to **Computer Configuration --> Policies --> Software Settings --> Software installation**. Your current deployment package will appear in the right pane. Right-click **Software installation**, then select **New --> Package**.



6. In the **Group Policy Management Editor** dialog that opens, select the desired MSI installation file and click **Open**.

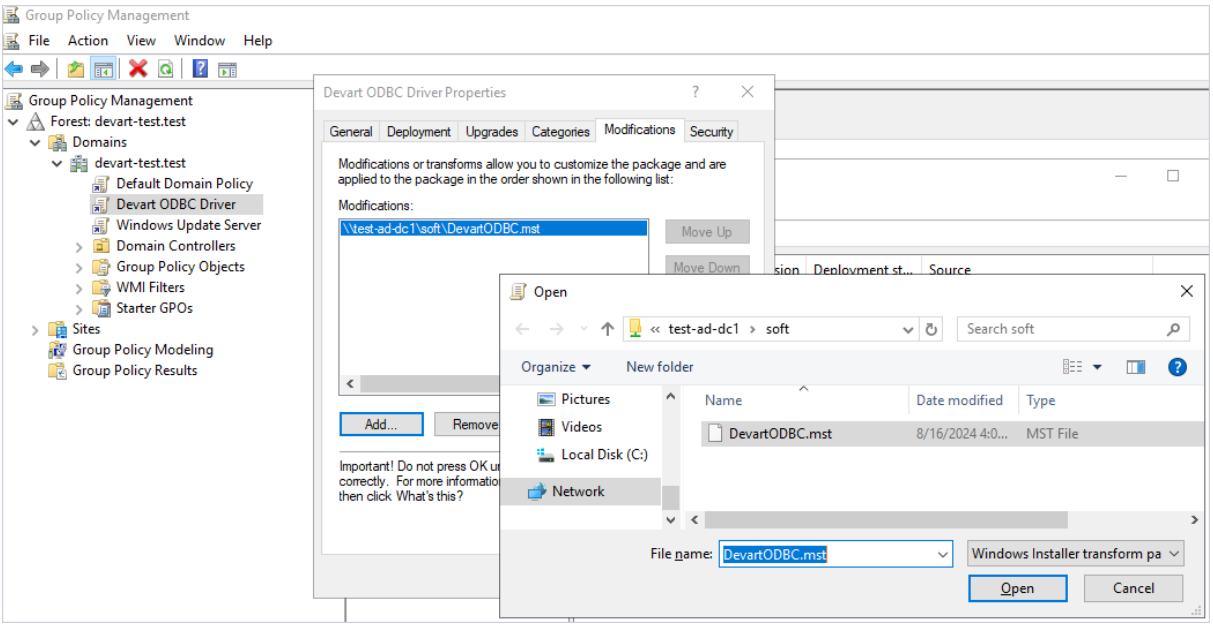


7. In the **Deploy Software** dialog, select **Advanced** to specify the software deployment method. The **Advanced** deployment method allows you to make necessary modifications to the MSI file, such as creating the MST file in Orca.

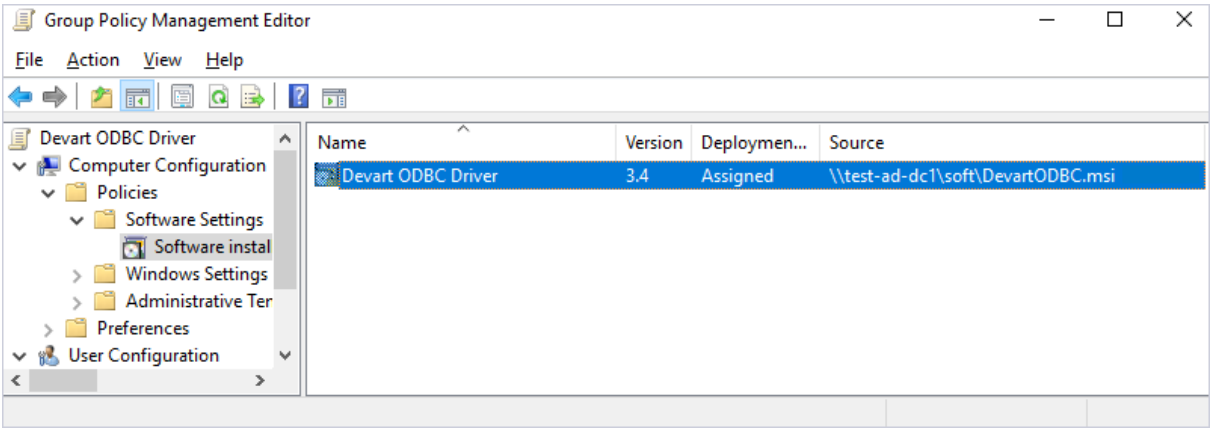


8. In the **Properties** dialog of the installation package that opens, go to the **Modifications** tab and select **Add**. Browse for the corresponding MST file, select it, and click **Open** to apply

the settings.



9. If configured correctly, the **Group Policy Management Editor** window should look as follows:

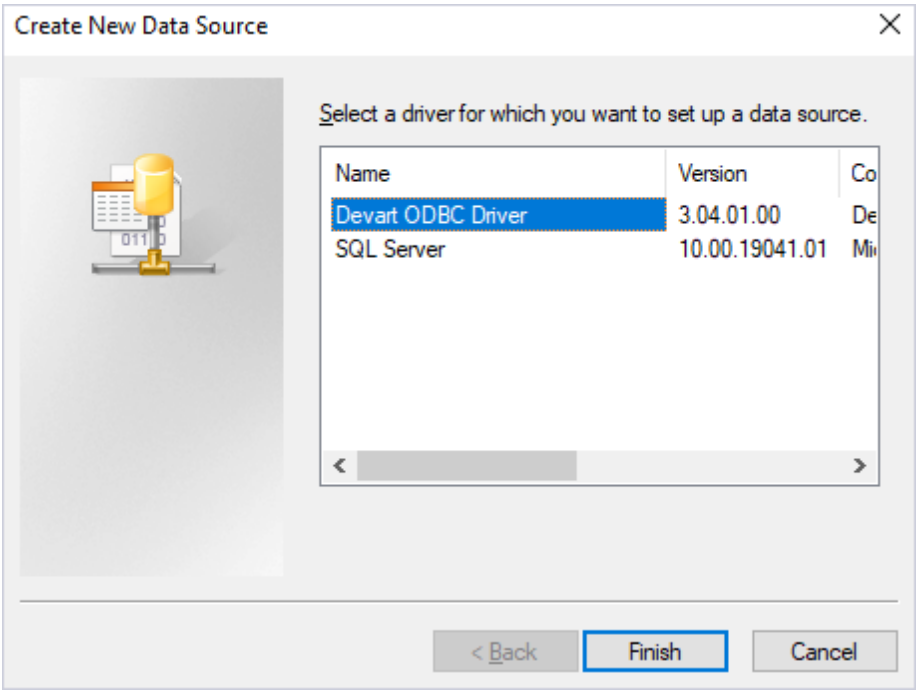


Client-Side Actions

For the ODBC Driver for PostgreSQL to be successfully installed on remote client machines, all domain users must restart their computers after logging in for the first time.

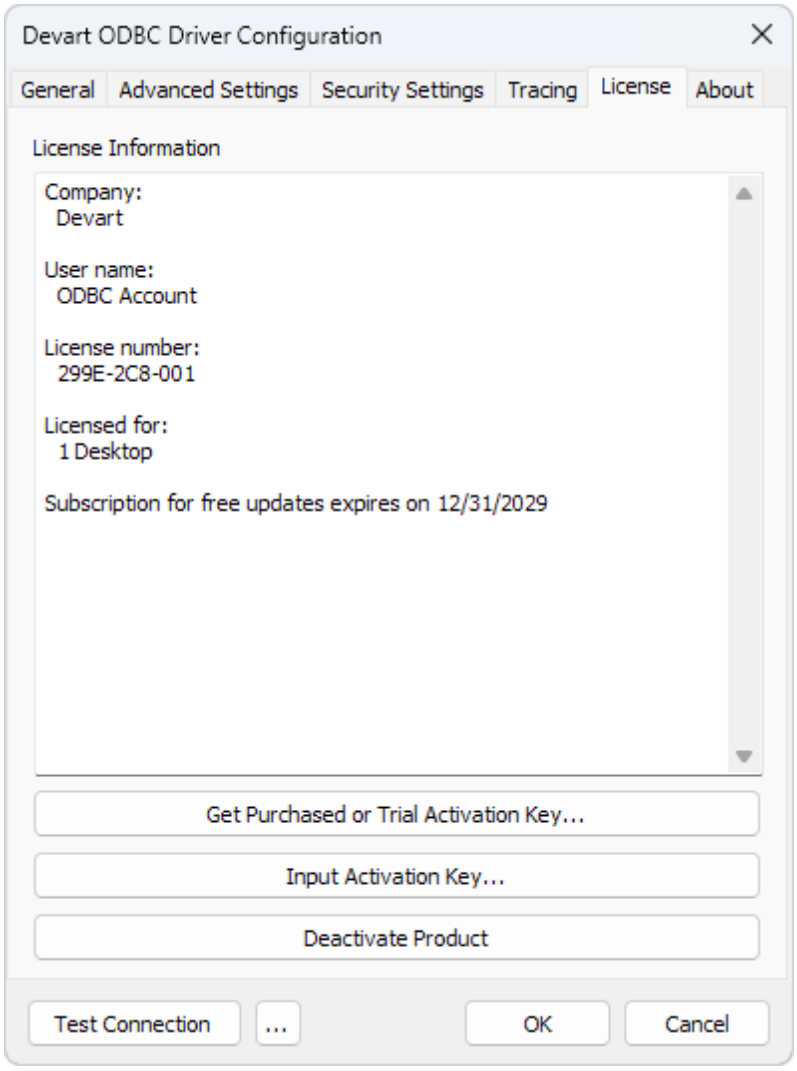
In case of successful deployment, the ODBC driver will be installed on the client's computer.

To verify, open the ODBC Data Source Administrator on the client's machine and add the deployed ODBC driver.



All information on the deployed driver is accessible upon clicking the **About** tab.

Similarly, the valid license key will be automatically activated after the successful installation of the ODBC Driver for PostgreSQL.



See Also

- Creating the MST File Using Orca
- Activating on Windows - ODBC Driver for PostgreSQL
- License Information - ODBC Driver for PostgreSQL

3.2.3 Software Upgrade

Automatic Software Update Using Group Policy

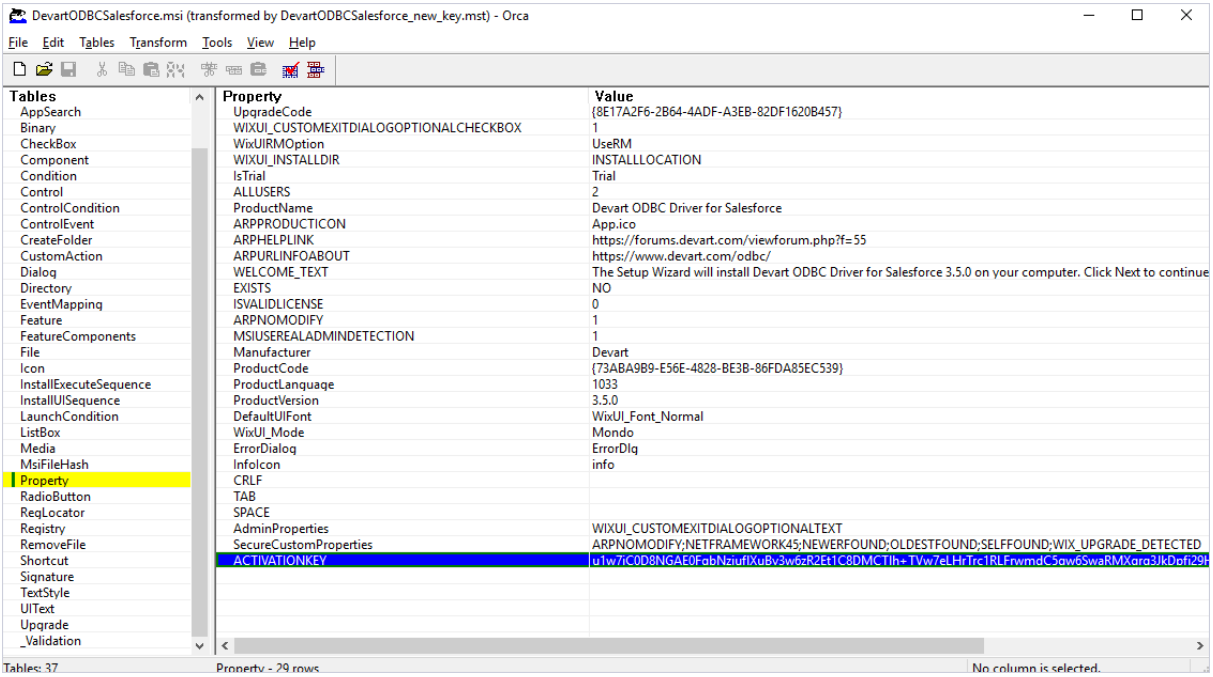
If the ODBC Driver for PostgreSQL was initially deployed through Group Policy, it can be easily updated to a newer version. Follow the steps below to update both the ODBC Driver for

PostgreSQL and the license to newer versions on all remote computers in the domain.

1. Download the ODBC Driver for PostgreSQL installation MSI file of a newer version and place it in the shared network folder.

2. Create a new MST file with a new license key using Orca.

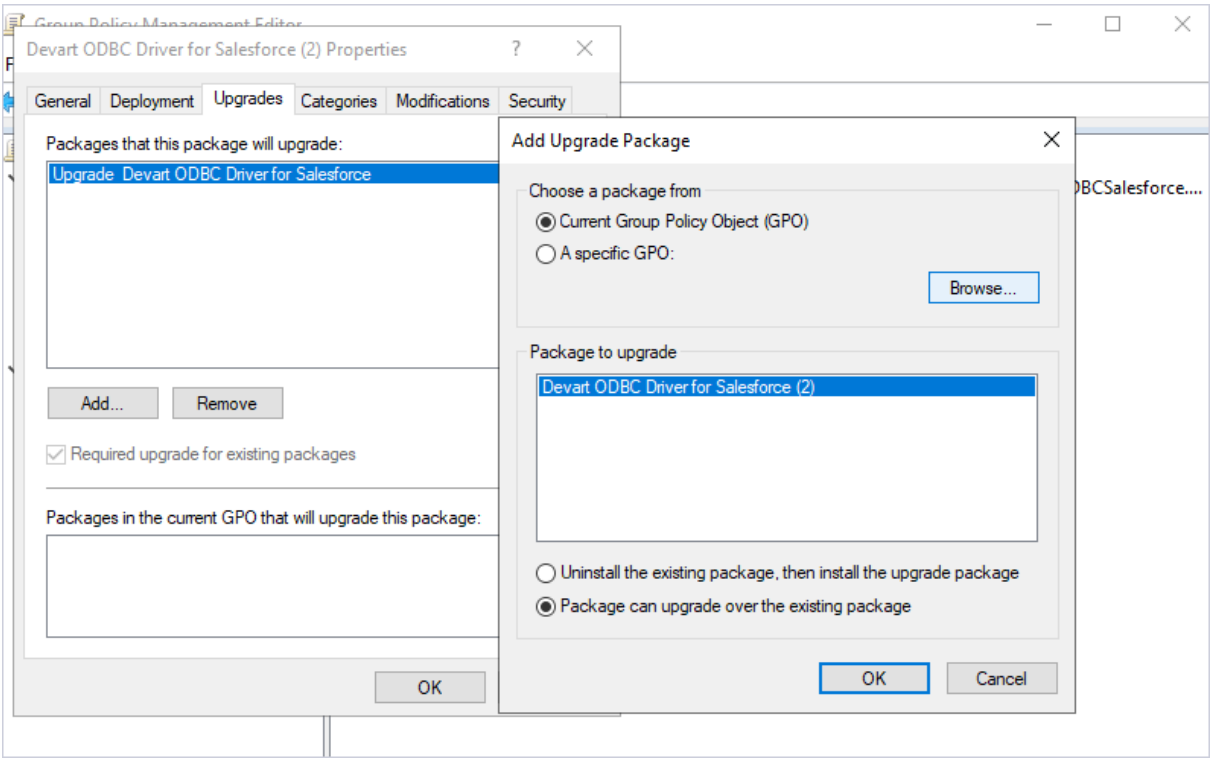
> If your license is still valid, there's no need to create a new MST file. Use the current MST file instead.



> The MSI file for the ODBC Driver for Salesforce is taken as an example to illustrate the Group Policy installation process. Use the same steps described in this section when installing the ODBC Driver for PostgreSQL.

3. Follow the same workflow as outlined in Step 4 to Step 7 of the ODBC Driver for PostgreSQL Remote Deployment and Activation section.

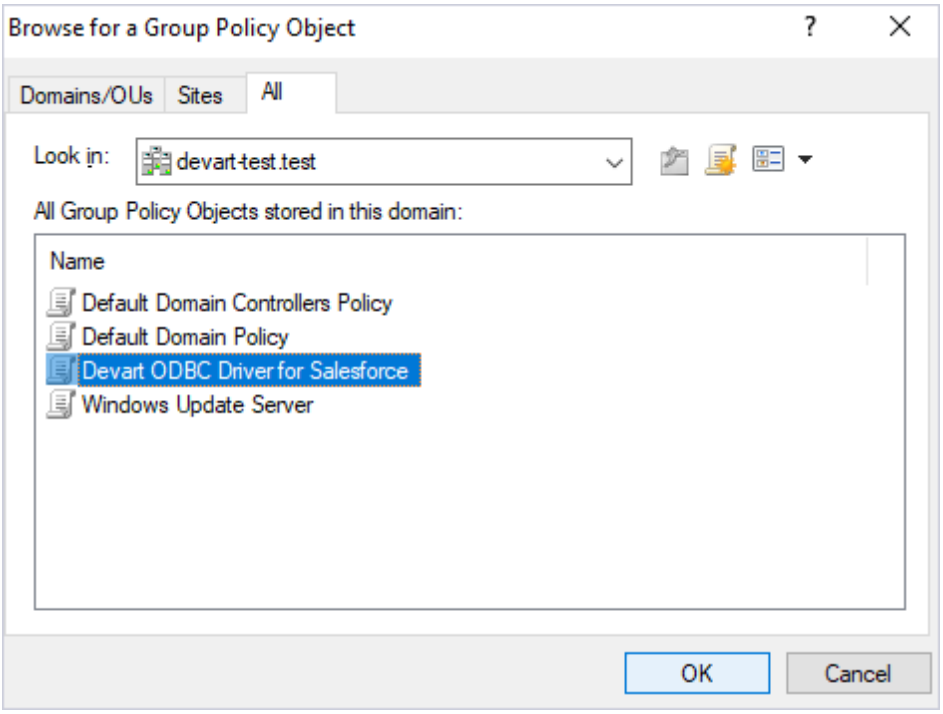
4. In the **Properties** dialog that appears after selecting the **Advanced** deployment method, go to the **Upgrades** tab and click **Add**.



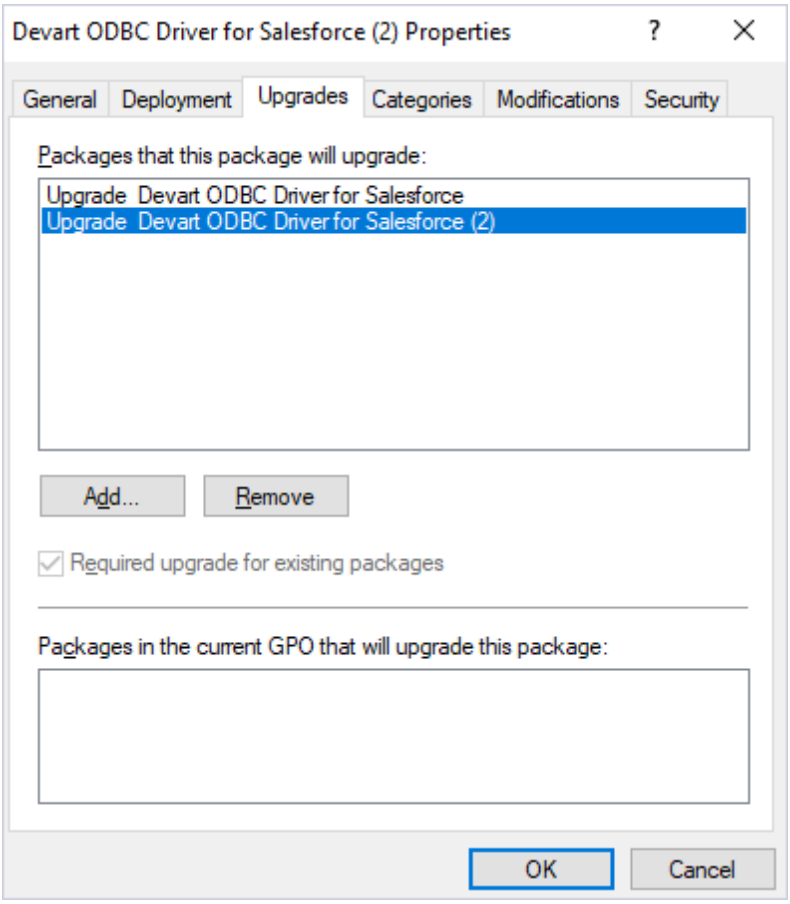
Make sure to select the following check boxes while adding the package:

- **Current Group Policy Object**
- **Package can upgrade over the existing package**

5. Browse for the corresponding GPO object and click **OK** to apply the settings.

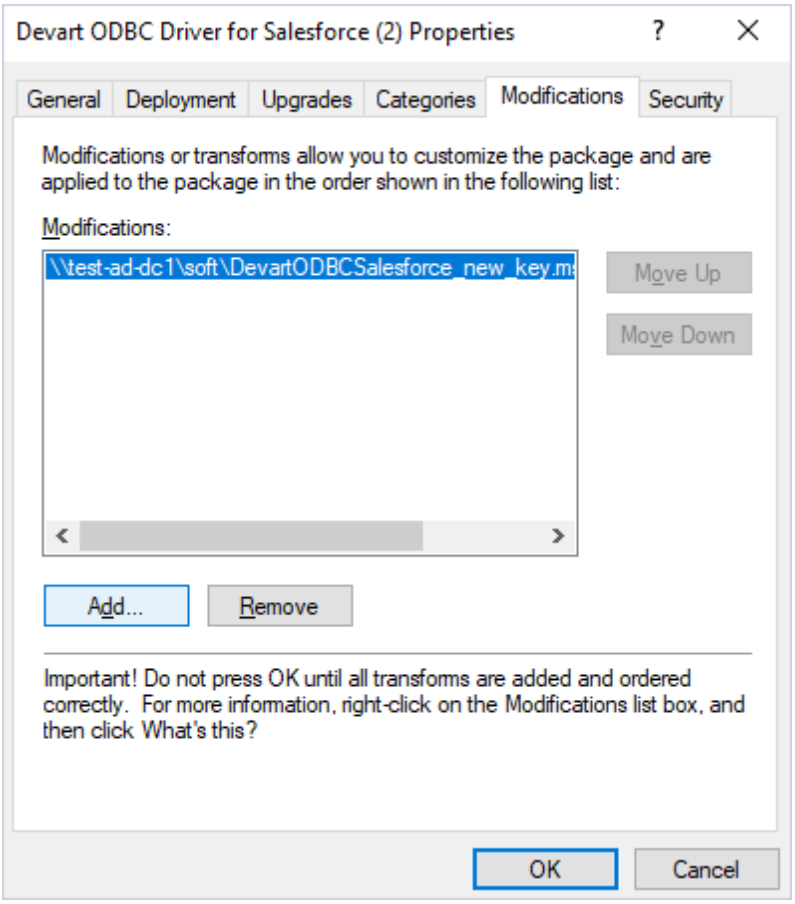


6. Now the **Upgrades** tab of the **Properties** dialog will list a new package with a newer version.

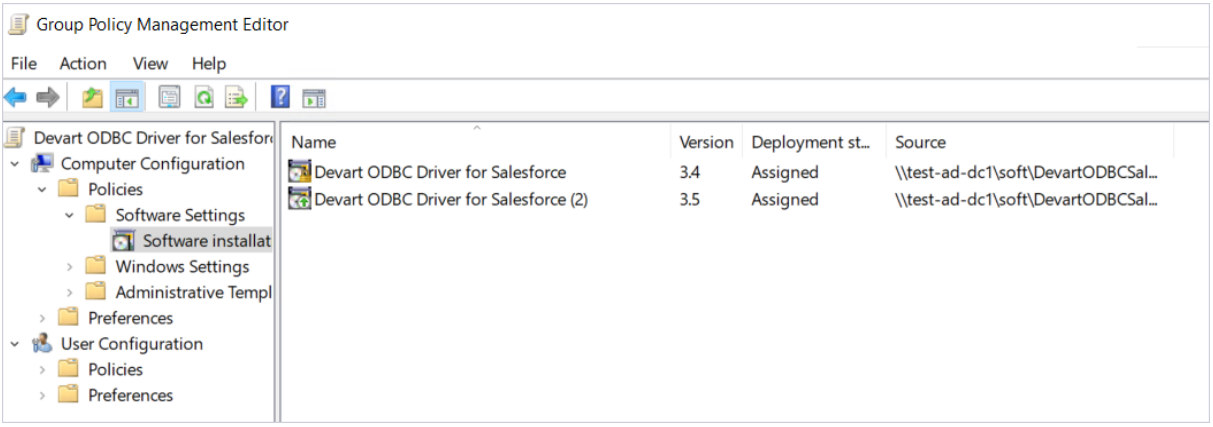


7. Go to the **Modifications** tab in the same properties dialog, click **Add** and browse to the MST file.

   We have already created a new MST file with a new license key in Step 2.



8. In case of a positive outcome both the old and new versions of the driver package will be displayed in the Group Policy Management Editor.



Once the GPO configuration on the server is complete, the ODBC Driver for PostgreSQL will

automatically update to the latest version each time a client computer restarts.

Client-Side Actions

To update the ODBC Driver for PostgreSQL to a newer version on remote client machines, all domain users must restart their computers after their first login.

If successful, both the driver and the license key will be automatically updated to the new version on remote computers. For detailed instructions on how to view the technical details of the ODBC Driver for PostgreSQL after upgrading, refer to Client-Side Actions.

See Also

- Creating the MST File Using Orca
- Remote Deployment and Activation - ODBC Driver for Microsoft Access
- Activating on Windows - ODBC Driver for PostgreSQL
- License Information - ODBC Driver for PostgreSQL

3.3 Product Activation

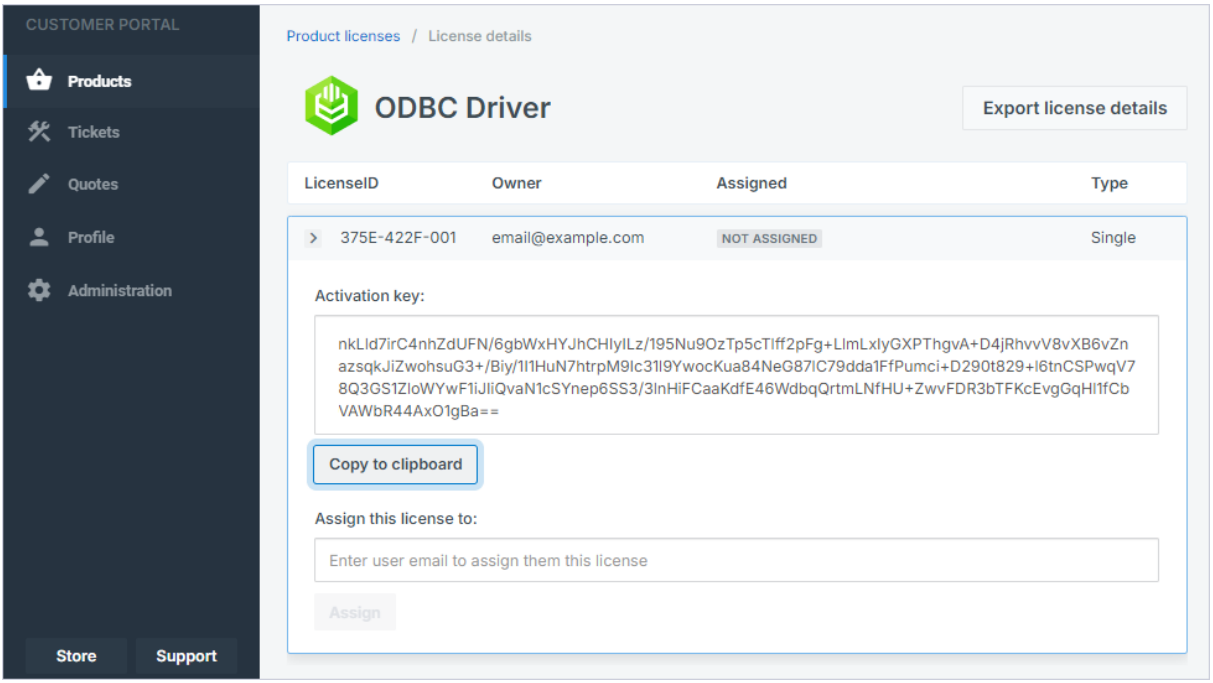
See how to activate Devart ODBC Driver for PostgreSQL:

- Obtaining Activation Key
- Activation on Windows
- Activation on macOS
- Activation on Linux
- Where to see the license information

3.3.1 Obtaining Activation Key

Follow these steps to obtain your product activation key:

- **From the Customer Portal:**
  1. Open the Customer Portal and sign in.
  2. On the **Product licenses** page, select the driver.
  3. Click **Copy to clipboard** to copy the activation key.



- **From the registration email:**
  1. Locate the registration email you received from Devart after installing the driver. This email contains a Purchased or Trial activation key.
  2. Copy the activation key.

See also:

- Activation on Windows
- Activation on macOS
- Activation on Linux

3.3.2 Activation on Windows

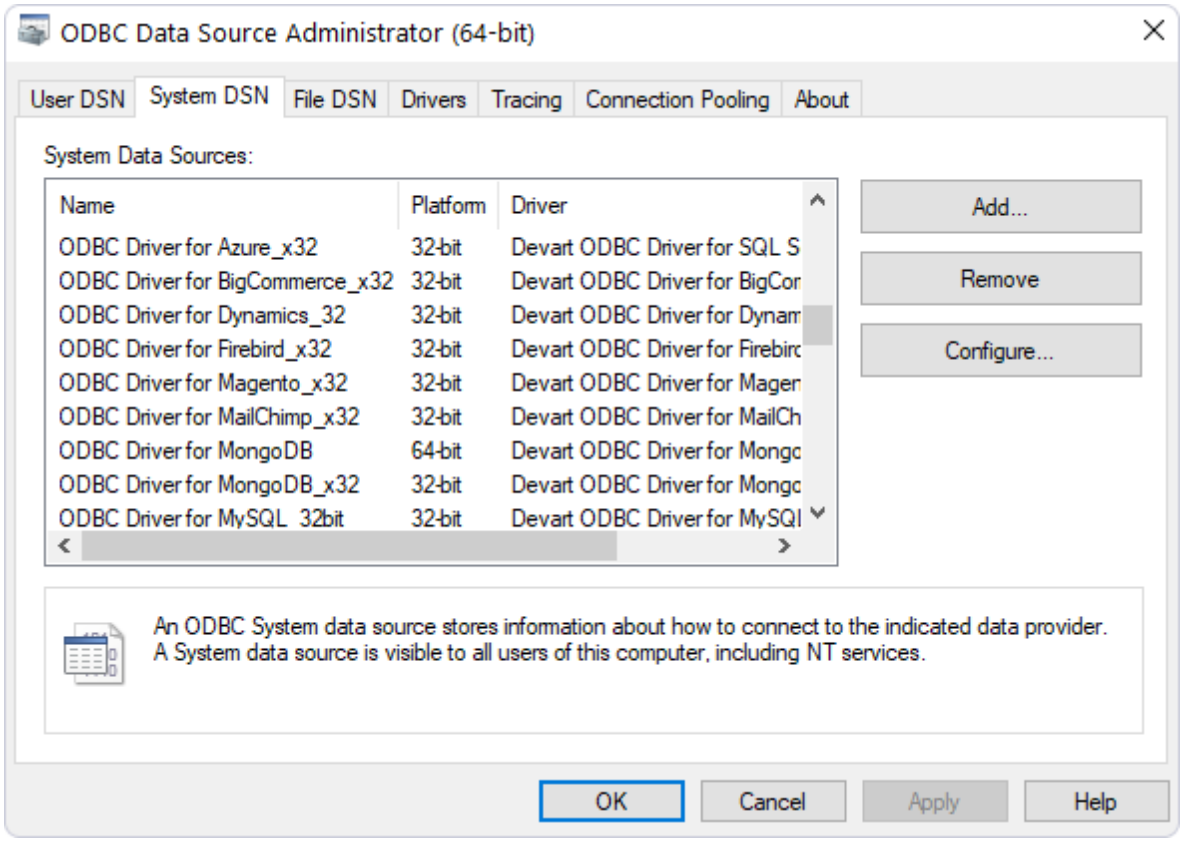
Driver Activation After Installation

To activate your installed driver, perform the following steps.

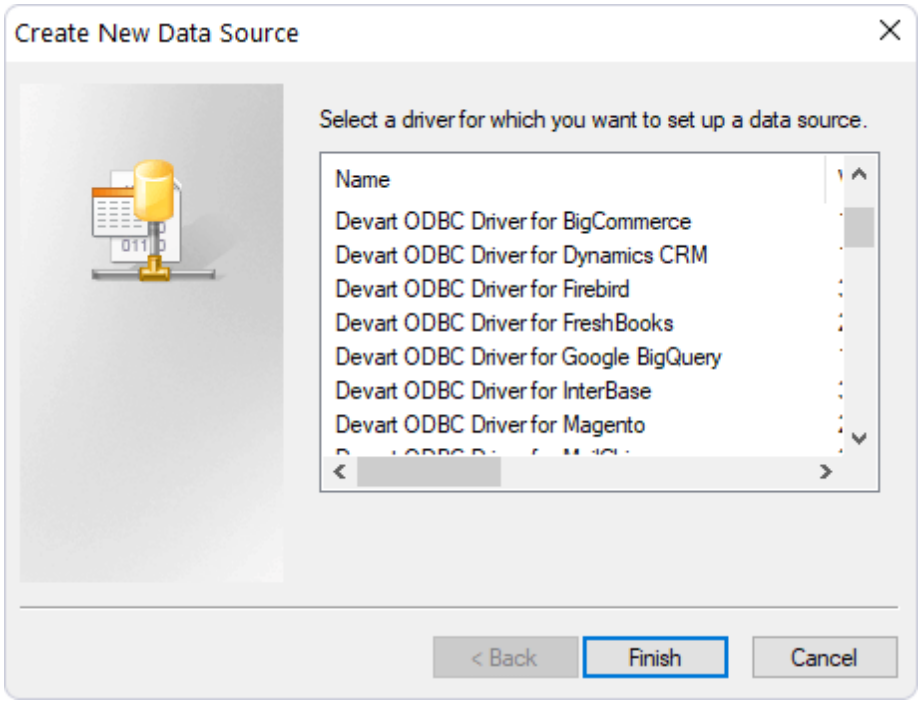
You need to activate the driver even for the trial version.

1. Open the ODBC Data Source Administrator.

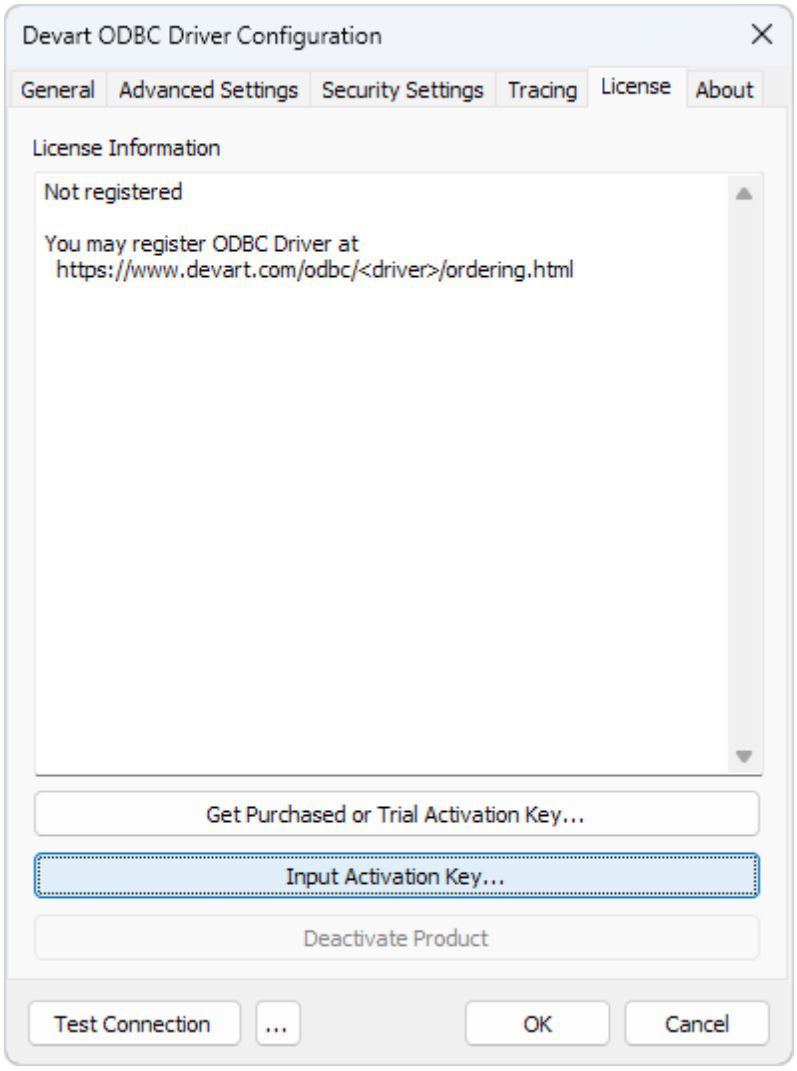
2. On the **System DSN** tab, click **Add**.



3. In the **Create New Data Source** dialog, select the installed driver, then click **Finish**.



4. In the configuration dialog, navigate to the **License** tab, and click **Input Activation Key**.

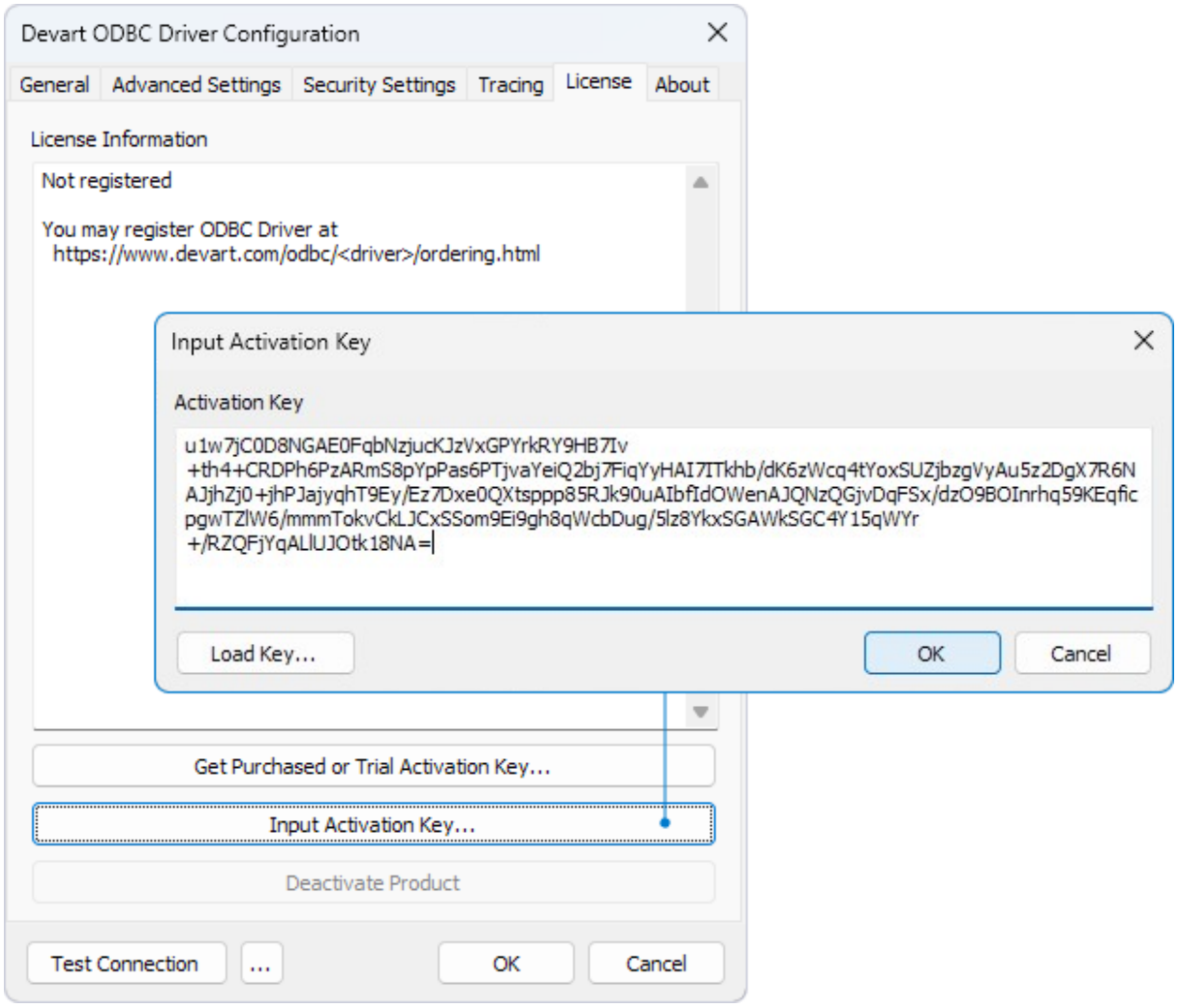


5. Here, you can activate the driver using one of the following methods:

- **Enter an activation key:** Paste your activation key into the corresponding box.
- **Load an activation file:** Click **Load Key** and select the file that contains the activation key.

You can find your activation key in the registration email or your Customer Portal account.

To open the Customer Portal, click **Get Purchased or Trial Activation Key**.



6. Click **OK**.

3.3.3 Activating on macOS

Driver Activation After Installation

If you didn't activate Devart ODBC Driver for PostgreSQL during installation, you can activate it later using one of two methods:

- Online via a console application (for Perpetual and Subscription licenses).
- Offline with an activation file (only for Perpetual licenses).

You need to activate the driver even for the trial version.

Activate Online via a Console Application

To activate the driver over the internet using a console application, follow these steps (this method works for both Perpetual and Subscription licenses):

1. In the console, go to the folder where the driver was installed. The default installation path is `/Library/ODBC/Devart/PostgreSQL`.
2. Optional: To open the Customer Portal in your browser and locate your activation key, run the following command:

```
./postgresqlodbcactivator -g
```

Alternatively, you can find your activation key in the registration email.

3. Run the activation command with superuser privilege, providing either the driver activation key or the path of the file with the key:

- To activate using the activation key:

```
sudo ./postgresqlodbcactivator -a <activation_key>
```

Replace `<activation_key>` with the driver activation key.

- To activate using a file:

```
sudo ./postgresqlodbcactivator -a <file_path>
```

Replace `<file_path>` with the full path of the file containing the driver activation key.

When the process is complete, the driver is activated, and the License Summary is displayed in the console.

Activate Offline With an Activation File

To activate the driver offline (only for Perpetual licenses), follow these steps:

1. Go to the folder where the driver was installed. The default installation path is `/Library/ODBC/Devart/PostgreSQL`.
2. In that folder, create a file with the `activation.key` name.

3. Copy the activation key from the registration email or your Customer Portal account and paste it into the created file.
4. Save the file.

The driver gets activated.

See also:

- Activation on Windows
- Activation on Linux

3.3.4 Activation on Linux

Driver Activation After Installation

If you didn't activate Devart ODBC Driver for PostgreSQL during installation, you can activate it later using one of two methods:

- Online via a console application (for Perpetual and Subscription licenses).
- Offline with an activation file (only for Perpetual licenses).

You need to activate the driver even for the trial version.

Activate Online via a Console Application

To activate the driver over the internet using a console application, follow these steps (this method works for both Perpetual and Subscription licenses):

1. In the console, go to the folder where the driver was installed. The default installation path is:
   - For the DEB package: `/usr/share/devart/odbcpostgresql`
   - For the RPM package: `/usr/local/devart/odbcpostgresql`
2. Optional: To open the Customer Portal in your browser and locate your activation key, run the following command:

```
./postgresqlodbcactivator -g
```

Alternatively, you can find your activation key in the registration email.

3. Run the activation command with superuser privilege, providing either the driver activation key or the path of the file with the key:

- To activate using the activation key:

```
sudo ./postgresqlodbcactivator -a <activation_key>
```

Replace `<activation_key>` with the driver activation key.

- To activate using a file:

```
sudo ./postgresqlodbcactivator -a <file_path>
```

Replace `<file_path>` with the full path of the file containing the driver activation key.

When the process is complete, the driver is activated, and the License Summary is displayed in the console.

Activate Offline With an Activation File

To activate the driver offline (only for Perpetual licenses), follow these steps:

1. Go to the folder where the driver was installed. The default installation path is:

- For the DEB package: `/usr/share/devart/odbcpostgresql`
- For the RPM package: `/usr/local/devart/odbcpostgresql`

2. In that folder, create a file with the `activation.key` name.

3. Copy the activation key from the registration email or your Customer Portal account and paste it into the created file.

4. Save the file.

The driver gets activated.

See also:

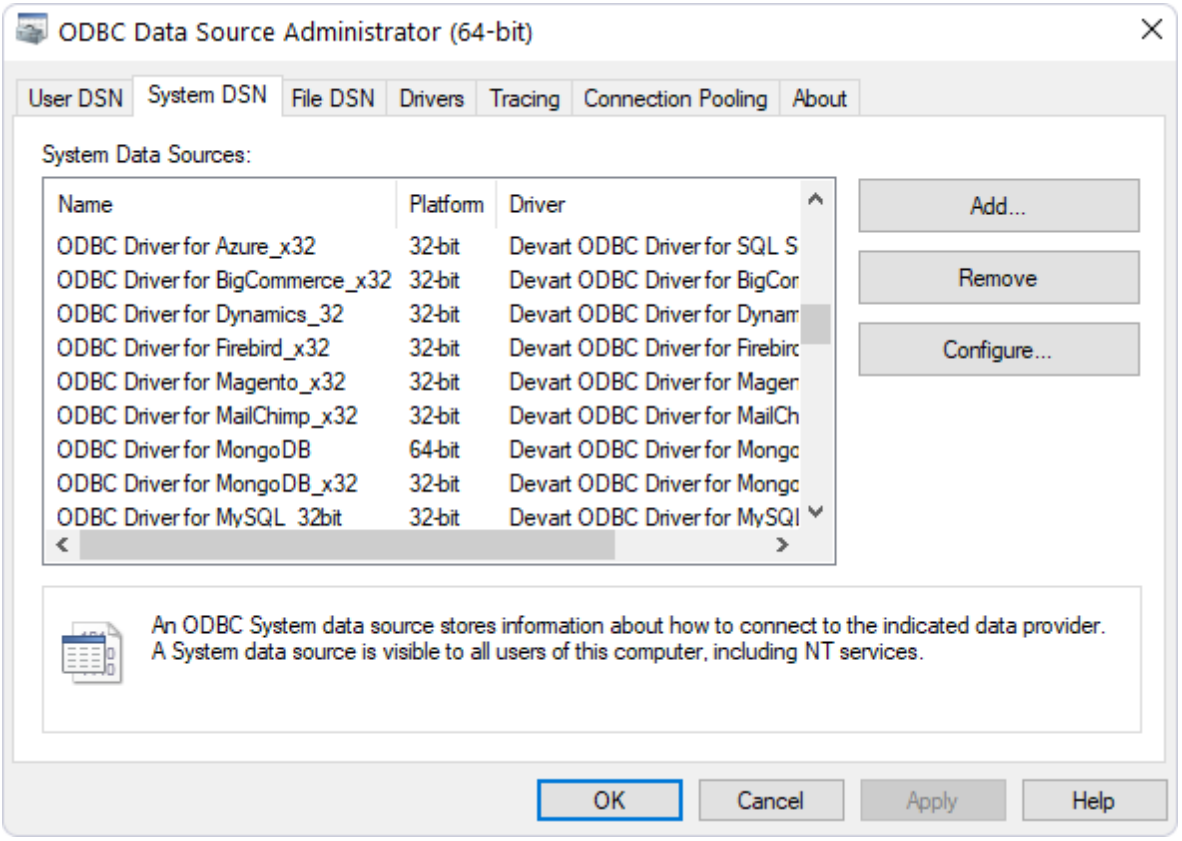
- Activation on Windows

- Activation on macOS

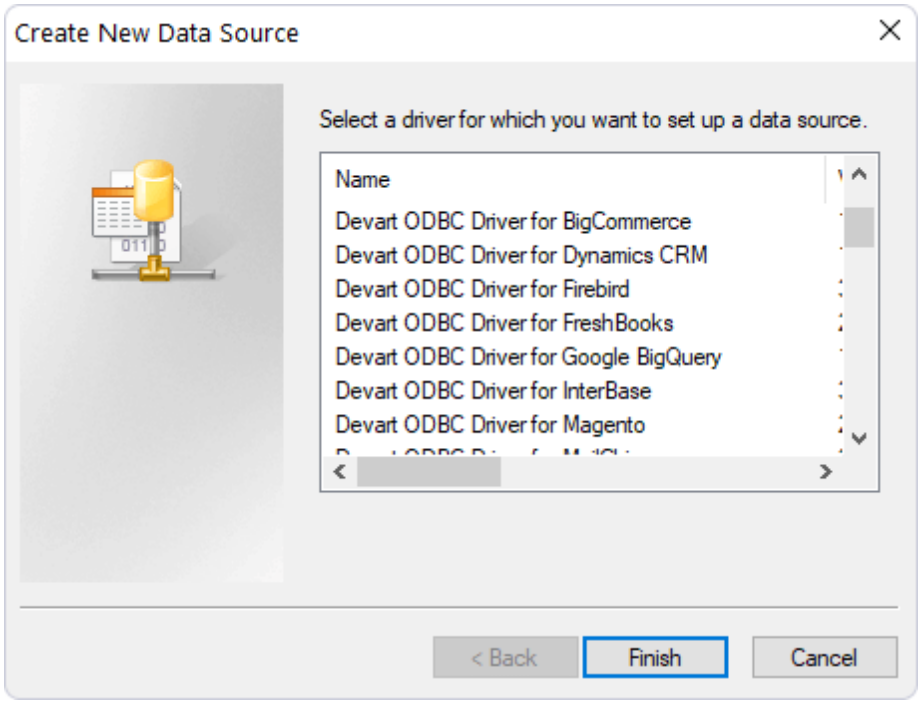
3.3.5 Where to See the License Information?

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

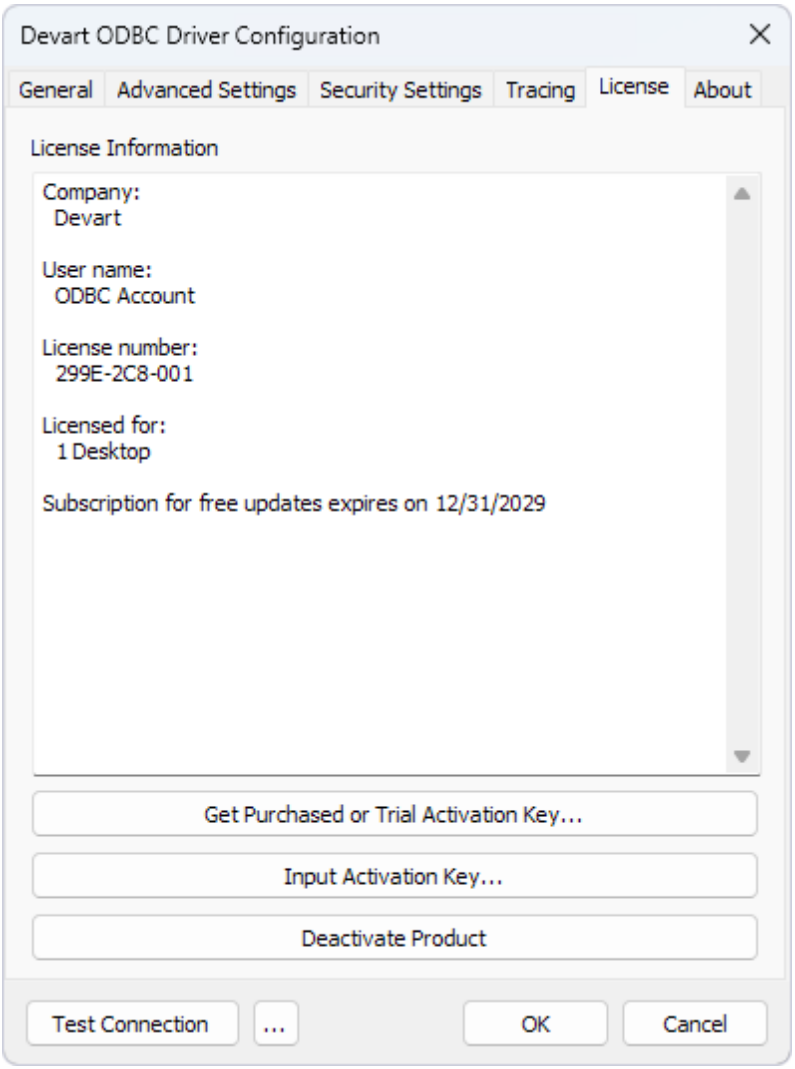
1. Open the ODBC Data Source Administrator.
2. On the **System DSN** tab, click **Add**.



3. Select the driver, then click **Finish**.



4. In the configuration dialogue, navigate to the **License** tab to view the license details.



3.4 Connecting to PostgreSQL

See how to connect to the ODBC Driver for PostgreSQL:

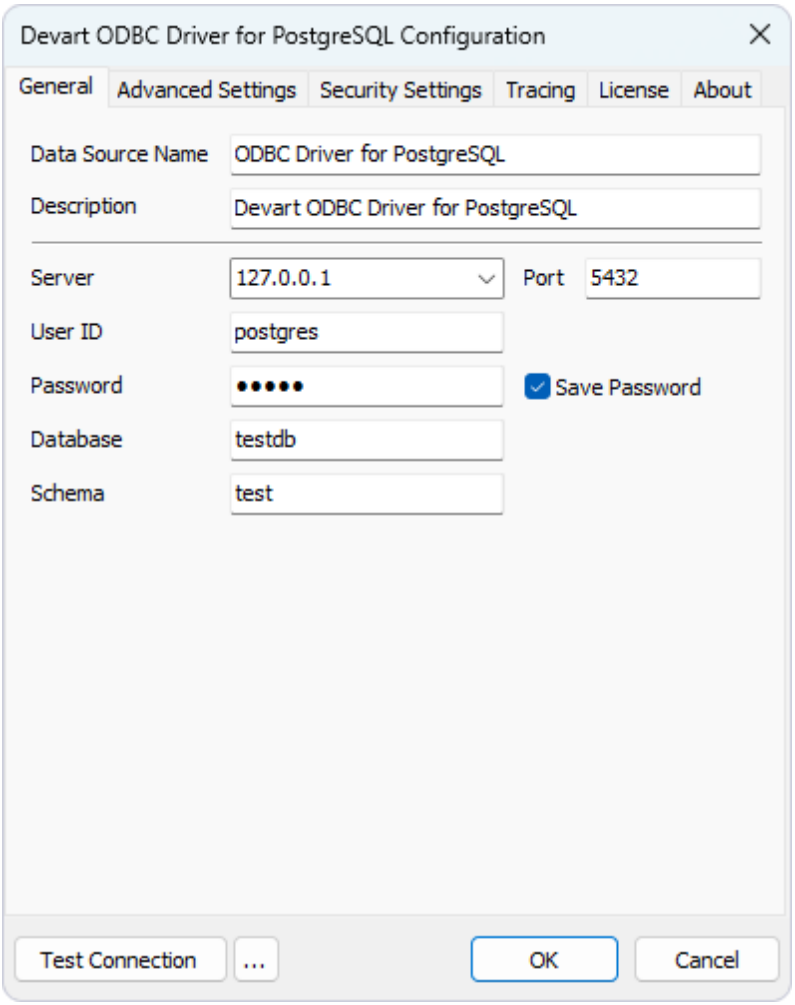
- Windows DSN Configuration
- macOS DSN Configuration
- Linux DSN Configuration

3.4.1 Windows

Windows DSN Configuration

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

1. Open the ODBC Data Source Administrator.
   - Type `ODBC Data Sources` in the Windows search box and choose the application that matches the bitness of the third-party 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** tab. Most applications work with both types, yet some applications require a specific type of DSN.
3. Click **Add**. The **Create New Data Source** dialog will appear.
4. Select **Devart ODBC Driver for PostgreSQL** and click **Finish**. The driver setup dialog will open.
5. Enter the connection information in the appropriate fields.



6. You can test the connectivity by clicking **Test Connection**.

7. Click **OK** to save the DSN.

See Also

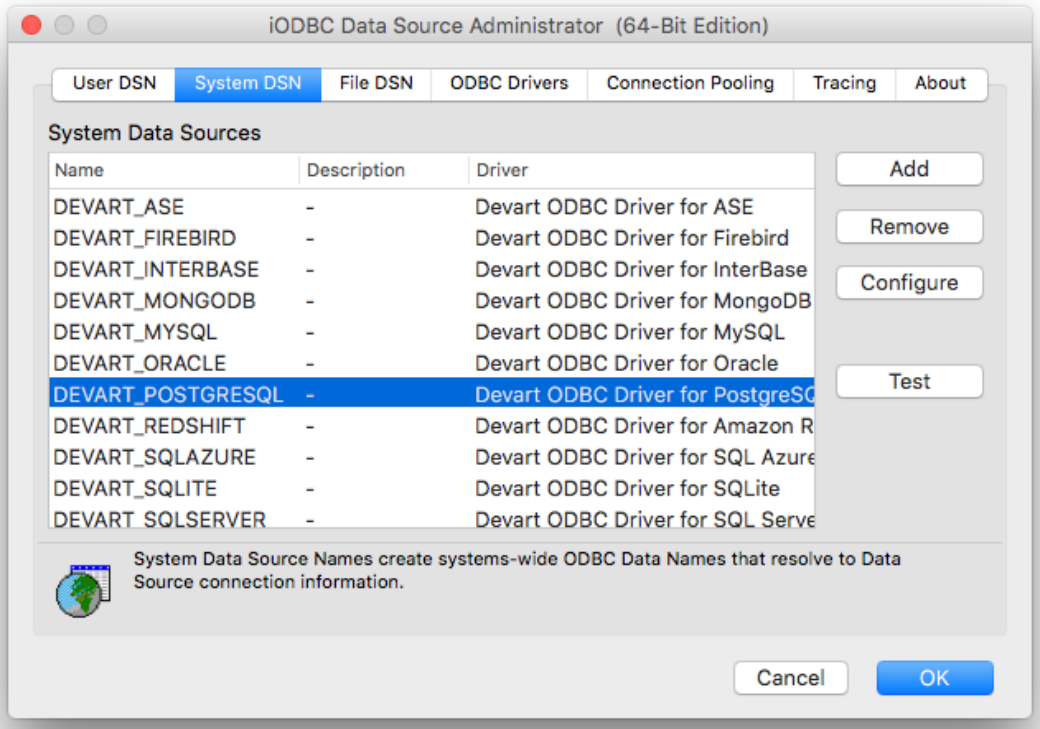
Connection Options

3.4.2 Mac

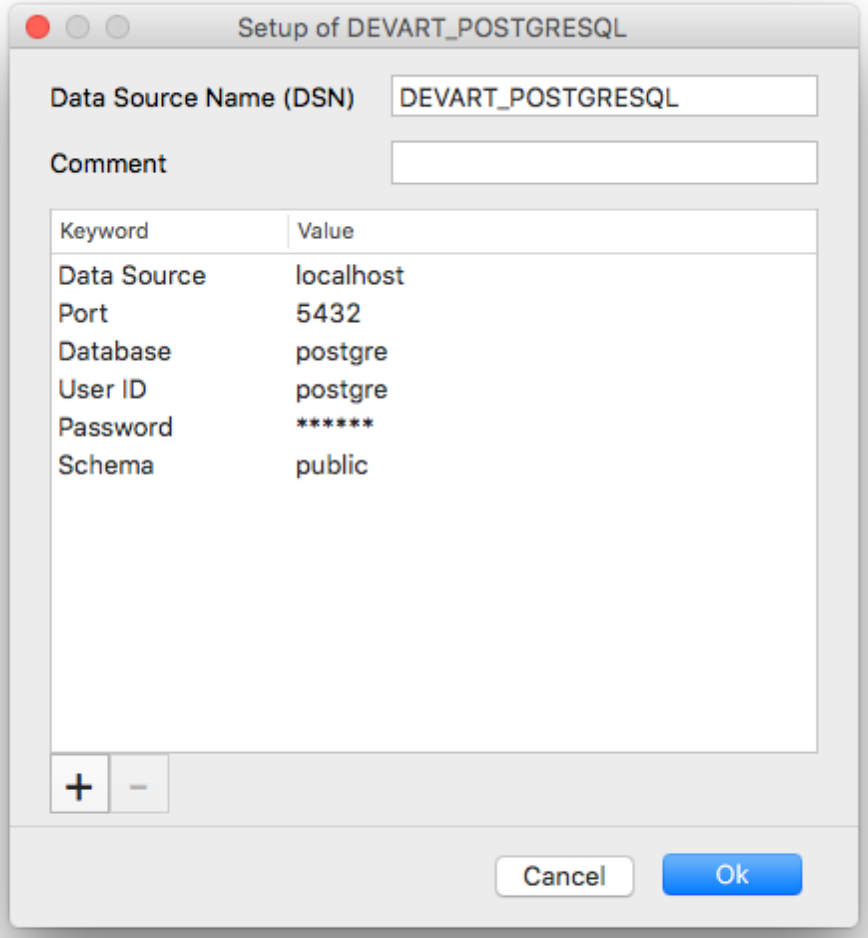
macOS DSN Configuration

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

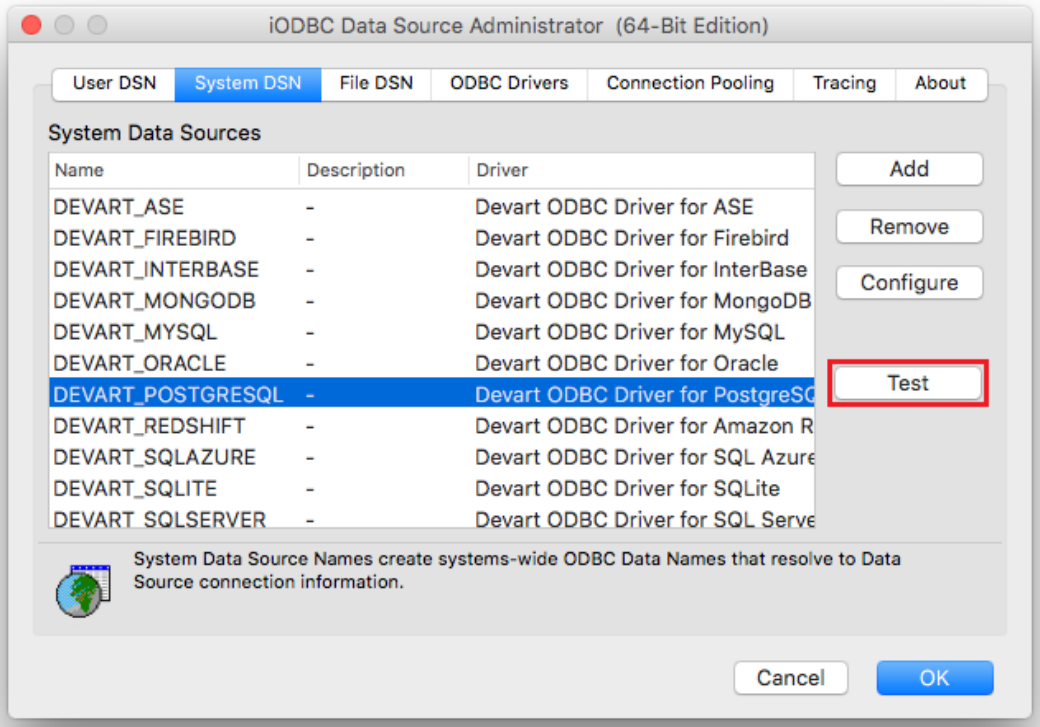
1. Run the iODBC utility of the required bitness. Find the DEVART_POSTGRESQL section and click the Configure button:



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

Connection Options

3.4.3 Linux

Linux DSN Configuration

After the linux (DEB or RPM) driver is installed, a DSN with the name DEVART_POSTGRESQL is created. You can use it to test the connection with the POSTGRESQL server. For this, perform the following steps:

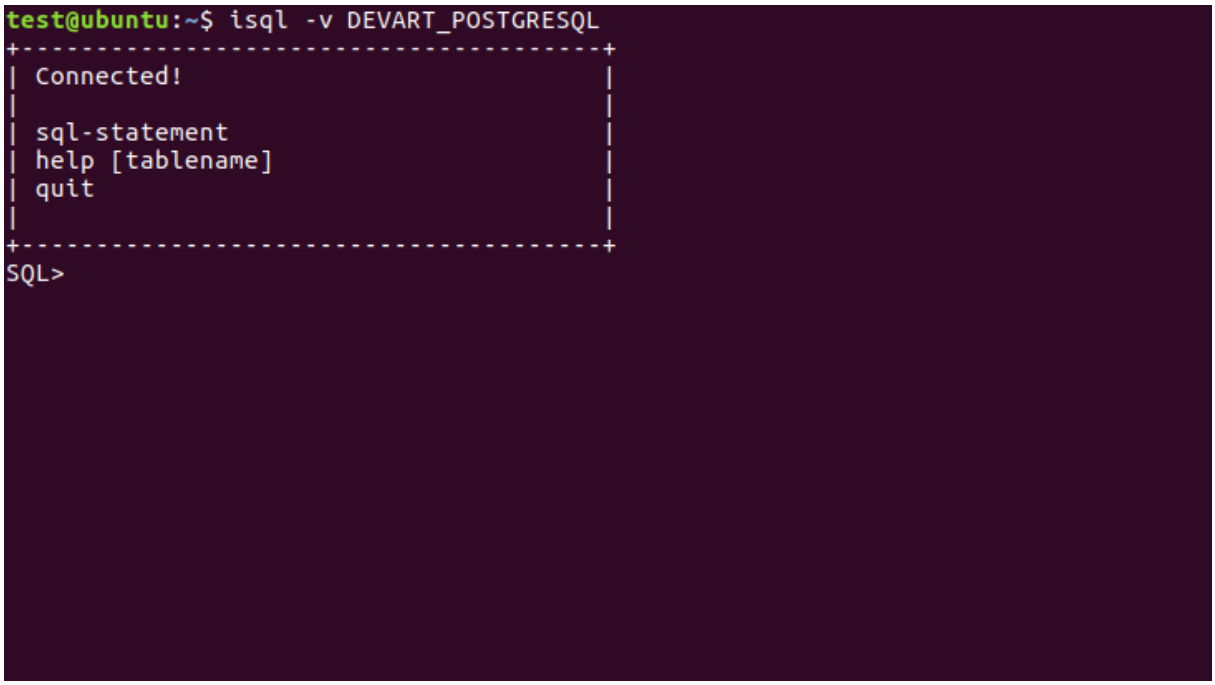
1. Open the odbc.ini file located in the /etc folder. Find the DEVART_POSTGRESQL section and specify the required connection settings:

```
User ID=<your PostgreSQL user name>
Password=<your PostgreSQL password>
```

```
Data Source=<your PostgreSQL data source>
Schema=<your PostgreSQL schema>
Port=<your PostgreSQL port>
Database=<your PostgreSQL database name>
```

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

```
isql -v DEVART_POSTGRESQL
```



See Also

Connection Options

3.5 Connection String Parameters

PostgreSQL ODBC Connection String Parameters

The following table lists the connection string parameters for PostgreSQL.

Parameter	Description

Server	Serves to supply the server name for login.
Port	Used to specify the port number for the connection. 5432 by default.
User ID	Used to supply a unique User ID for login.
Password	Used to supply a password for login.
Database	Used to set the name of the database
Schema	Used to change the search path of the connection to the specified schema, or get the first value from the search path.
Advanced Settings	
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	
Connection Timeout	The time (in seconds) to wait for a connection to open before terminating an attempt. The default value is 15.

`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`.

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.
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 parameter value should be changed if any third-party tool supports only Ansi string types or Unicode ones.
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 strings.
Charset	Used to set the character set that ODBC driver for PostgreSQL uses to read and write character data.
UseUnicode	Informs server that all data between client and server sides will be passed

	in Utf8 coding. Setting this option converts all fields of the String type into WideString that allows to work correctly with symbols of almost all languages simultaneously. On the other hand, it causes a delay in working. If the Use Unicode property is enabled, the Charset property will be ignored. The default value is False.
Protocol	Used to set the version of protocol for communication with PostgreSQL server.
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.
Uuid With Braces	Specifies whether UUIDs (Universally Unique Identifiers) are displayed with curly braces.
EnablePgGIS	Enables or disables support for the PostGIS extension in PostgreSQL.
Multiple Connections	Opens parallel connections to a database.
Read Only	The connection is restricted to read-only operations.

PostgreSQL ODBC Connection String sample

```
DRIVER={Devart ODBC Driver for PostgreSQL};Data
Source=localhost;User
```

```
ID=postgres;Password=postgres;Database=postgres;Schema=public
```

See also:

- SSL Connection
- SSH Connection
- HTTP Tunneling

3.6 Connecting to Heroku Postgres

Connecting to Heroku Postgres using ODBC driver for PostgreSQL

Heroku is a cloud-based multilingual platform-as-a-service (PaaS) that is designed for hosting applications and web services, working with loaded applications, reducing the need for complex work with the server, and rapid scaling of applications. It also simplifies and speeds up the development cycle.

This article explains how to connect to the Heroku Postgres using our ODBC driver for PostgreSQL.

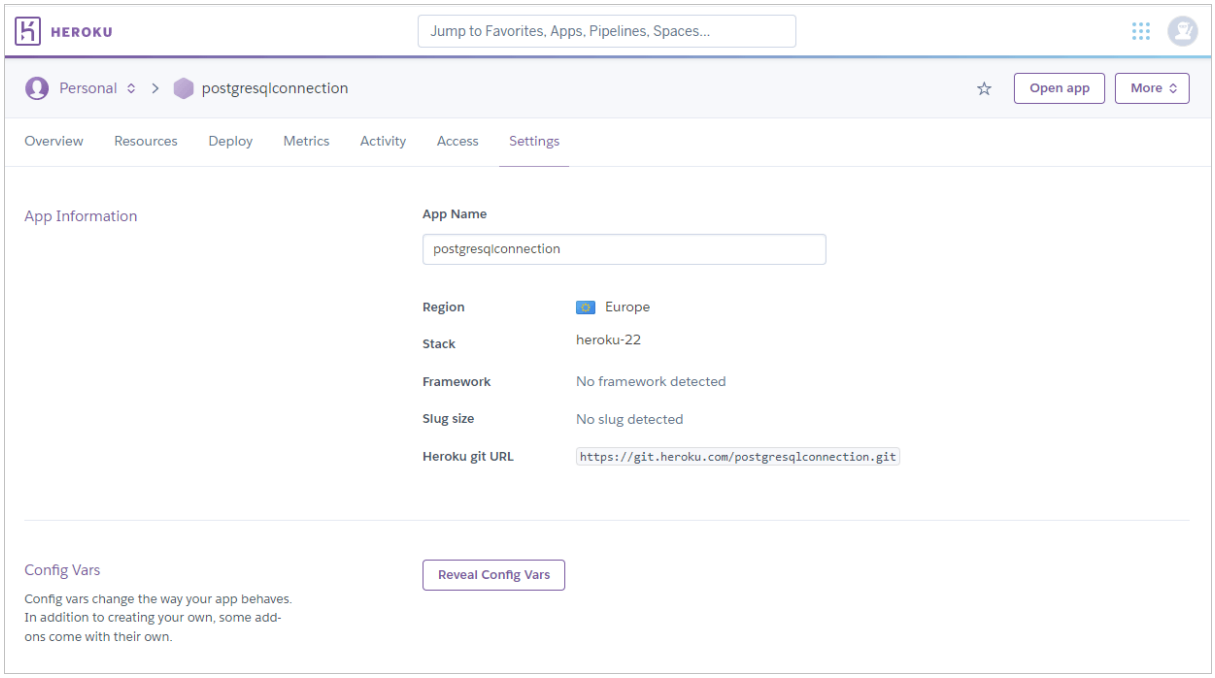
To connect to your Heroku Postgres database, you need to get the URL of your database. You may get it through the **Heroku Dashboard**.

Using the Heroku Dashboard

This method is easier if you are using the platform for the first time.

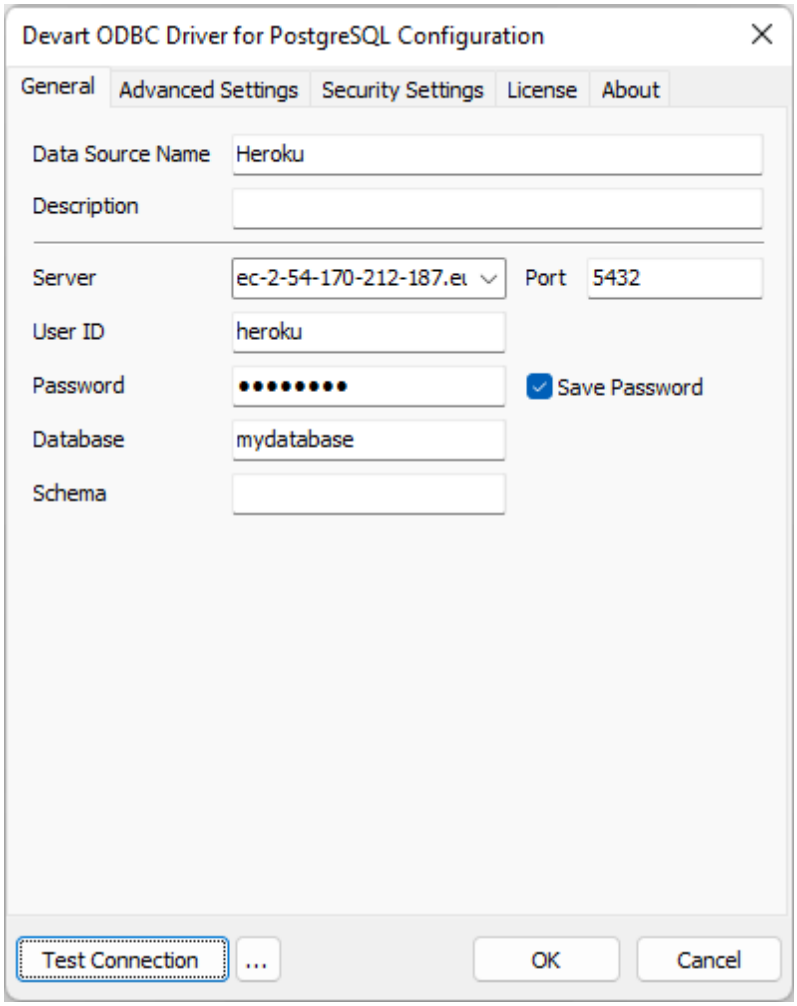
1. Go to the application on Heroku you want to connect to.
2. Switch to the **Settings** tab and press the **Reveal Config Vars** button. The list of configuration variables will appear.
3. In the **Key** field, you will see the key value in the following format:

```
< postgres: //< username >: < password >@< hostname/server >/< databasename
```

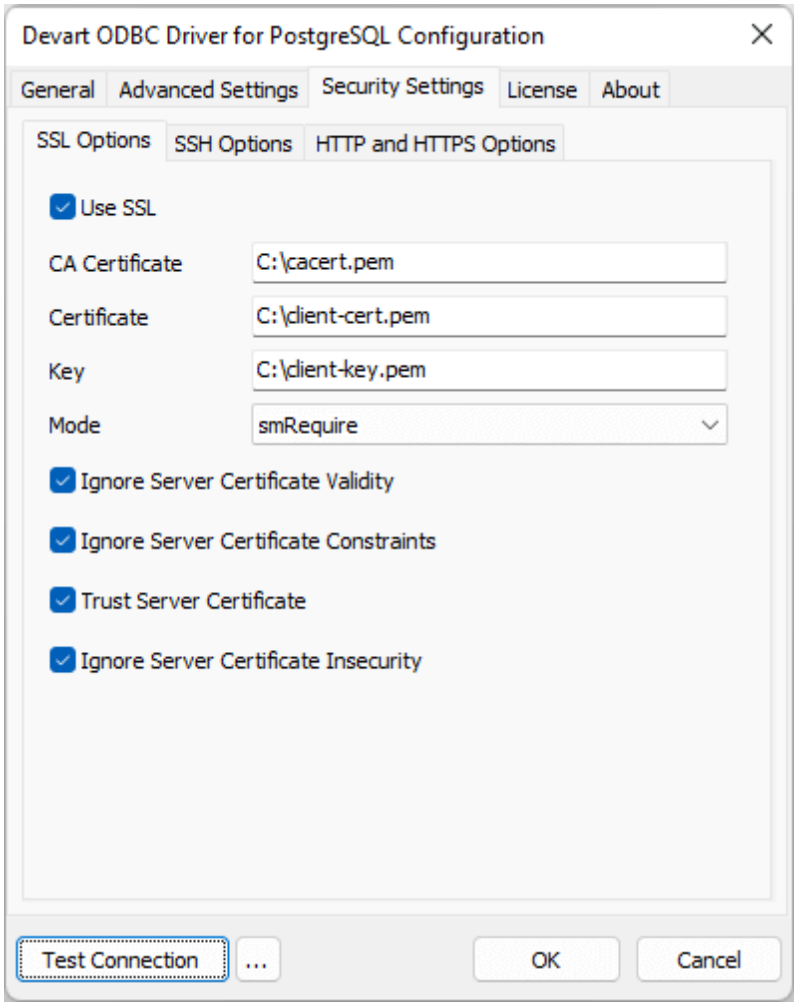


Now, you can establish a connection to Heroku Postgres through the ODBC Data Source Administrator. It is assumed that you have installed the ODBC driver for PostgreSQL on your computer.

1. Open the ODBC Data Source Administrator.
2. Select the **User DSN** and add an appropriate driver.
3. In the driver setup dialog, enter the necessary data that you got from the Heroku configuration value for DATABASE_URL.



4. Switch to the **Security Settings** > **SSL Options** and mark **Use SSL**.
5. To establish a secure connection you need to insert certificates. How to get and upload the certificate you can find on the Heroku SLL page.
6. After the certificate is downloaded, complete the configuration of the SSL options as seen in the picture below.



7. To test the connectivity, click the **Test Connection** button.

8. Click **Ok** to save the DSN.

3.7 Secure Connections

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

- SSL Connection
- SSH Connection
- HTTP Tunneling

3.7.1 SSL Connection

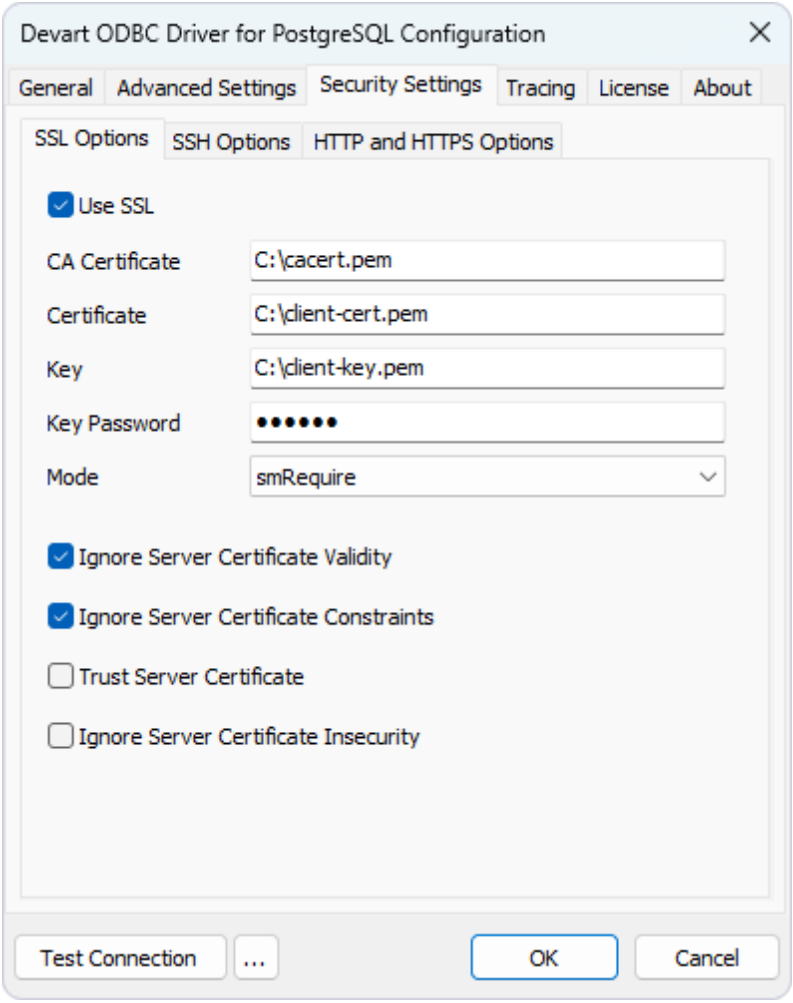
Connecting to PostgreSQL Using SSL

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 rarely 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 PostgreSQL due to its easier configuration and higher performance, compared to SSH.

See the PostgreSQL documentation for more information on how to secure TCP/IP connections with SSL .

To establish an SSL connection to PostgreSQL, specify the SSL certificates and client private key:



SSL Options

Option	Description
`Use SSL`	Enables SSL connections.
`CA Certificate`	The CA certificate.
`Certificate`	The client certificate.
`Key`	The private client key.
`Key Password`	The password for the private client key. Leave it empty if the client key is not encrypted.

`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`	Specifies whether to verify the security of the server certificate's signature during an SSL handshake. The default value is `False`.

`Mode`	Specifies whether and with what priority an SSL connection will be negotiated with the server.
	
	`smRequire`
	The default value. Only SSL connections are allowed.
	
	`smPrefer`
	Negotiates, first trying an SSL connection, then if it fails, tries a regular non-SSL connection.
	
	`smAllow`

	Negotiates, first trying a non-SSL connection, then if it fails, tries an SSL connection.
	
	`smVerifyCA`
	Verifies the server's identity by validating the server certificate chain up to the root certificate installed on the client machine.
	
	`smVerifyFull`
	Verifies the server's identity by validating the server certificate chain up to the root certificate installed on the client machine and checks that the server's hostname matches the details in the server certificate.

Sample SSL Connection String

```
DRIVER={Devart ODBC Driver for PostgreSQL};Data
Source=myServer;Port=myPort;Database=myDatabase;User
ID=myUsername;Password=myPassword;Schema=mySchema;Use SSL=True;SSL
CA Cert=C:\myCaCertificate.pem;SSL Cert=C:
\myClientCertificate.pem;Key=C:\myPrivateClientKey.pem
```

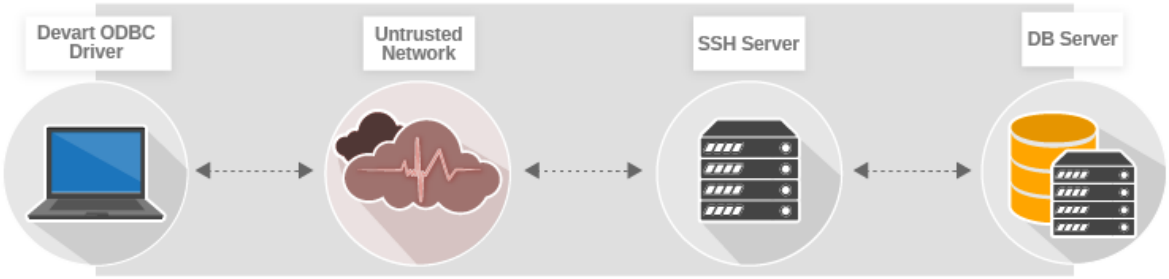
3.7.2 SSH Connection

Connecting to PostgreSQL Using SSH

This section discusses how to connect to PostgreSQL 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.

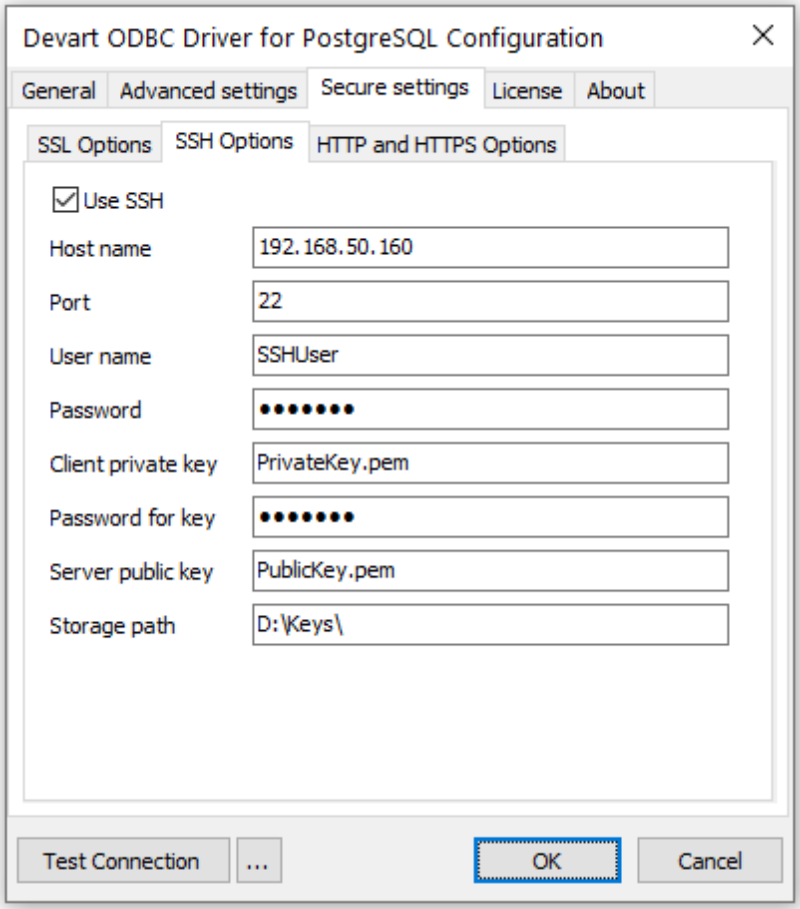
The SSH server listens on default port 22 (this port can be changed) for incoming TCP connections. The SSH client begins the initial TCP handshake with the server and verifies the server's identity. The client and server agree upon the encryption protocol and negotiate a session key. The server then authenticates the client and spawns the right environment. The ODBC driver for PostgreSQL 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 PostgreSQL. Below is a simplified diagram representing the SSH tunneling.



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

SSH Connection Options

To establish an SSH connection to PostgreSQL, 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.
`SSH Password`	The password for the account on the SSH server.

`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.
`SSH Storage Path`	The directory where the encryption keys are stored.

Sample Connection String:

```
DRIVER=Devart ODBC Driver for PostgreSQL;Data
Source=myHost;Database=myDatabase;User
ID=myUsername;Password=myPassword;Port=myPort;Schema=mySchema;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
```

3.7.3 HTTP Tunneling

Connecting to PostgreSQL Using HTTP Tunneling

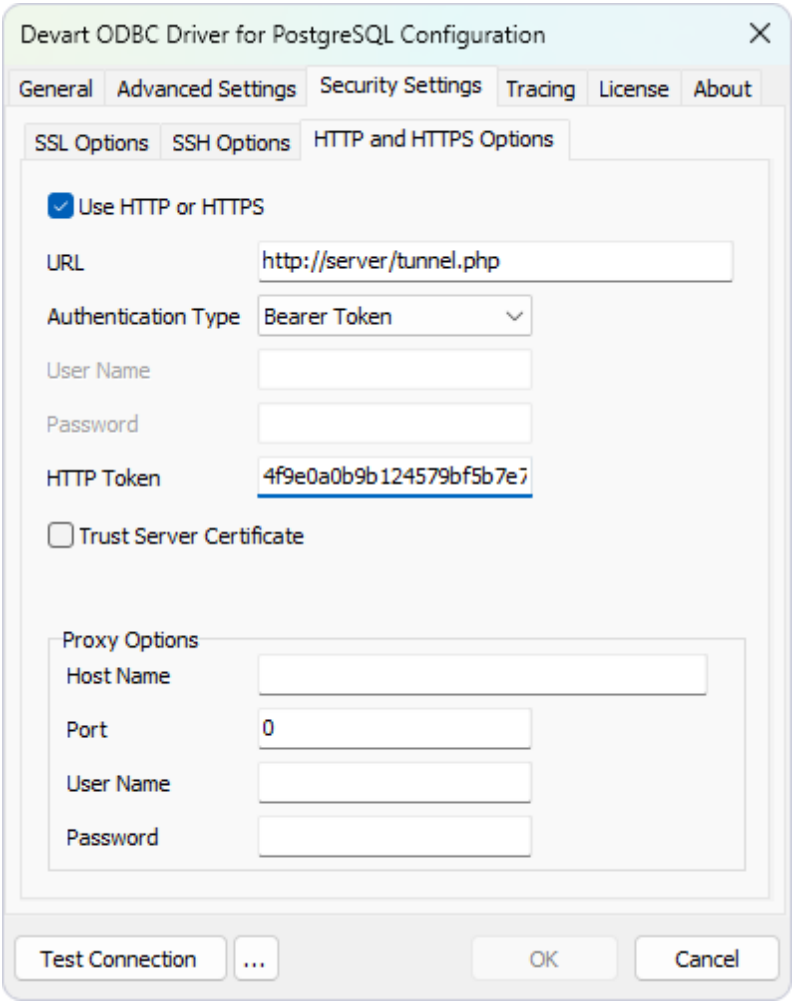
This section discusses how to connect the ODBC driver to PostgreSQL through an HTTP tunnel. If you need to connect to PostgreSQL in conditions of restricted connectivity, e.g. when a database server is hidden behind a firewall, or you need to transmit private network data through a public network, you 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 PostgreSQL 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\PostgreSQL \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.

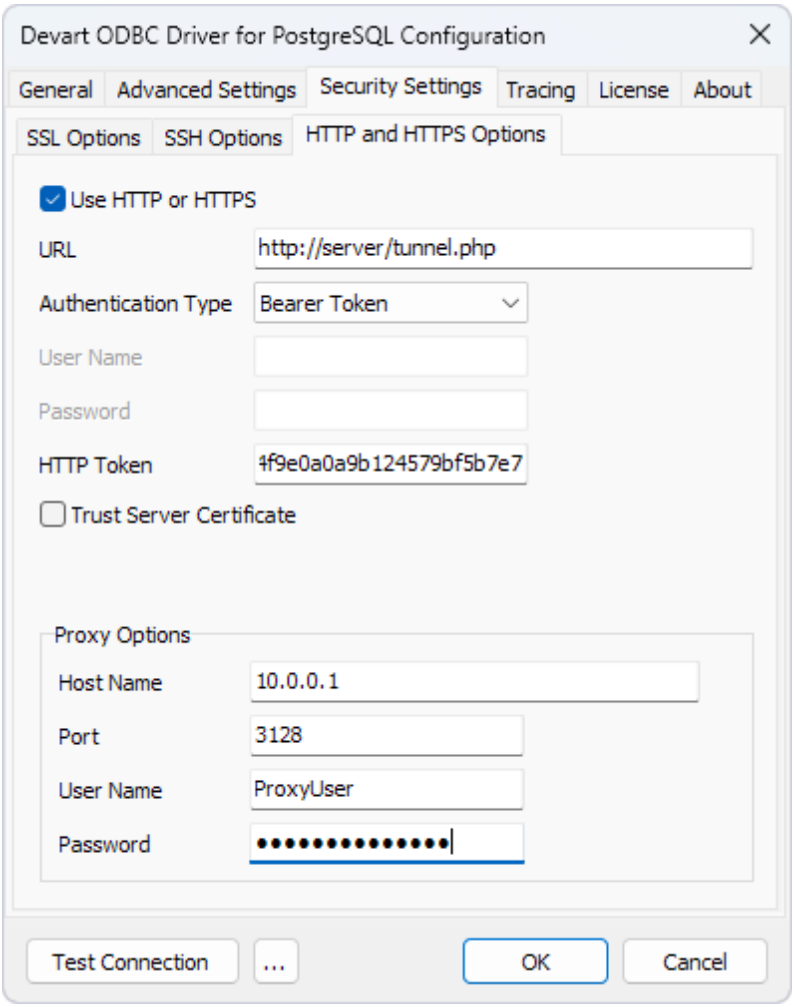
`Http User Name`	The username for the password-protected directory that contains the HTTP tunneling script.
`Http Password`	The password for the password-protected directory that contains the HTTP tunneling script.
`Http Trust Server Certificate`	Specifies whether to verify the server certificate during an SSL handshake. When `True`, the driver bypasses walking the certificate chain to verify the certificate. The default value is `False`.
`Http Token`	Stores a token for HTTP authorization. The `Token` property holds the Bearer token used to access the protected directory that contains the HTTP tunneling script.
`Http Authentication Type`	Specifies the HTTP authorization type. The `AuthenticationType` property specifies the HTTP authorization type used to access the secure directory that contains the HTTP tunneling script..

Sample Connection String Using HTTP Tunneling

```
DRIVER=Devart ODBC Driver for PostgreSQL;Data
Source=myHost;Database=myDatabase;User
ID=myUsername;Password=myPassword;Port=myPort;Schema=mySchema;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 User Name	The proxy username.
Proxy Password	The proxy password.

Sample Connection String Using HTTP Tunneling and Proxy Server

```
DRIVER=Devart ODBC Driver for PostgreSQL;Data
Source=myHost;Database=myDatabase;User
```

```
ID=myUsername;Password=myPassword;Port=myPort;Schema=mySchema;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 PostgreSQL.

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

3.8 Sandboxed Apps on macOS

Sandboxed Apps on macOS

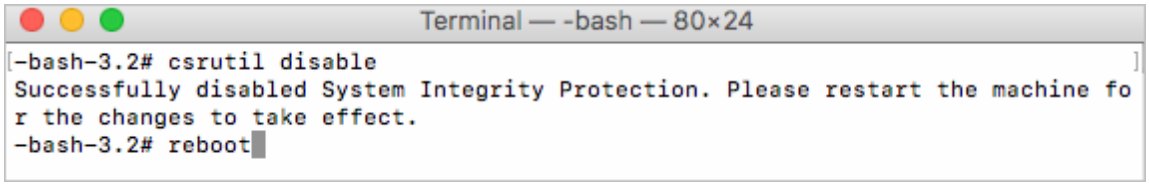
Sandboxed applications don't have permission to access iODBC Driver Manager on macOS. This is caused by the System Integrity Protection (SIP) technology on macOS which protects your files and folders from potentially malicious software by locking the application. When accessing a data source from an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

Note: All third-party applications distributed through the Mac App Store are sandboxed.

Disabling System Integration Protection (SIP) on macOS

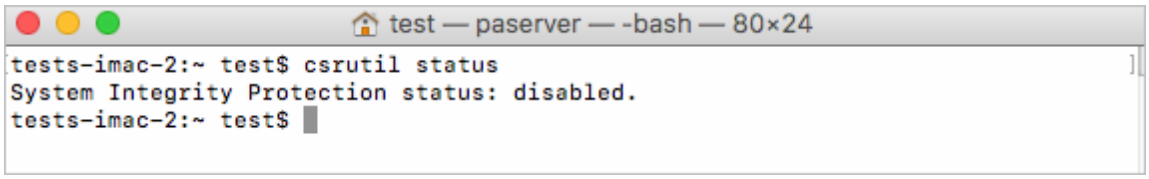
To resolve the issue, you should turn off SIP on your computer:

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`.



4. Restart your computer.

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



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

3.9 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.

3.10 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.

3.11 Usage Statistics

Usage Statistics

ODBC Driver for PostgreSQL can collect anonymous usage statistics. This data helps us improve product quality, resolve issues faster, and better understand how our products are used.

The collected data is anonymous and does not include personal information. The amount of transmitted data is minimal and is used only for diagnostic and product improvement purposes.

Collected Data

The driver collects the following data:

- Product name and version.
- Name of the process (application) using the driver.
- License information: license type, license number, and license status.
- Operating system name and version, number of processor cores.
- An anonymous user identifier.

> The user identifier is an internal ID generated only for statistical purposes. It is not the operating system user name and cannot be used to identify the actual user.

- An anonymous hardware identifier.

> The hardware identifier is an internal ID generated only for statistical purposes. It does not

contain any data that can identify specific hardware.

- Database server name and version.
- Names of connection parameters used to connect to the database server.

Only parameter names are collected. We do not collect parameter values such as database name, user name, or password.

- Connection result: success, or a numeric error code if the connection fails.

Only the numeric error codes are collected. We do not collect full error messages, which might contain sensitive data (for example, database or user names).

Default Settings

Usage statistics is enabled by default when you install the driver.

To disable usage statistics, follow the instructions for your operating system:

- Enable or Disable Usage Statistics on Windows
- Enable or Disable Usage Statistics on macOS
- Enable or Disable Usage Statistics on Linux

3.11.1 Enable or Disable on Windows

Enable or Disable Usage Statistics on Windows

Usage statistics is enabled by default when you install the driver. You can disable it in one of the following ways:

- **During installation**: In the installation wizard or from the command line.
- **After installation**: By editing the Windows Registry.

Disable Usage Statistics in the Installation Wizard

To disable usage statistics in the installation wizard, clear the **Improvement Program** checkbox on the last page of the wizard. The checkbox is selected by default.

Disable Usage Statistics From the Command Line

When you install the driver from the command line, you can disable usage statistics by adding the `/NOUSAGESTATISTICS` parameter to the command.

Silent and Very Silent Mode

To disable statistics during silent or very silent installation with the EXE installer, run one of the following commands:

```
DevartODBCPostgreSQL.exe /NOUSAGESTATISTICS /SILENT
DevartODBCPostgreSQL.exe /NOUSAGESTATISTICS /VERYSILENT
```

Quiet Mode

To disable statistics during quiet installation with the MSI installer, run the following command as an administrator:

```
msiexec /i DevartODBCPostgreSQL.msi /q NOUSAGESTATISTICS=true
```

Change Usage Statistics Settings in the Windows Registry

To enable or disable usage statistics for an installed driver, edit the Windows Registry as

follows:

1. Open the Registry Editor. To do this, press **Win+R**, type `regedit` in the **Run** dialog, and press **Enter**.
2. Depending on your driver version, navigate to one of the following keys:
   - 64-bit driver: `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Devart ODBC Driver for PostgreSQL`
   - 32-bit driver: `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ODBC\ODBCINST.INI\Devart ODBC Driver for PostgreSQL`
3. Set the value of the `UsageStatistics` parameter to `False` to disable statistics, or `True` to enable statistics.

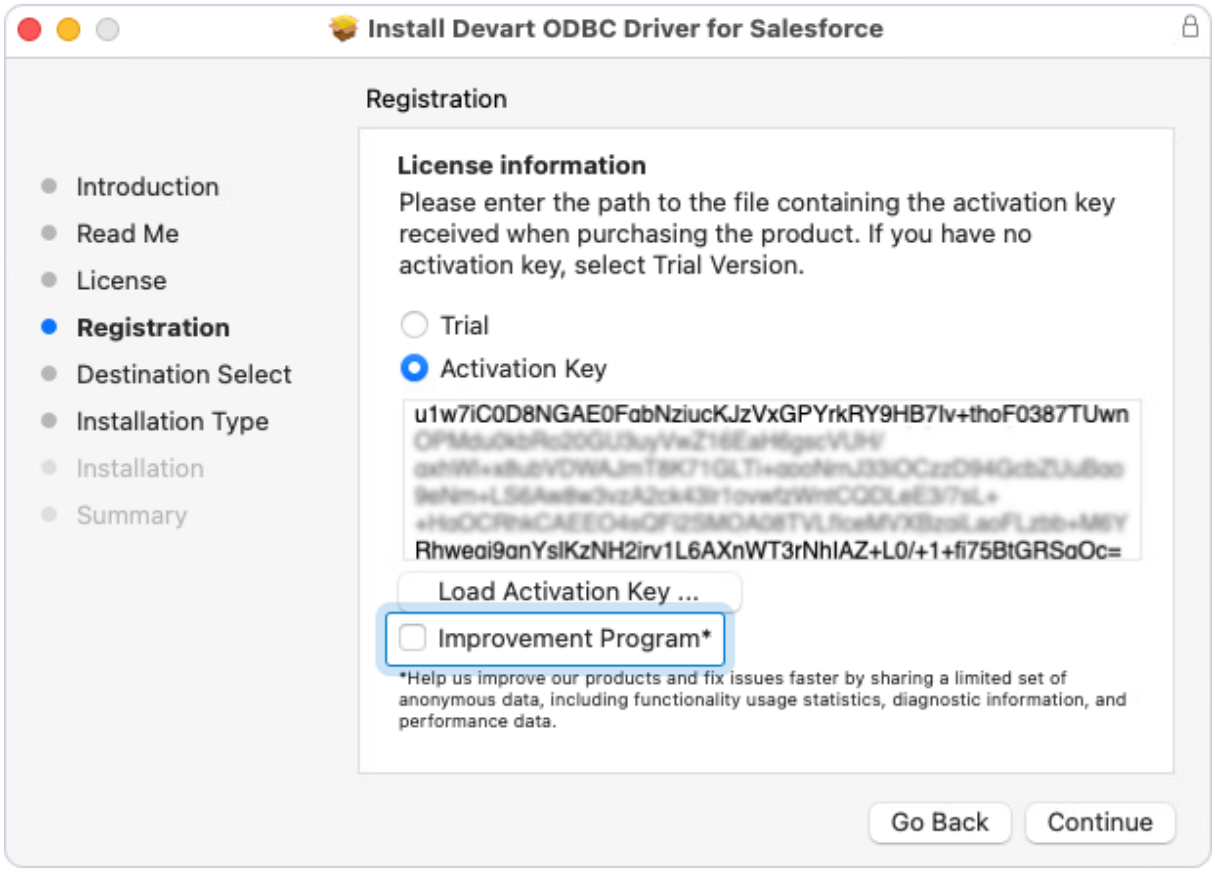
3.11.2 Enable or Disable on macOS

Enable or Disable Usage Statistics on macOS

Usage statistics is enabled by default when you install the driver. You can disable it in the installation wizard or later using a console application.

Disable Usage Statistics in the Installation Wizard

To disable usage statistics in the installation wizard, on the **Registration** page, clear the **Improvement Program** checkbox.



Enable or Disable Usage Statistics in a Console Application

To enable or disable usage statistics using a console application:

1. In the console, go to the folder where the driver was installed. The default installation path for the driver is `/Library/ODBC/Devart/PostgreSQL`.
2. Run the activation command with superuser privileges using the `-u` option. Set the value to `false` to disable usage statistics or `true` to enable it.
   - To disable usage statistics: `sudo ./postgresqlodbcactivator -u false`
   - To enable usage statistics: `sudo ./postgresqlodbcactivator -u true`

3.11.3 Enable or Disable on Linux

Enable or Disable Usage Statistics on Linux

Usage statistics is enabled by default when you install the driver. The graphical installer

doesn't provide an option to disable usage statistics. You can disable statistics during package installation or after installation using a console application.

Disable Usage Statistics During Package Installation

To disable usage statistics when installing a DEB or RPM package, set the `NOUSAGESTATISTICS` environment variable to `true`.

DEB Package

To disable usage statistics when installing a DEB package, run the following command:

```
sudo NOUSAGESTATISTICS=true dpkg -i devartodbcpostgresql.deb
```

RPM Package

To disable usage statistics when installing an RPM package, run the following command:

```
sudo NOUSAGESTATISTICS=true rpm -ivh devartodbcpostgresql.rpm
```

Enable or Disable Usage Statistics After Installation

To enable or disable usage statistics for an installed driver, use a console application.

1. In the console, go to the folder where the driver was installed. The default installation path is:
   - DEB package: `/usr/share/devart/odbcpostgresql`
   - RPM package: `/usr/local/devart/odbcpostgresql`
2. Run the activation command with superuser privileges using the `-u` option. Set the value to `false` to disable usage statistics or `true` to enable it.
   - To disable usage statistics:

```
sudo ./postgresqlodbcactivator -u false -i /etc
```

   - To enable usage statistics:

```
sudo ./postgresqlodbcactivator -u true -i /etc
```

3.12 Supported Data Types

Data Type Mapping

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

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

PostgreSQL Data Types	ODBC Data Types
TEXT	SQL_LONGVARCHAR
CHARACTER VARYING	SQL_VARCHAR
CHARACTER	SQL_CHAR
BIT	SQL_BIT
SMALLINT	SQL_SMALLINT
INTEGER	SQL_INTEGER
BIGINT	SQL_BIGINT
OID	SQL_INTEGER
SMALLSERIAL	SQL_SMALLINT
SERIAL	SQL_INTEGER
BIGSERIAL	SQL_BIGINT
BYTEA	SQL_LONGVARBINARY
NUMERIC	SQL_NUMERIC
DECIMAL	SQL_DECIMAL
REAL	SQL_REAL
DOUBLE PRECISION	SQL_DOUBLE
DATE	SQL_TYPE_DATE
TIME	SQL_TYPE_TIME
TIMESTAMP	SQL_TYPE_TIMESTAMP
INTERVAL YEAR	SQL_INTERVAL_YEAR
INTERVAL MONTH	SQL_INTERVAL_MONTH
INTERVAL DAY	SQL_INTERVAL_DAY
INTERVAL HOUR	SQL_INTERVAL_HOUR
INTERVAL MINUTE	SQL_INTERVAL_MINUTE
INTERVAL SECOND	SQL_INTERVAL_SECOND

INTERVAL YEAR TO MONTH	SQL_INTERVAL_YEAR_TO_MONTH
INTERVAL DAY TO HOUR	SQL_INTERVAL_DAY_TO_HOUR
INTERVAL DAY TO MINUTE	SQL_INTERVAL_DAY_TO_MINUTE
INTERVAL DAY TO SECOND	SQL_INTERVAL_DAY_TO_SECOND
INTERVAL HOUR TO MINUTE	SQL_INTERVAL_HOUR_TO_MINUTE
INTERVAL HOUR TO SECOND	SQL_INTERVAL_HOUR_TO_SECOND
INTERVAL MINUTE TO SECOND	SQL_INTERVAL_MINUTE_TO_SECOND

3.13 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 PostgreSQL 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.

4 Using in Third-Party Tools

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

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

4.1 Using in DBeaver

This section describes how to connect DBeaver to PostgreSQL using Devart ODBC Driver for PostgreSQL.

- Connect DBeaver Community to PostgreSQL through ODBC
- Connect DBeaver Enterprise to PostgreSQL through ODBC

4.1.1 Connect DBeaver Community to PostgreSQL through ODBC

DBeaver Community and DBeaver Enterprise let users connect to PostgreSQL via ODBC, enabling SQL-based querying, reporting, and data management.

If you need basic ODBC connectivity to PostgreSQL and are comfortable with manual configuration using a generic ODBC Connection, choose DBeaver Community—a free, open-source database management tool.

If you require a simplified connection setup with built-in ODBC support, enhanced security, and performance features, you may try DBeaver Enterprise. For more information on connecting to PostgreSQL data from DBeaver Enterprise, see Connect DBeaver Enterprise to PostgreSQL through ODBC.

Initial configuration

1. Download `jdbc-odbc-bridge-jre7.jar` and `x64/JdbcOdbc.dll` from Github.

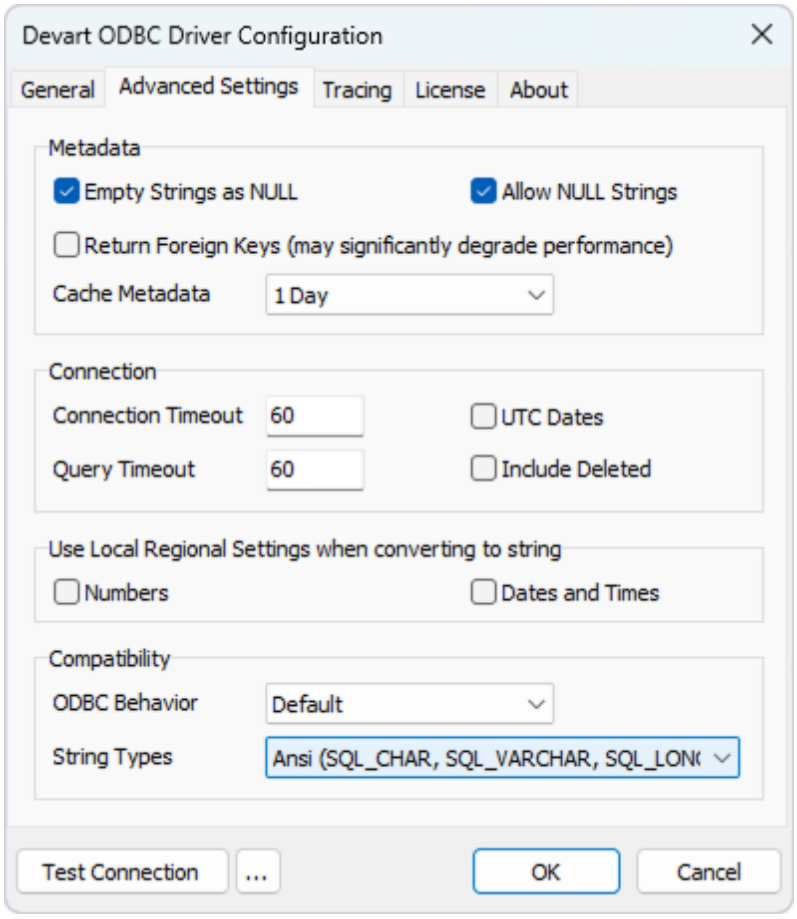
2. Download the **Microsoft Visual C++ 2010 Service Pack 1 Redistributable Package** from the Microsoft website.

The built-in legacy ODBC driver was removed in DBeaver Community Edition 23.1. If you're using an earlier version of DBeaver Community, skip steps 1 and 2.

3. Configure an ODBC data source. For more information, see Windows DSN Configuration.

4. On the **Advanced Settings** tab of the DSN configuration window, select **Ansi** from the **String Types**.

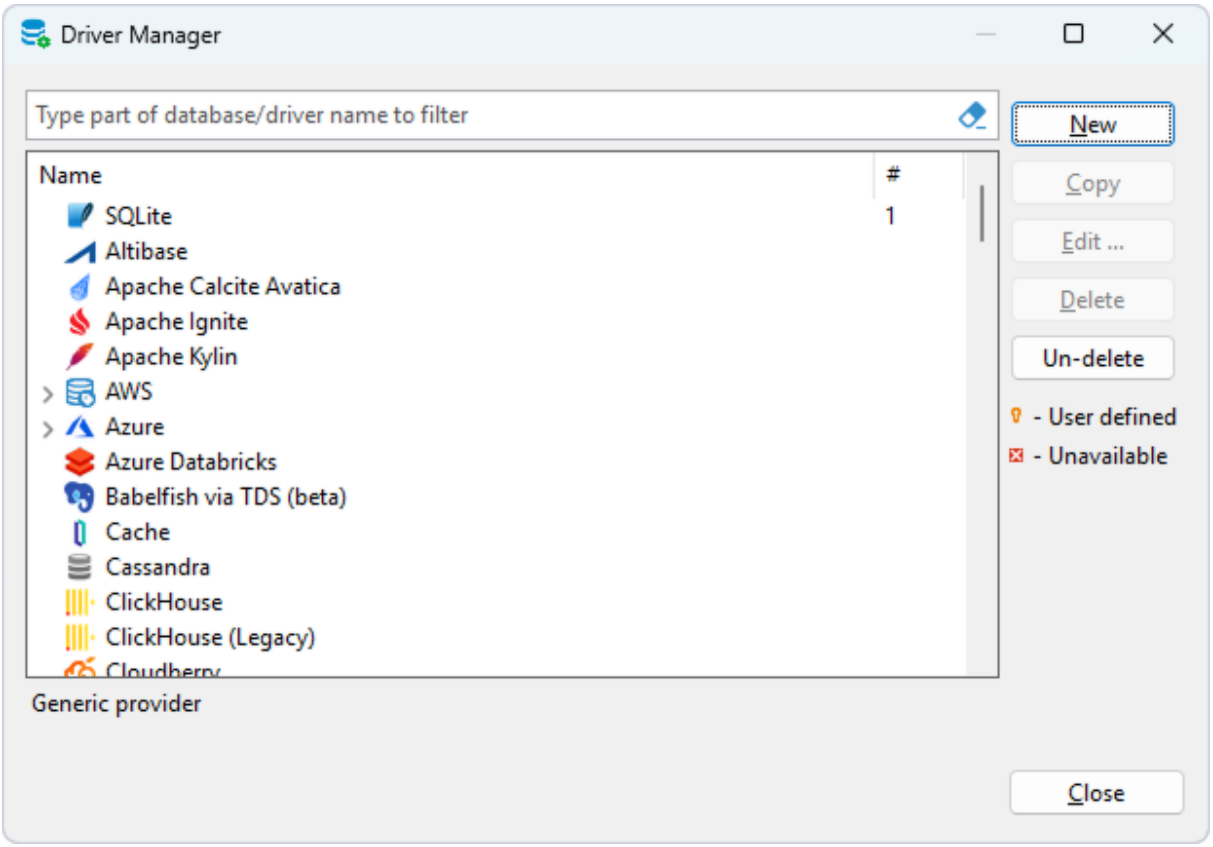
This option is required for the proper display of the SQL_WVARCAHAR data type in DBeaver. It also ensures that all string types will be returned as **SQL_CHAR**, **SQL_VARCHAR**, and **SQL_LONGVARCHAR**.



Connect to PostgreSQL

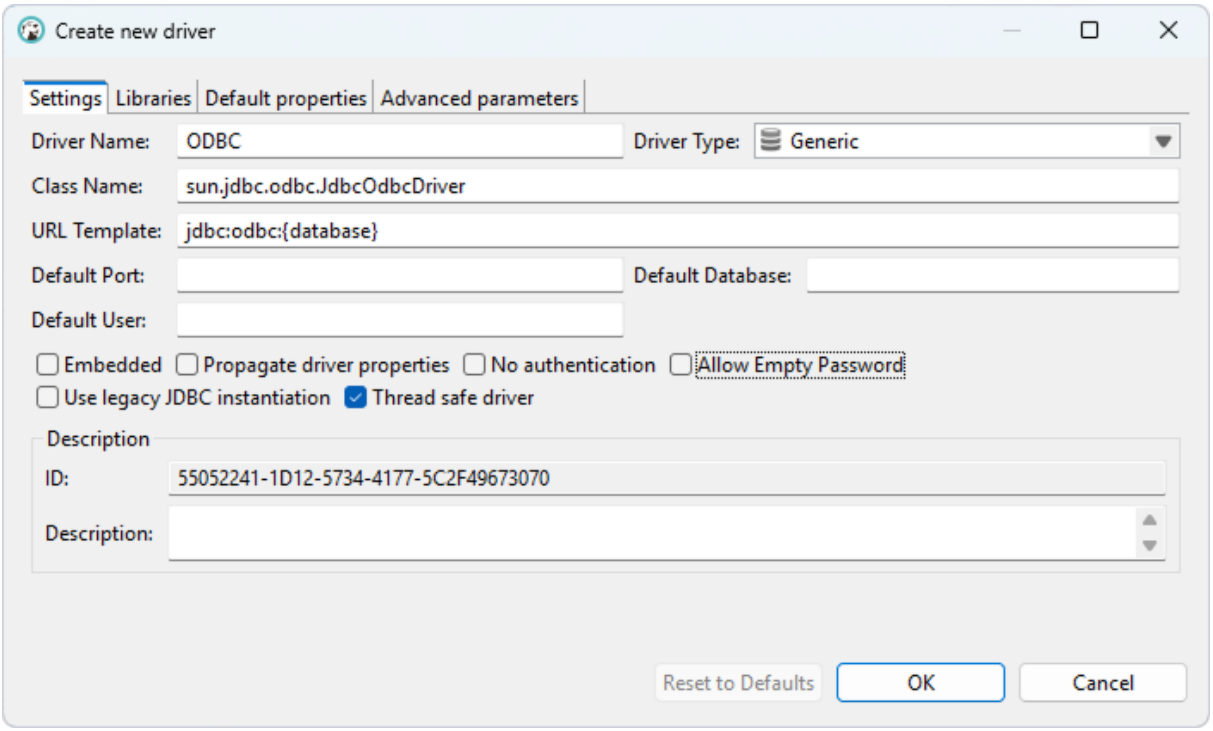
1. In DBeaver, select **Database** > **Driver Manager**.

2. Click **New**.



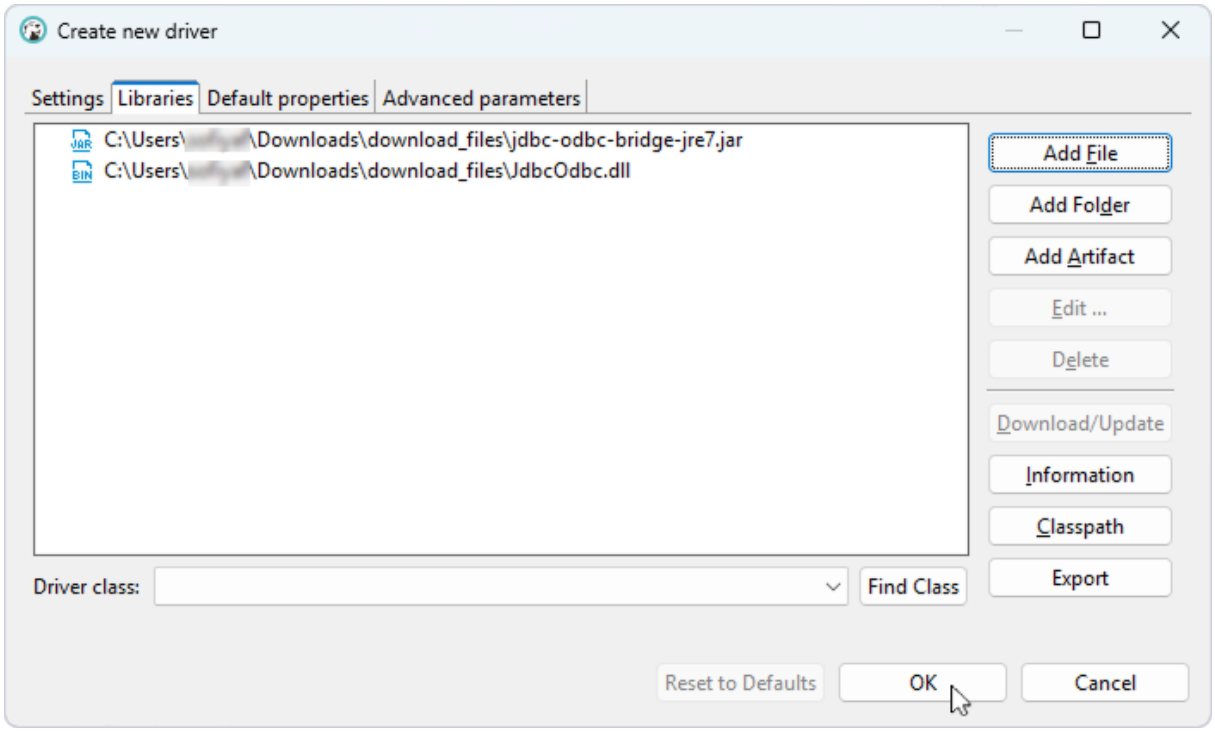
3. Configure the following properties for a new driver:

- In the **Driver Name** field, enter *ODBC*.
- In the **Class Name** field, enter *sun.jdbc.odbc.JdbcOdbcDriver*
- In the **URL Template** field, select *jdbc:odbc:{database}*.

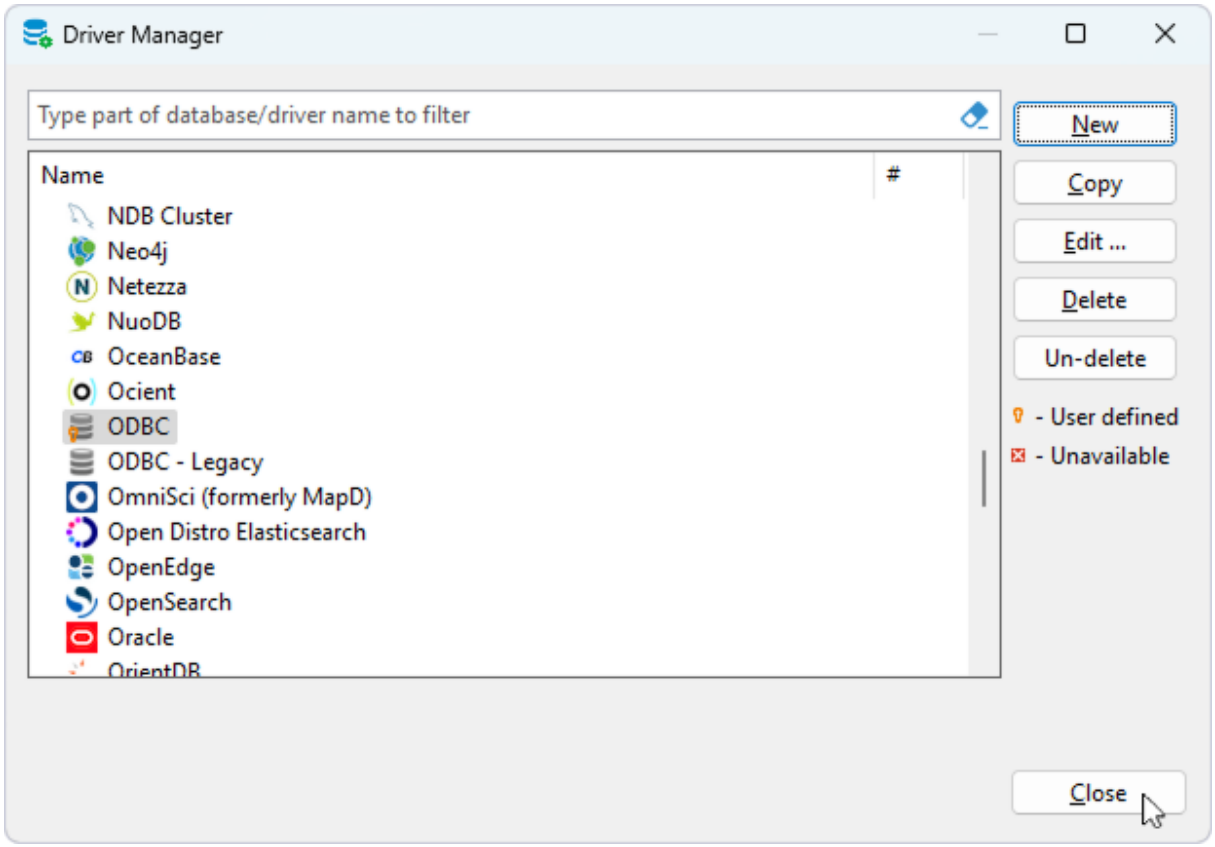


4. On the **Libraries** tab, click **Add File**.

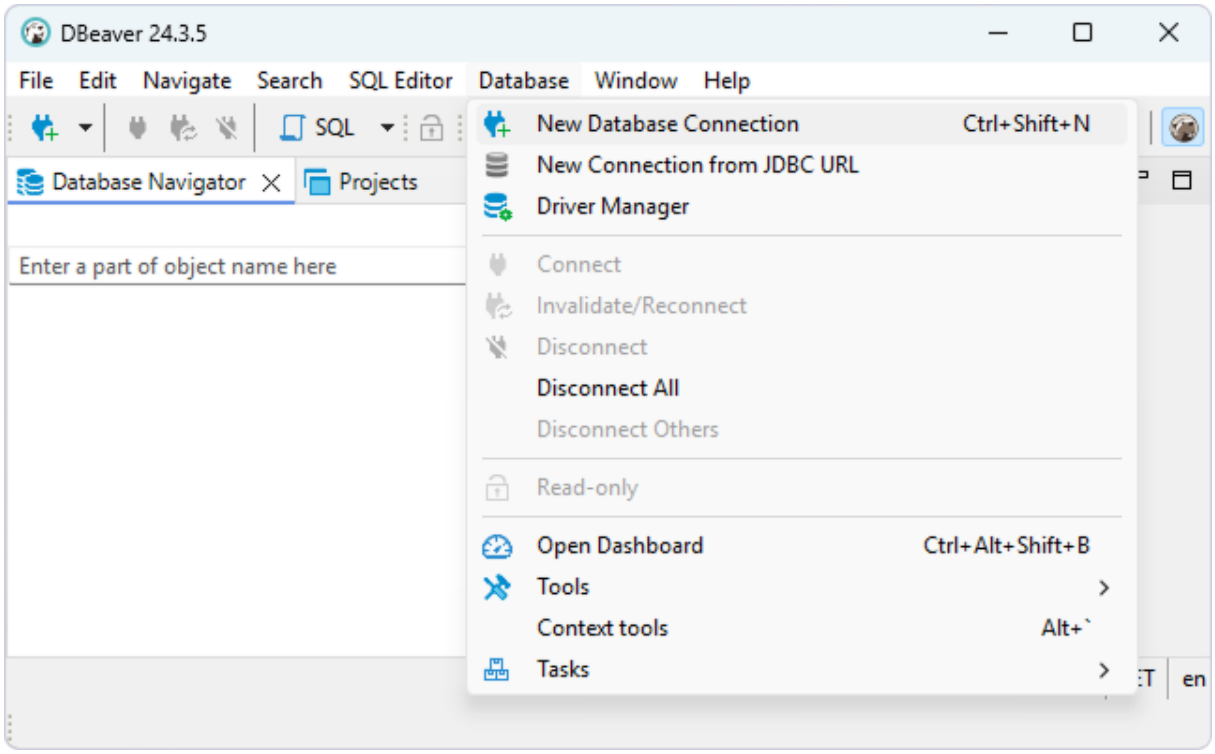
5. Select the `jdbc-odbc-bridge-jre7.jar`, then click **OK**. After that, select `JdbcOdbc.dll`, then click **OK**.



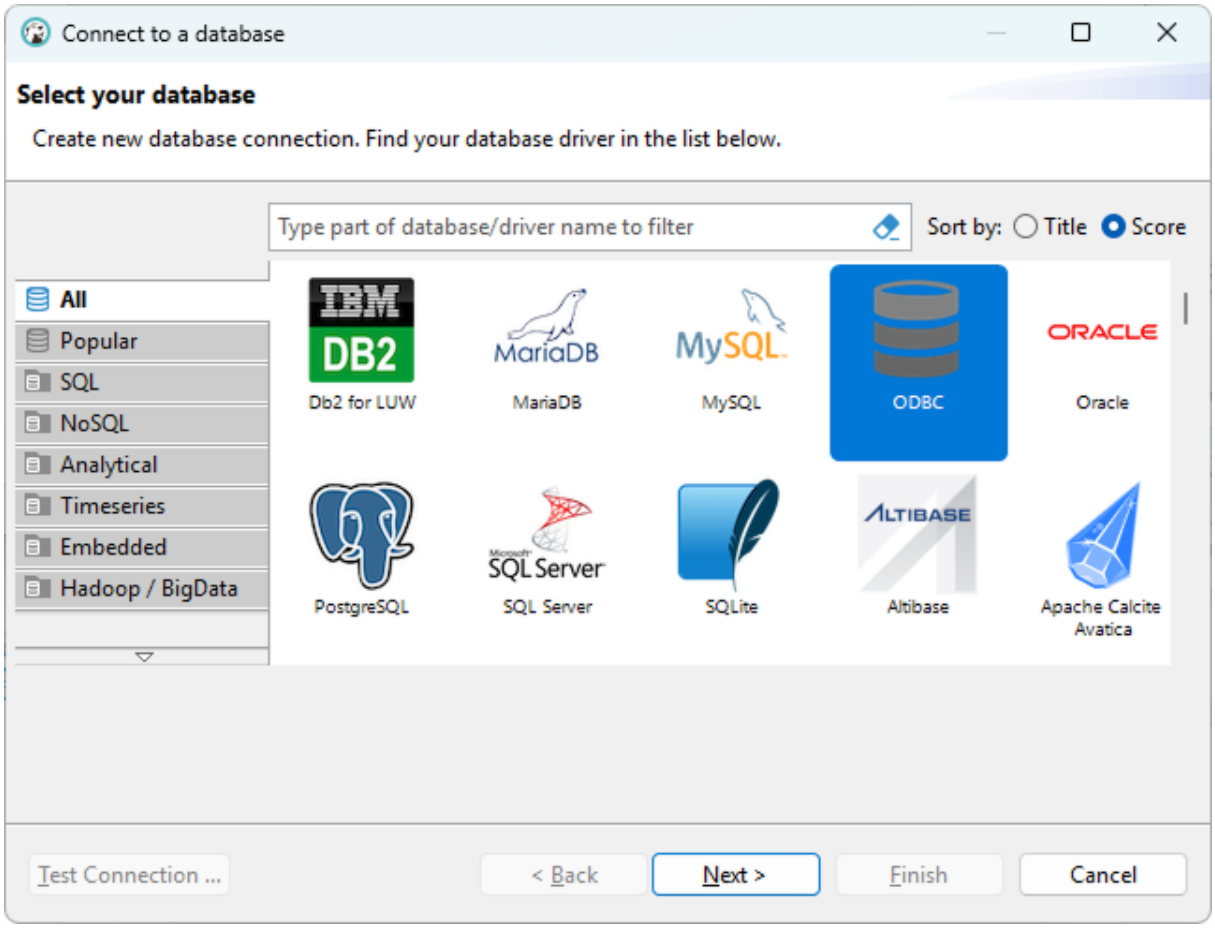
6. Once a new ODBC driver appears on the list, click **Close**.



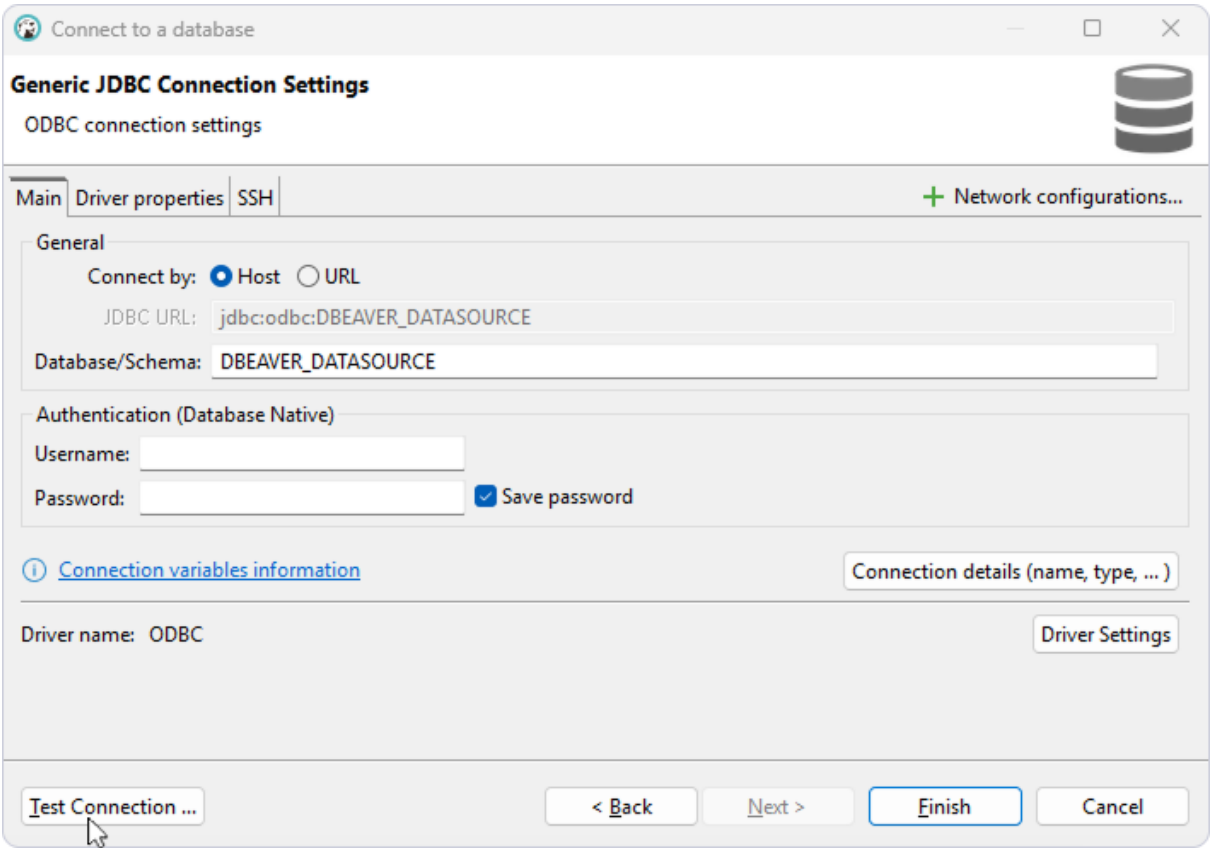
7. Select **Database** > **New Database Connection**.



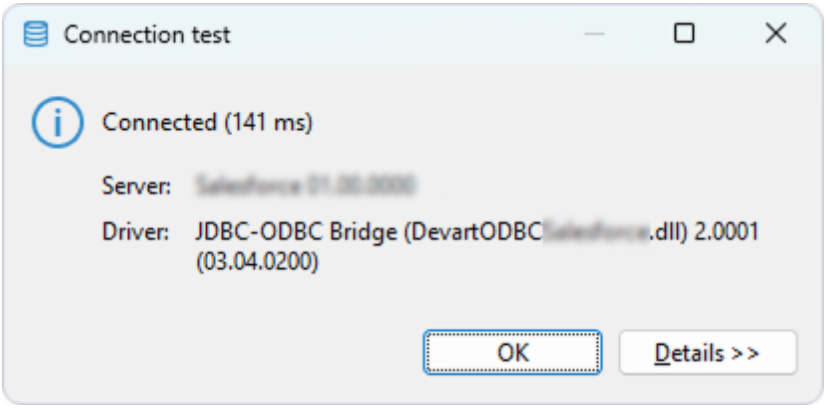
8. Select the **ODBC** driver, then click **Next**.



9. In the **Database/Schema** field, specify the name of your DSN.



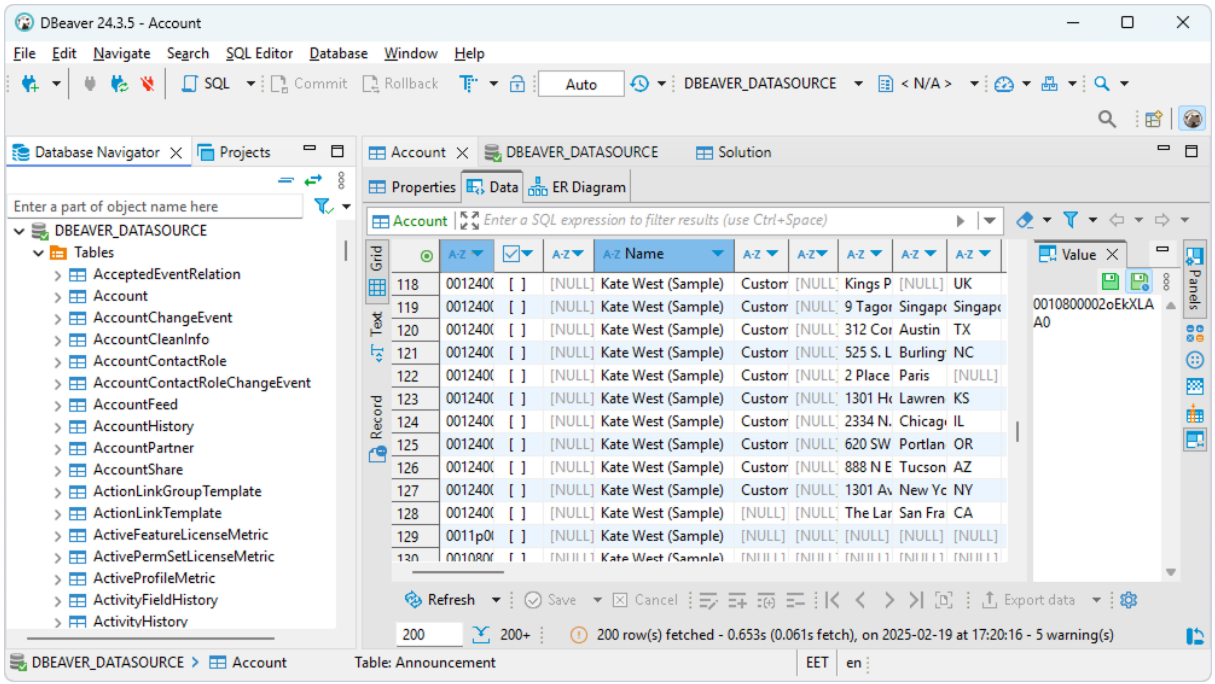
10. Optional: Select **Test Connection** to verify the connection settings.



11. Click **Finish**.

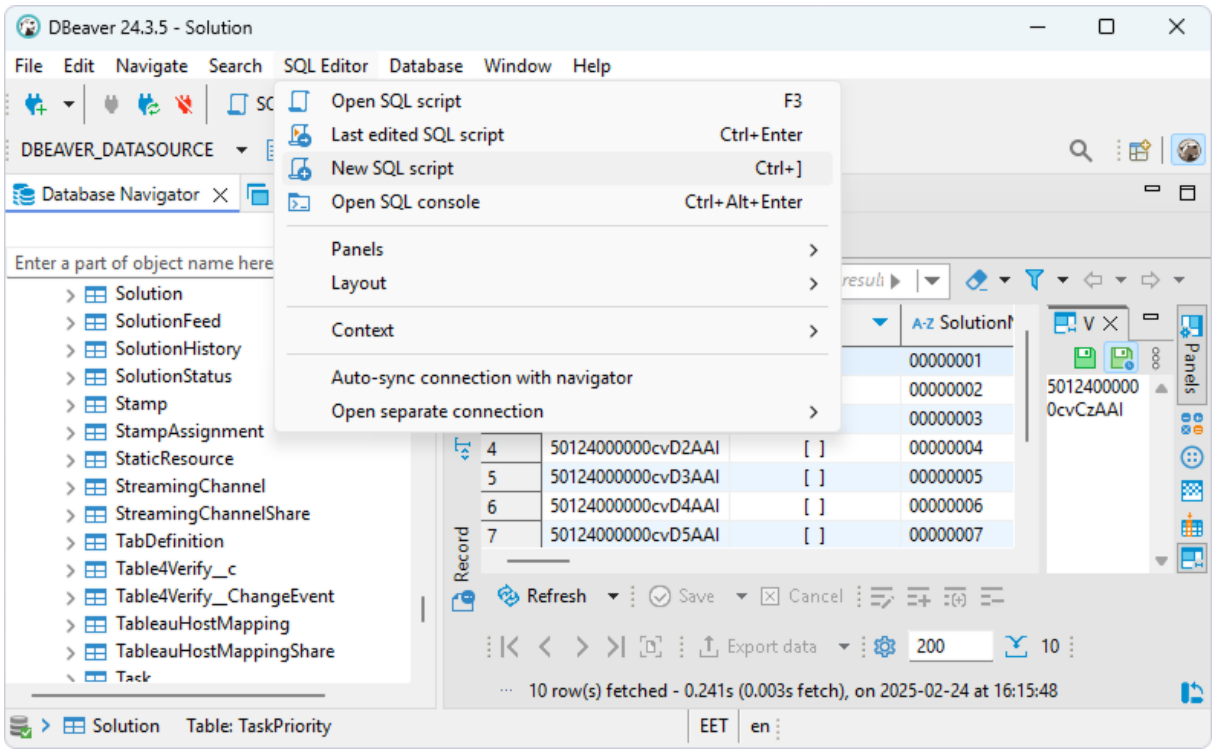
The database appears on the left pane.

12. To view the data stored in a table, expand the database structure and click the needed table.

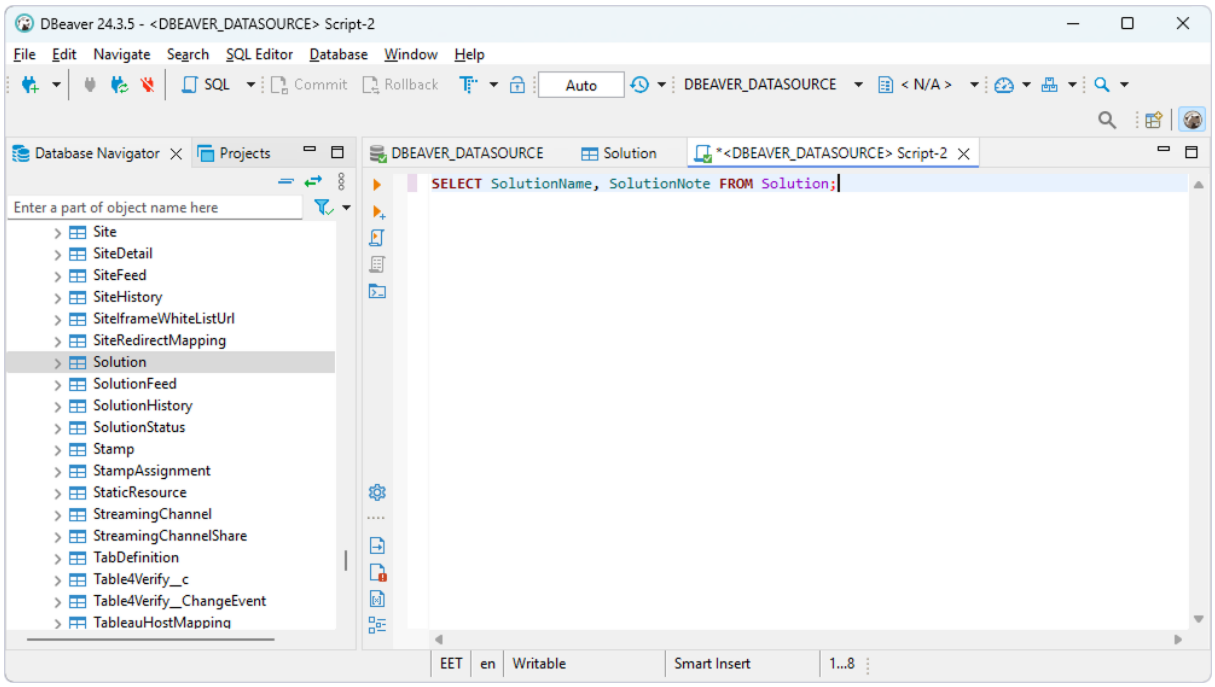


Query PostgreSQL data

1. Select **SQL Editor** > **New SQL script**.

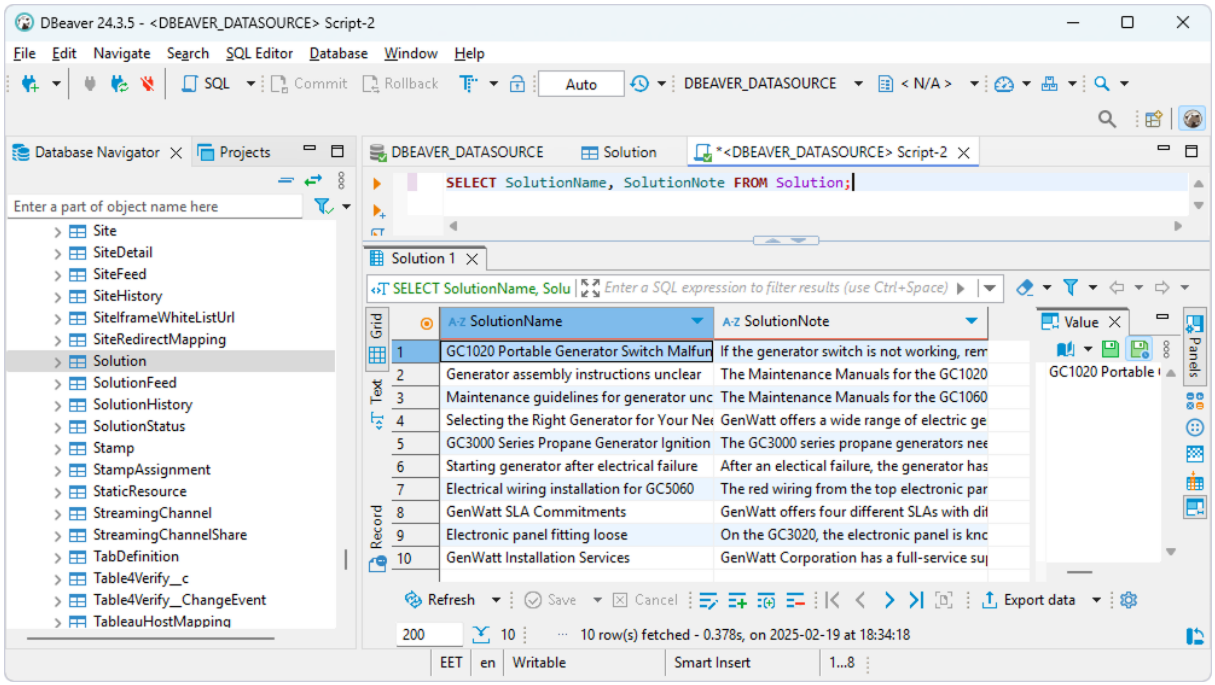


2. Enter your query.



3. Select **SQL Editor** > **Execute SQL query**.

The query results are displayed in the main window.



4.1.2 Connect DBeaver Enterprise to PostgreSQL through ODBC

DBeaver Enterprise and DBeaver Community let users connect to PostgreSQL via ODBC, enabling SQL-based querying, reporting, and data management.

If you require a simplified connection setup with built-in ODBC support, enhanced security, and performance features, you may try DBeaver Enterprise.

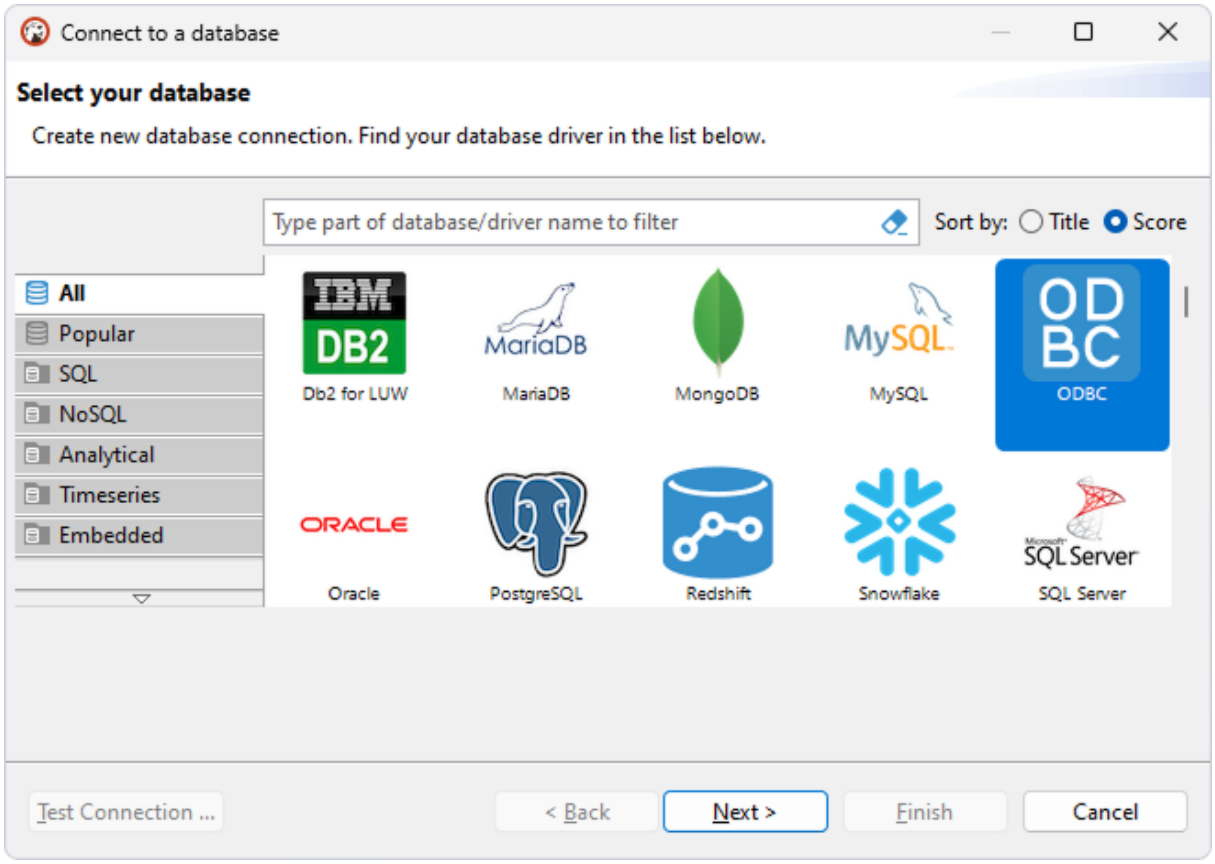
If you need basic ODBC connectivity to PostgreSQL and are comfortable with manual configuration using a generic ODBC connection, choose DBeaver Community—a free, open-source database management tool. For more information on connecting to PostgreSQL data from DBeaver Community, see Connect DBeaver Community to PostgreSQL through ODBC.

Connect to PostgreSQL

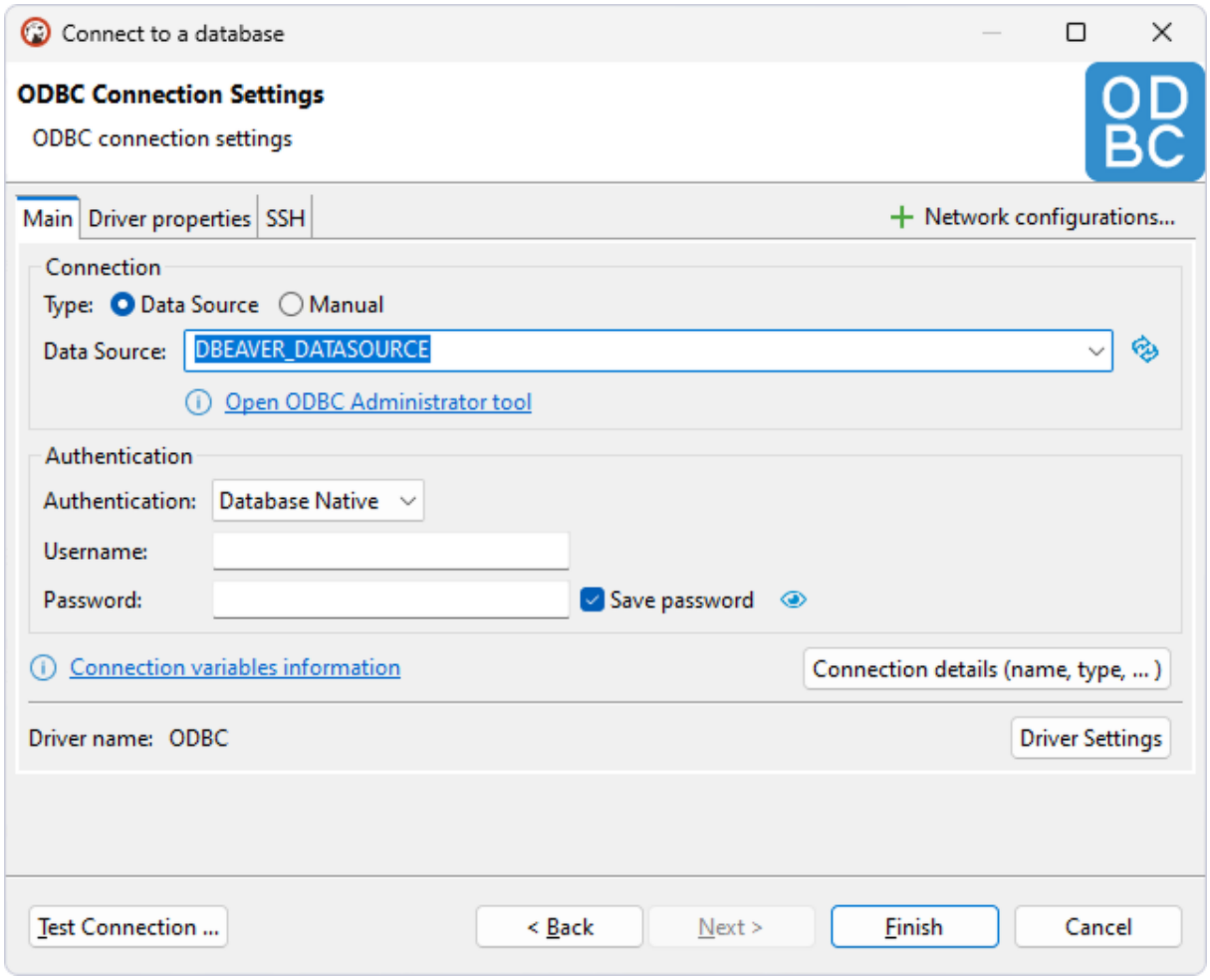
To connect to the PostgreSQL database from DBeaver Enterprise:

1. Select **Database** > **New Database Connection**.

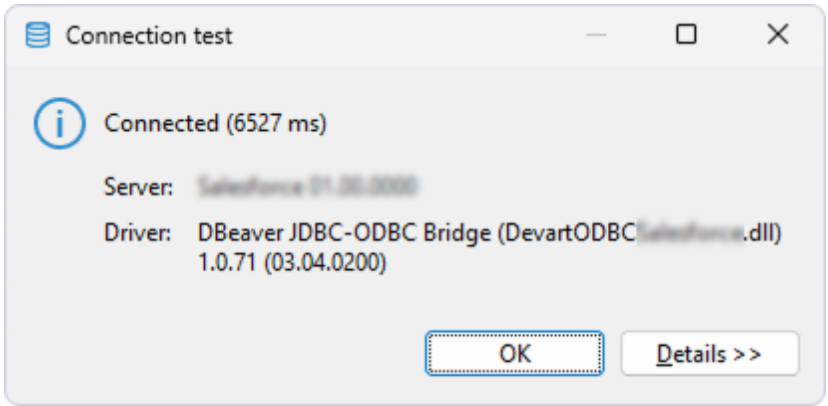
2. Select the **ODBC** driver and click **Next**.



3. In the **Database Source** field, specify the name of your DSN.



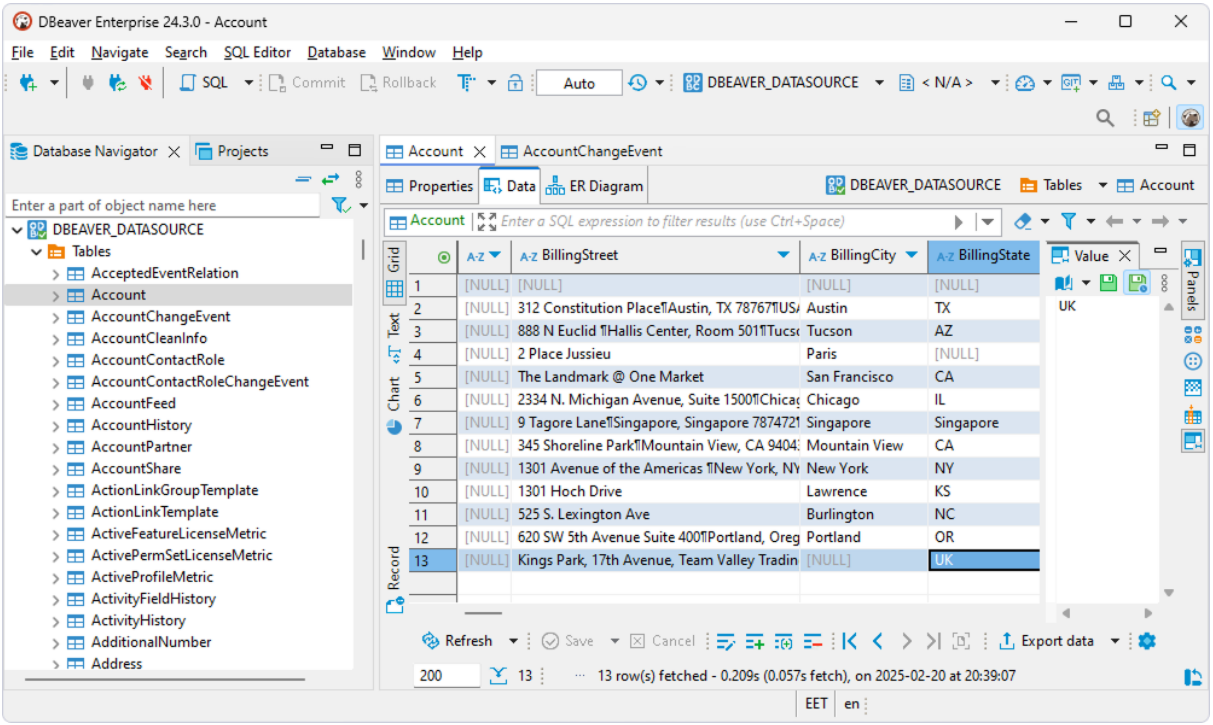
4. Optional: Select **Test Connection** to verify the connection settings.



5. Click **Finish**.

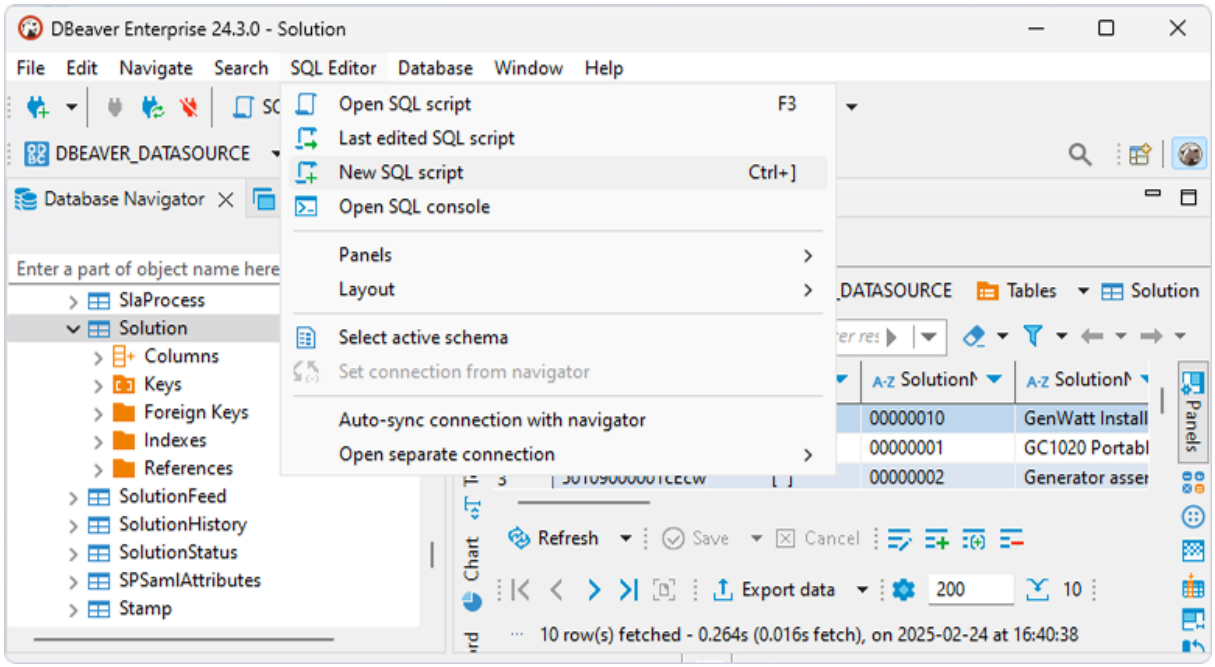
The database appears on the left pane.

6. To view the data stored in a table, expand the database structure and click the needed table.

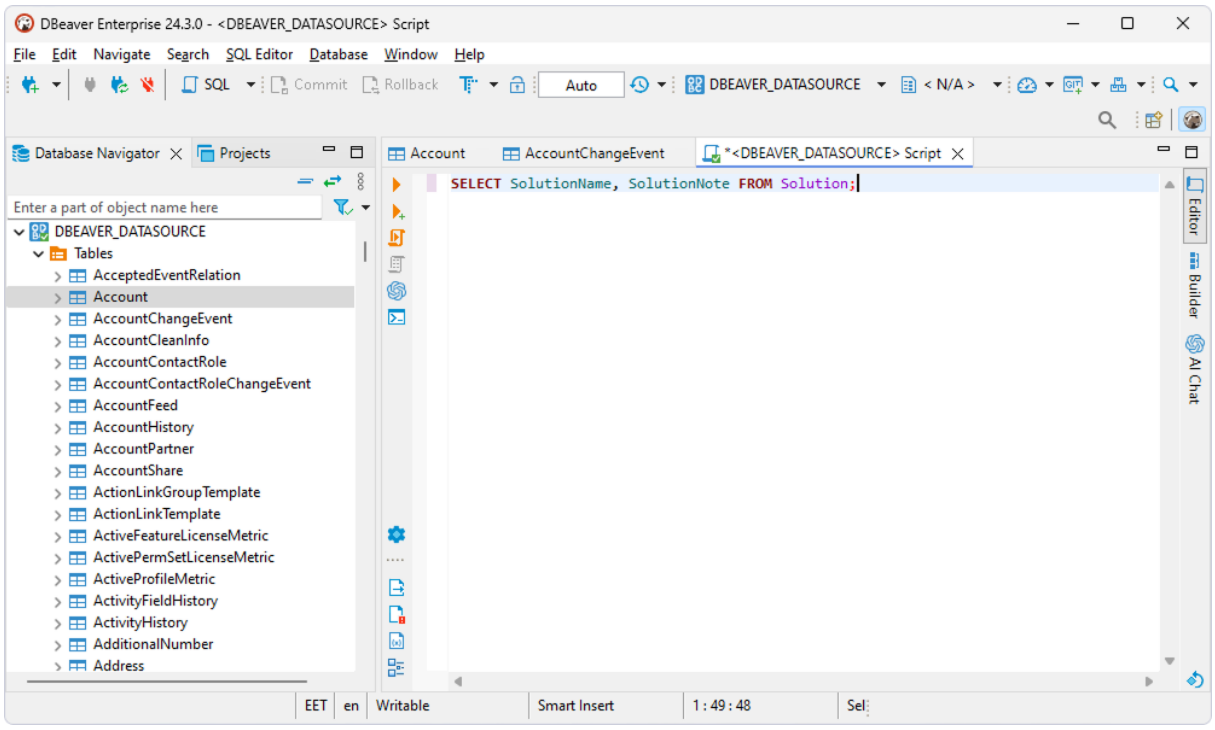


Query PostgreSQL data

1. Select **SQL Editor** > **New SQL script**.

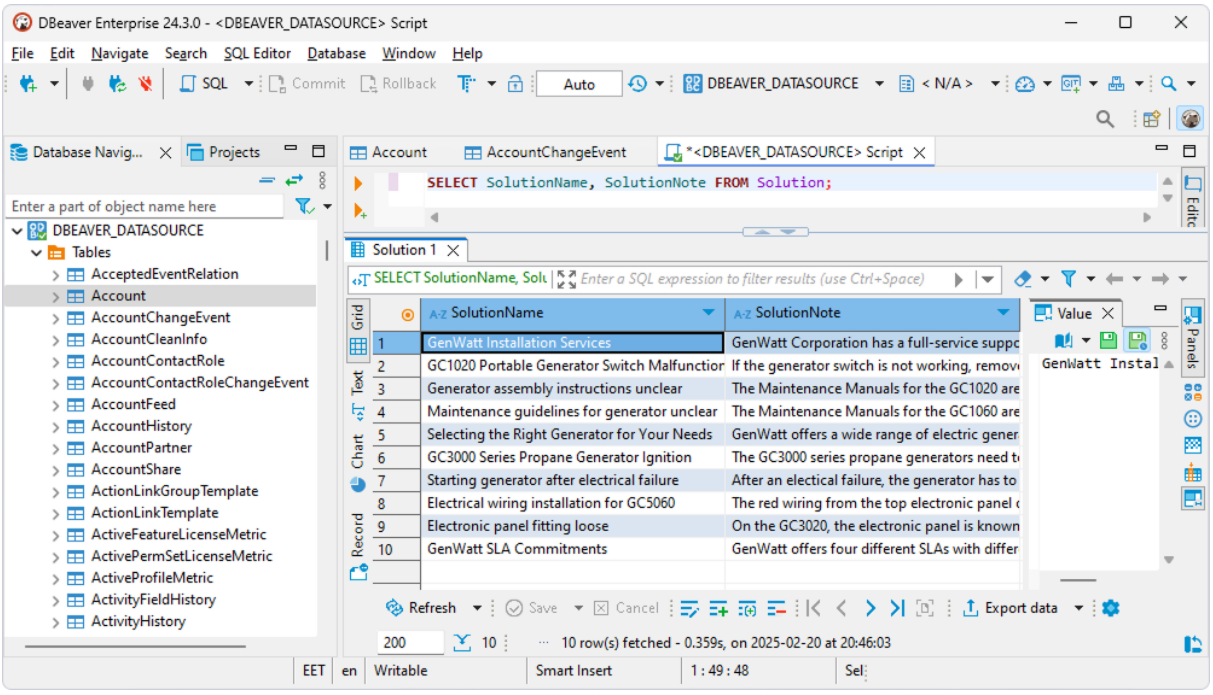


2. Enter your query.



3. Select **SQL Editor** > **Execute SQL query**.

The query results are displayed in the main window.

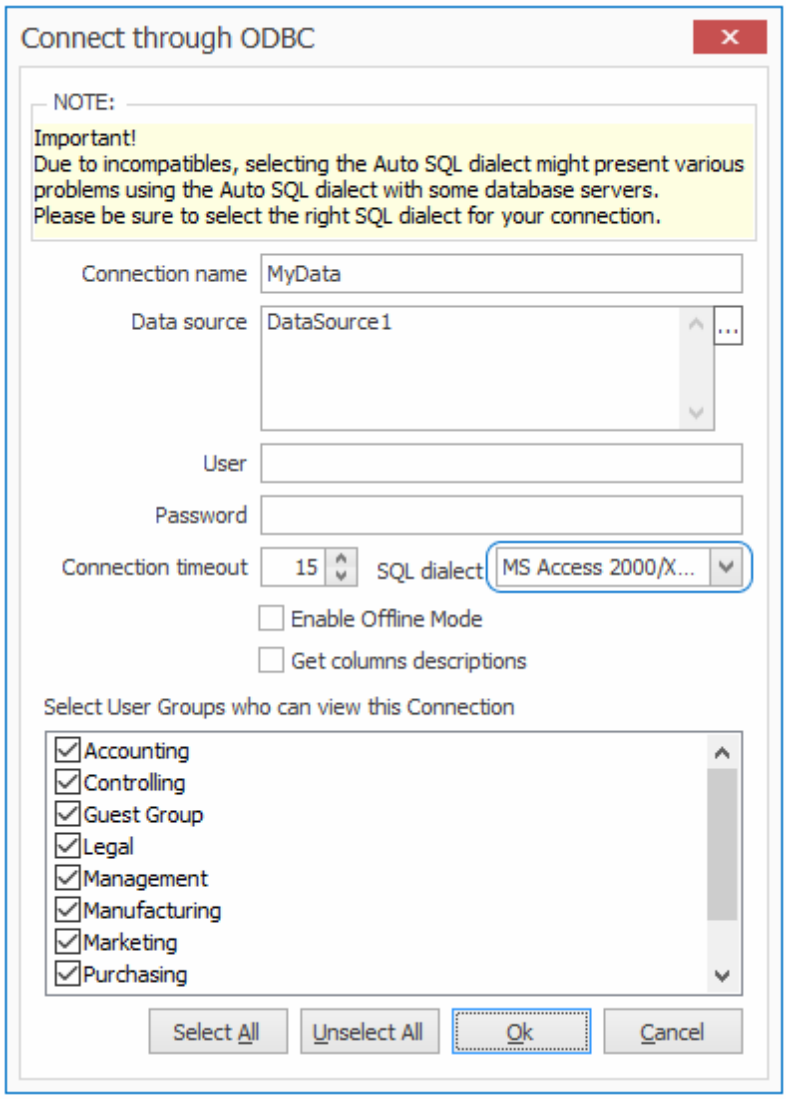


4.2 Using in DBxtra

Troubleshooting PostgreSQL ODBC Connection in DBxtra

This page explains how to troubleshoot your ODBC connection to PostgreSQL in DBxtra.

Due to incompatibilities between DBxtra and PostgreSQL, leaving the `SQL dialect` property to its default might present various issues. To resolve compatibility issues, set the property to `MS Access 2000/XP/2003` or `ANSI SQL/2003` for DBxtra version 11.0.1 or newer, and to `ANSI SQL/2003` for versions prior to 11.0.1.



4.3 Using in Informatica PowerCenter

You can access PostgreSQL data from Informatica PowerCenter on Windows and Linux.

- Connect Informatica PowerCenter to PostgreSQL on Windows
- Connect Informatica PowerCenter to PostgreSQL on Linux

4.3.1 Connect to Informatica PowerCenter on Windows

You can connect Informatica PowerCenter to PostgreSQL through an ODBC driver on Windows to unify and manage data across these systems.

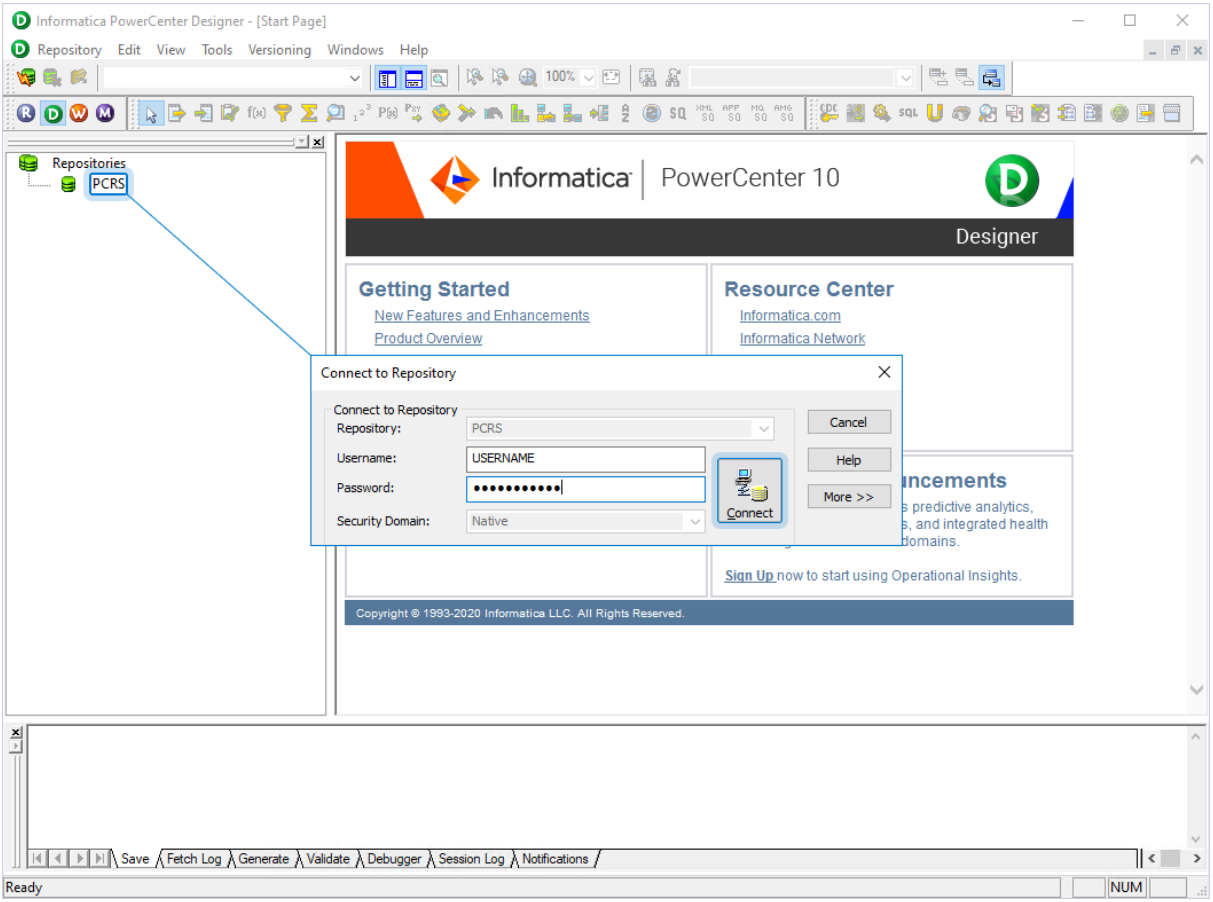
Prerequisites

- Configure the Informatica services.
- Install the PowerCenter Client tools.
- Create a repository folder in PowerCenter Repository Manager.
- Install Devart ODBC Driver for PostgreSQL. For instructions, see Installation.
- Configure a data source name (DSN). For instructions, see Windows DSN Configuration.

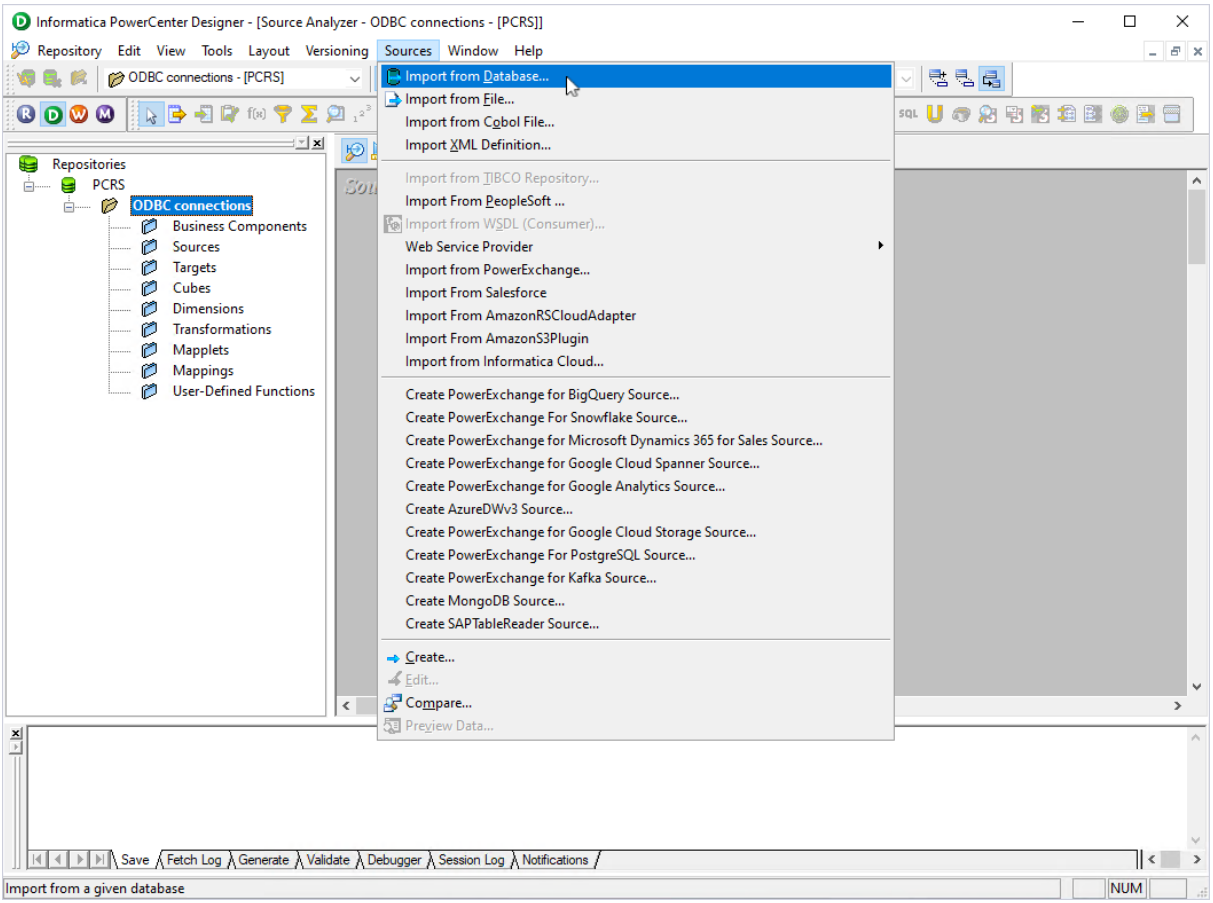
Add a data source in Informatica PowerCenter

Before you proceed, make sure PowerCenter Designer isn't running.

1. Open the `C:\Informatica\10.4.1\clients\PowerCenterClient\client\bin\powrmart.ini` file in a text editor.
2. In the `[ODBCDLL]` section, add `PostgreSQL=PMODBC.DLL`, then save the changes.
3. Open **PowerCenter Designer**.
4. Double-click the repository name (in this example, **PCRS**), enter your Informatica credentials, then click **Connect**.

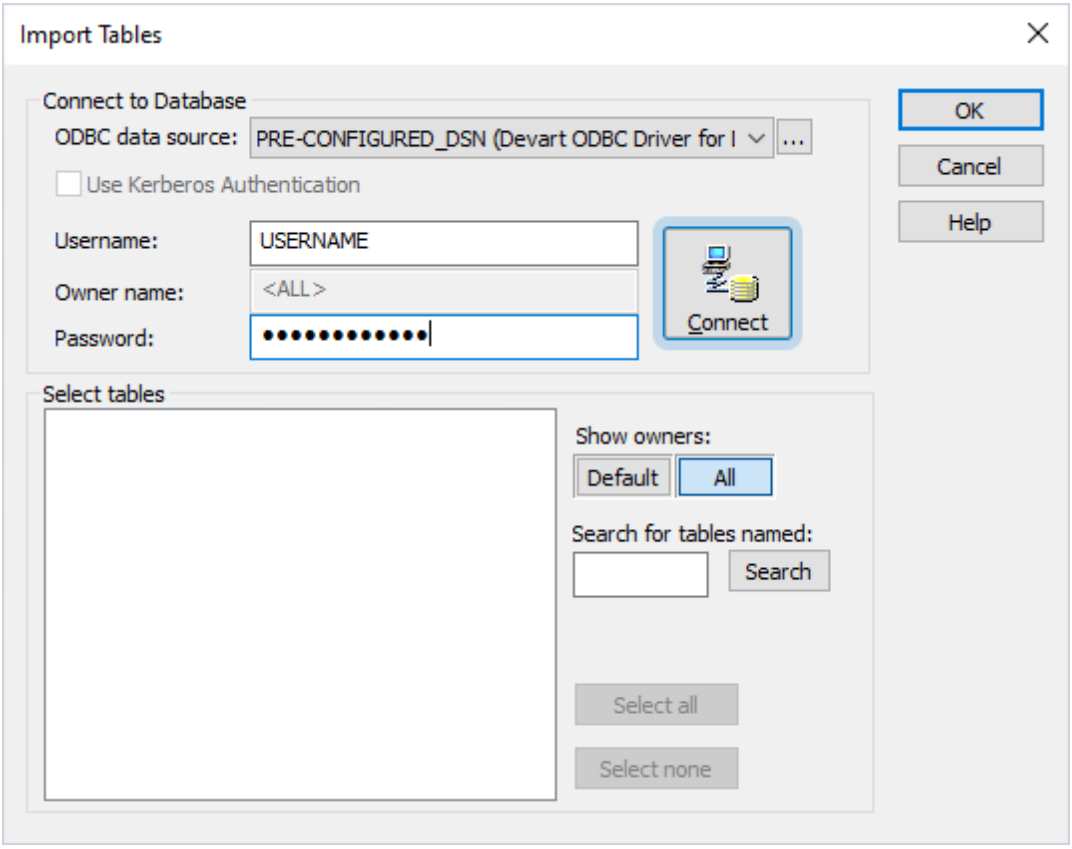


5. Double-click the repository folder (in this example, **ODBC connections**), then select **Sources** > **Import from Database**.

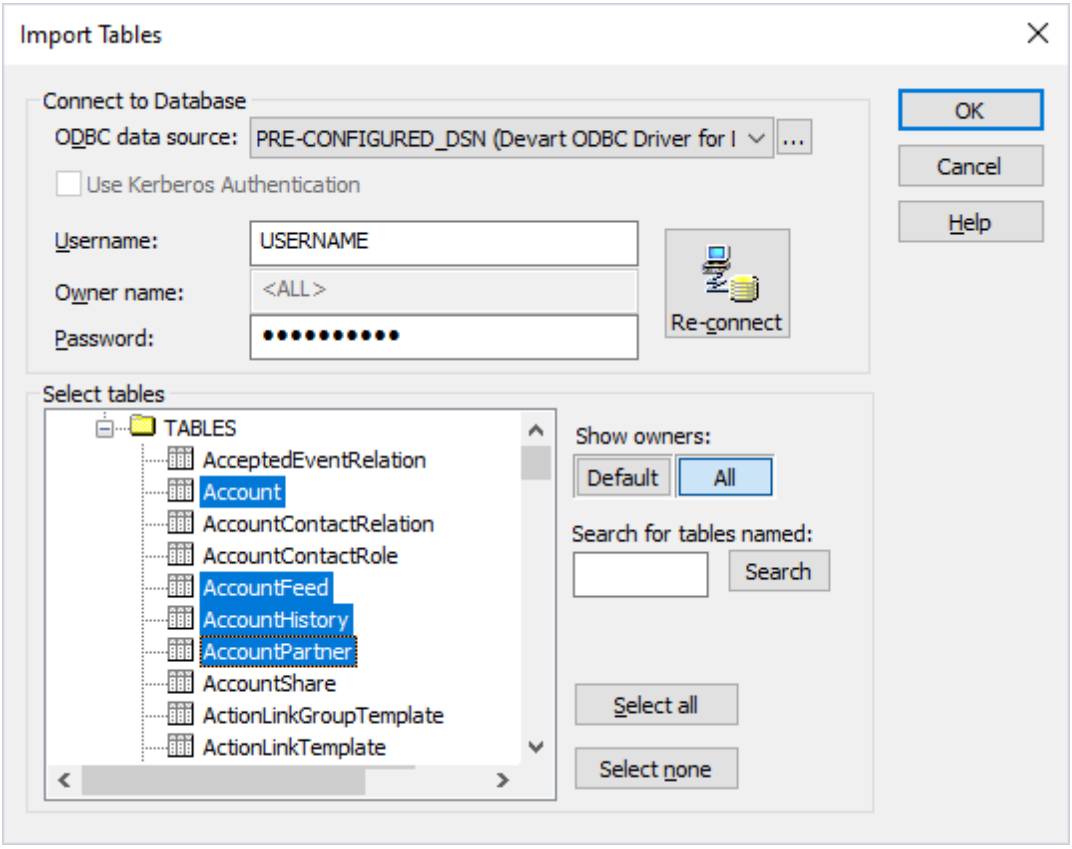


The Import Tables dialog opens.

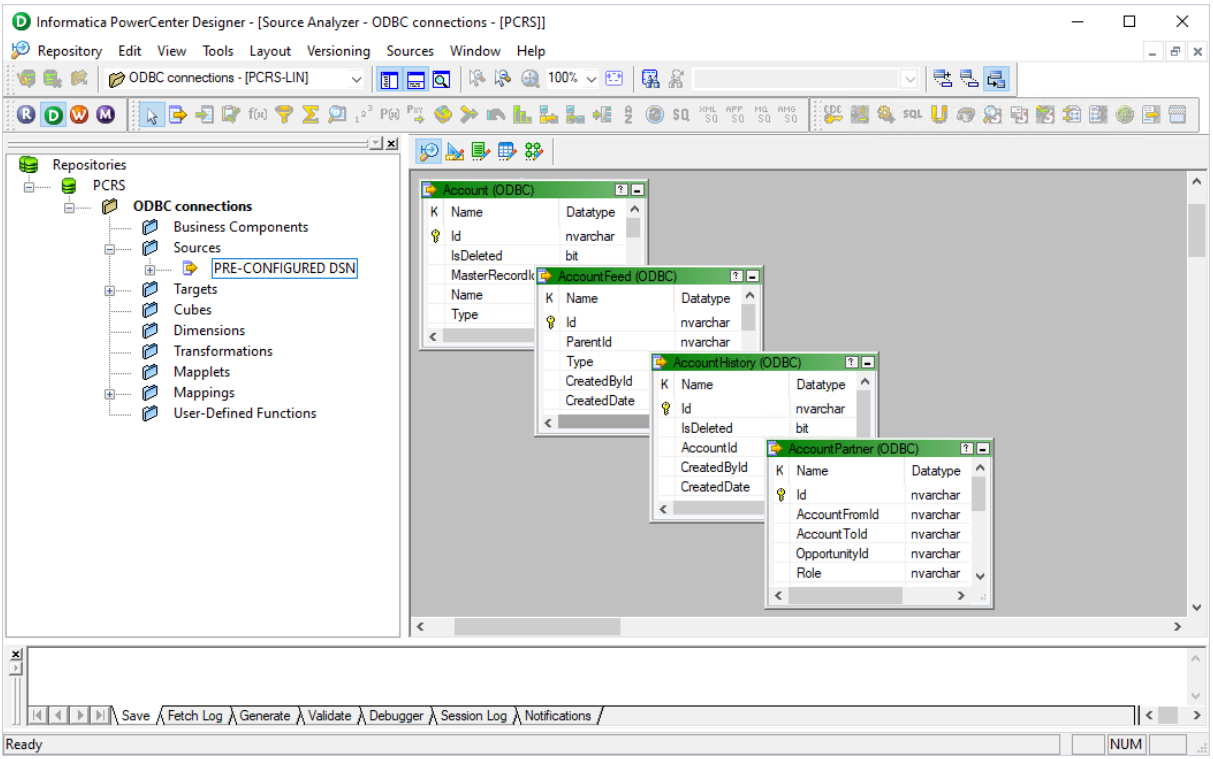
6. From the **ODBC data source** menu, select the needed DSN.
7. In the **Username** and **Password** fields, enter your PostgreSQL credentials.
8. Under **Show owners**, select **All**.
9. Click **Connect**.



10. In the **Select tables** section, expand the tree and select the tables you want to import.
11. Click **OK**.



The table schemas appear in the Source Analyzer, and the data source is added to the **Sources** subfolder of the repository folder. You can now create mappings and work with PostgreSQL data in Informatica PowerCenter.



4.3.2 Connect to Informatica PowerCenter on Linux

You can set up and verify a connection between Informatica PowerCenter and PostgreSQL through an ODBC driver on Linux.

Prerequisites

- Configure the Informatica services.
- Install Devart ODBC Driver for PostgreSQL. For instructions, see Installation.
- Configure a data source name (DSN). For instructions, see Linux DSN Configuration.

Connect to PostgreSQL

1. Navigate to the directory where the `ssgodbc.linux64` utility is located.

```
cd /opt/informatica/tools/debugtools/ssgodbc/linux64
```

2. Run the `ssgodbc.linux64` utility to verify the connection to PostgreSQL.

```
./ssgodbc.linux64 -d <your_dsn> -v
```

3. Run a SQL query to retrieve data.

```
SELECT Id,Name FROM <table>;
```

```
[oracle@itsm-7230-lin ~]$ cd /opt/informatica/tools/debugtools/ssgodbc/linux64
[oracle@itsm-7230-lin linux64]$ ./ssgodbc.linux64 -d DEVART_SALESFORCE -v
./ssgodbc.linux64: /lib64/libodbc.so: no version information available (required by ./ssgodbc.linux64)
Connected
ODBC version        = -03.52-
DBMS name           = -Salesforce-
DBMS version        = -01.00.0000-
Driver name         = --
Driver version      = -03.04.0200-
Driver ODBC version = -03.51-

Enter SQL string: SELECT Id,Name FROM Account;
                Id                                                                                                  Name
001KB000008RHMWYA4                                                                                                  Acme (Sample)
001KB000008RHMXYA4                                                                                                  Global Media (Sample)
```

4.4 Using in Microsoft Access

Connecting Microsoft Access to PostgreSQL Using an ODBC Driver

This article explains how to connect Microsoft Access to PostgreSQL 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 PostgreSQL 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 PostgreSQL. For the purpose of this article, we tested an ODBC connection to PostgreSQL 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 PostgreSQL.

Importing PostgreSQL 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 PostgreSQL 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 PostgreSQL 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 PostgreSQL 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 PostgreSQL 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.

4.5 Using in Microsoft Excel

Connecting to PostgreSQL from Microsoft Excel using ODBC Driver for PostgreSQL

You can use Microsoft Excel to access data from a PostgreSQL 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 Driver, you will need to use the 64-bit version of Excel.

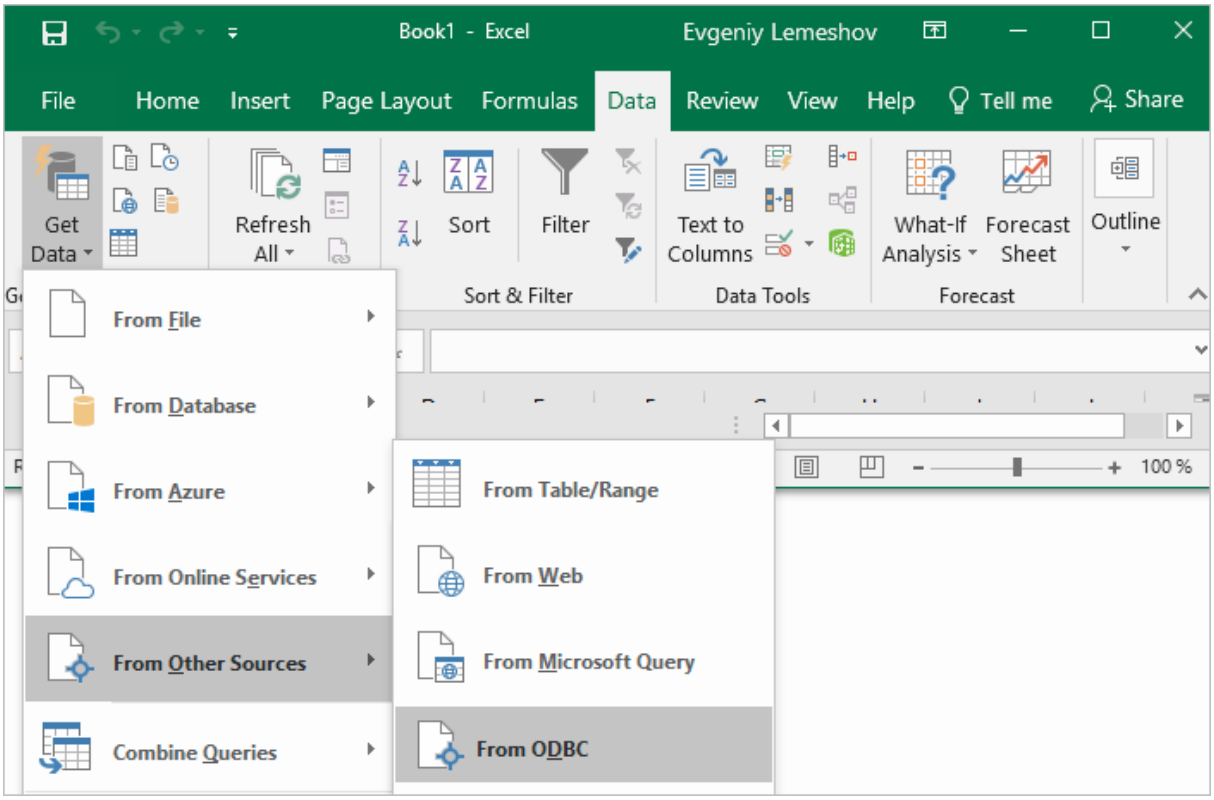
When working with Microsoft Excel, there are different ways of retrieving data from various data sources using our ODBC drivers. Please see the list of sections that will help you to connect Excel to PostgreSQL database:

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

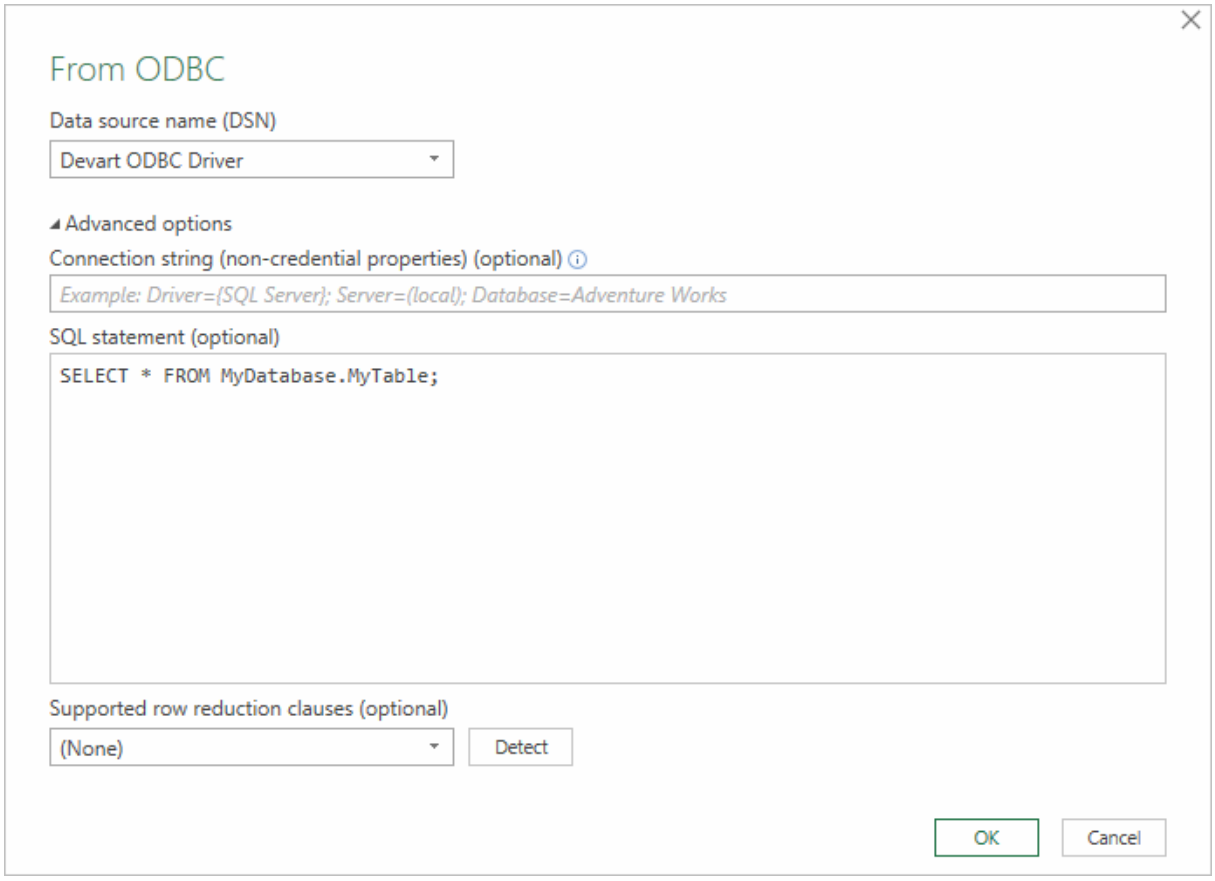
Connecting Excel to PostgreSQL with Get & Transform (Power Query)

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

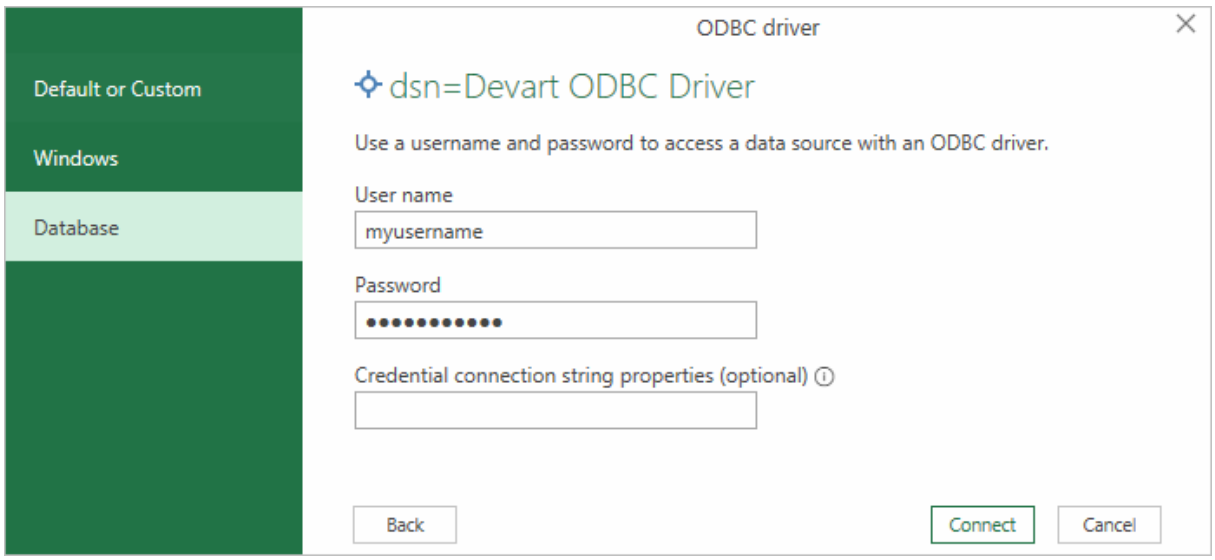
1. Click the **Data** in Excel, then expand the **Get Data** drop-down list. Click **From Other Sources** > **From ODBC**.



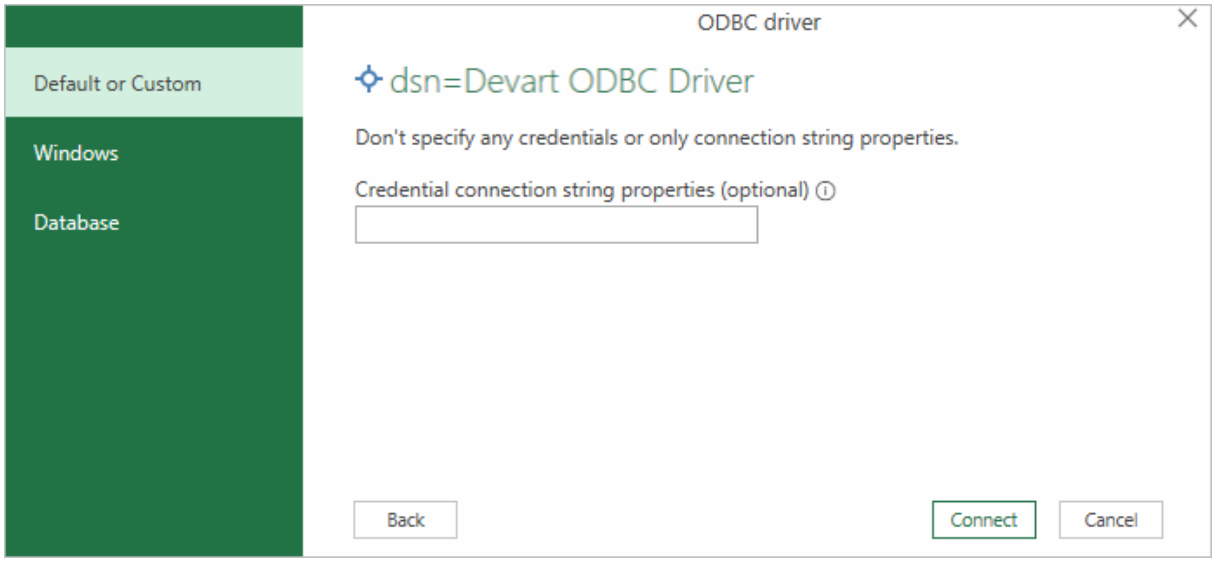
2. In the **From ODBC** dialog, choose your data source name (DSN). If you haven't configured your ODBC driver yet, you can expand the **Advanced Options** dialog box and enter the connection string for your data source (without credentials, which are defined in the credentials dialog box in the next step). Additionally, you can enter an SQL statement that will be executed right after establishing a connection to the data source. Click **OK**.



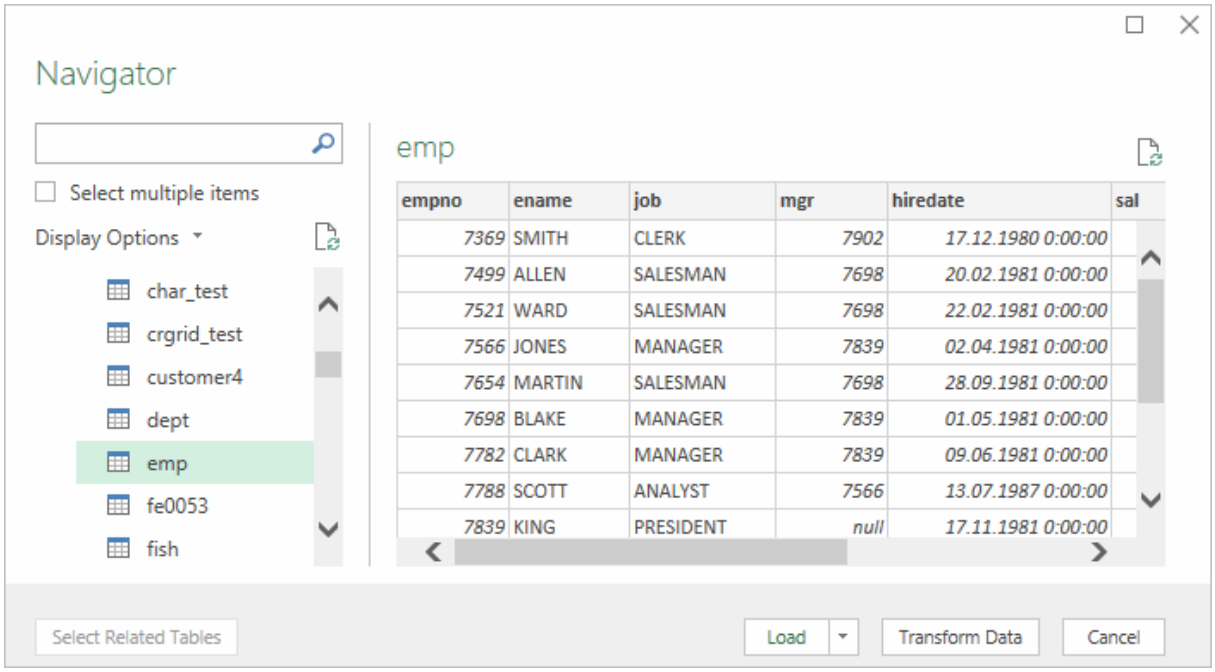
3. If you're using a database username or password, select **Database** and enter your credentials in the dialox bog, then click **Connect**.



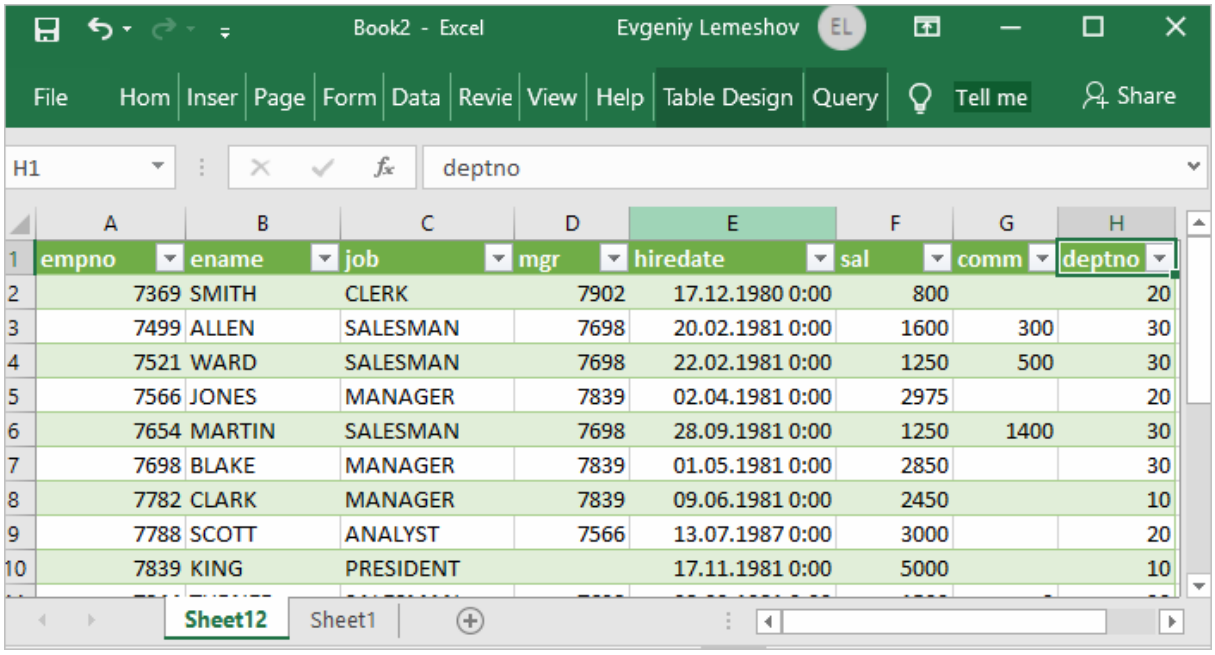
If your database is not password-protected or you've already specified your credentials in the ODBC data source settings, select **Default or Custom** and press **Connect**



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.



	A	B	C	D	E	F	G	H
1	empno	ename	job	mgr	hiredate	sal	comm	deptno
2	7369	SMITH	CLERK	7902	17.12.1980 0:00	800		20
3	7499	ALLEN	SALESMAN	7698	20.02.1981 0:00	1600	300	30
4	7521	WARD	SALESMAN	7698	22.02.1981 0:00	1250	500	30
5	7566	JONES	MANAGER	7839	02.04.1981 0:00	2975		20
6	7654	MARTIN	SALESMAN	7698	28.09.1981 0:00	1250	1400	30
7	7698	BLAKE	MANAGER	7839	01.05.1981 0:00	2850		30
8	7782	CLARK	MANAGER	7839	09.06.1981 0:00	2450		10
9	7788	SCOTT	ANALYST	7566	13.07.1987 0:00	3000		20
10	7839	KING	PRESIDENT		17.11.1981 0:00	5000		10

Connecting Excel to PostgreSQL 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 PostgreSQL with the Query Wizard

You can use this option to create a simple query for retrieving data from PostgreSQL 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 PostgreSQL with Microsoft Query

You can use this option to create a more complex query for retrieving PostgreSQL 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 PostgreSQL). 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 PostgreSQL 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 PostgreSQL), 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.

4.6 Using in Microsoft Visual Studio

Importing PostgreSQL Data into Visual Studio Through an ODBC Connection

A Visual Studio is a powerful tool containing features that allow editing, debugging, and compilating the code and creating applications that can be connected to any databases product and services on a local machine and network, and any type of cloud (private, public, or hybrid). To connect Visual Studio to a data source such as PostgreSQL, you can use an appropriate ODBC driver.

This guide describes how to connect to PostgreSQL and retrieve data importing them to Visual Studio with an ODBC driver. It is assumed that you have already installed and configured a DSN for ODBC driver for PostgreSQL.

1. Run Visual Studio Desktop and click **Tool** and select **Connect to Database**.
2. In the **Add connection** dialog box, select the **Microsoft ODBC Data Source** as a data source.
3. In the **Data source specification** point expand the **Data Source Name (DSN)** drop-down list and select the previously configured DSN for PostgreSQL. Alternatively, you can connect to the database by entering the DSN in a **Use connection string** field. To check whether your connection is successful, click **Test connection**. Click **OK**.
4. If your data source is password-protected, Visual Studio will prompt you for user credentials. Type your **Username** and **Password** in the respective fields and click **OK**.

5. In the Server Explorer you can see the database structure. Choose **Tables**, right-click the table you want to view the data of and select **Retrieve Data**. You can also preview the contents of the database objects by clicking on them.

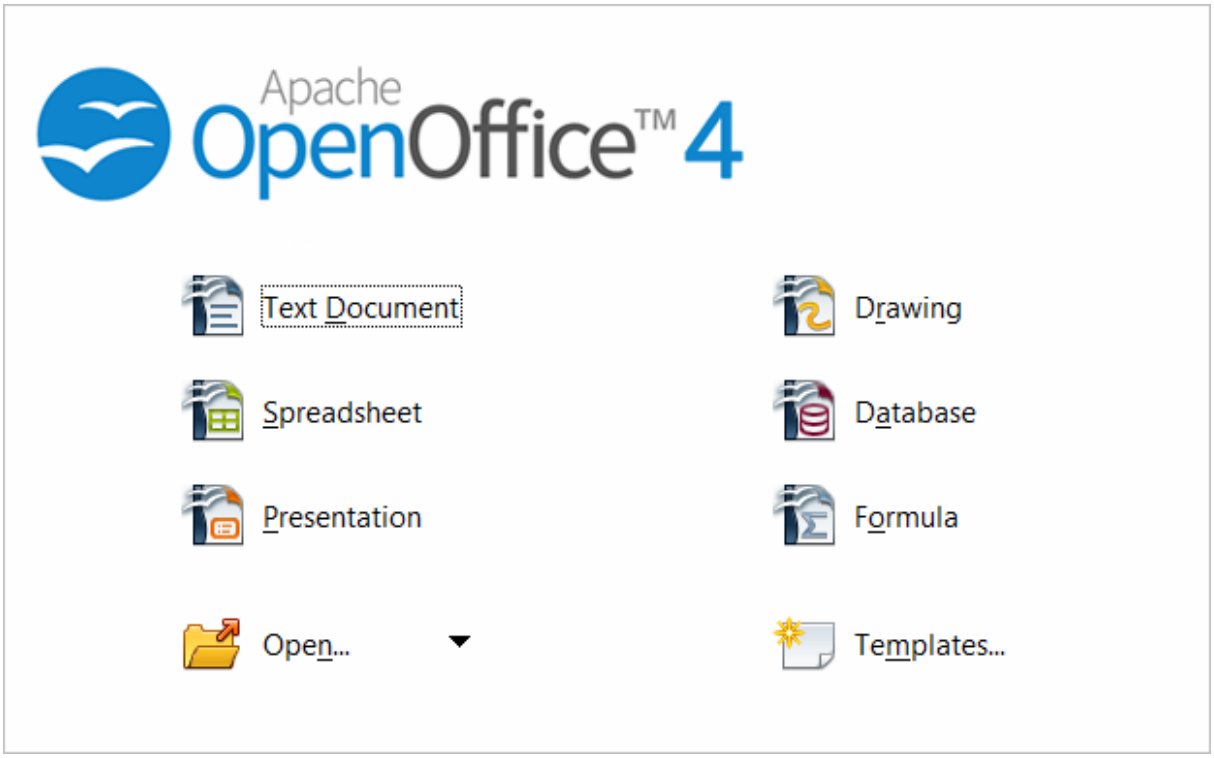
4.7 Using in OpenOffice and LibreOffice

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

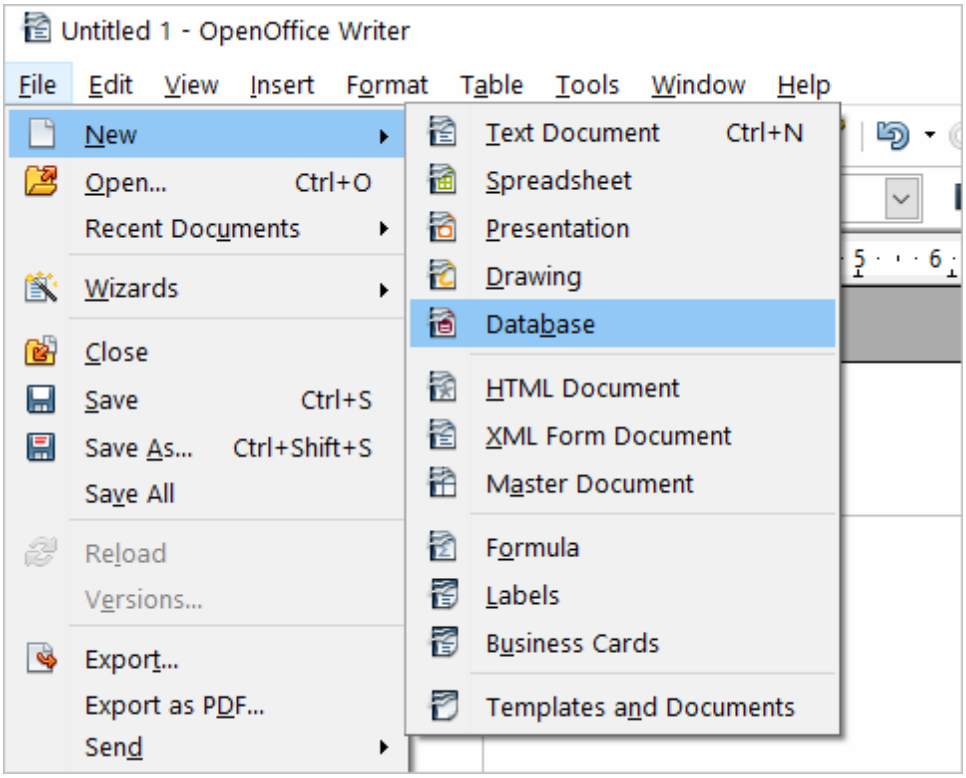
The article describes how to use Apache OpenOffice and LibreOffice to access ODBC data sources using the respective driver. You can access PostgreSQL data from Open Office Base or LibreOffice Base — desktop database management systems. Note that the Windows version of OpenOffice is 32-bit, and you may 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 PostgreSQL, perform the steps below:

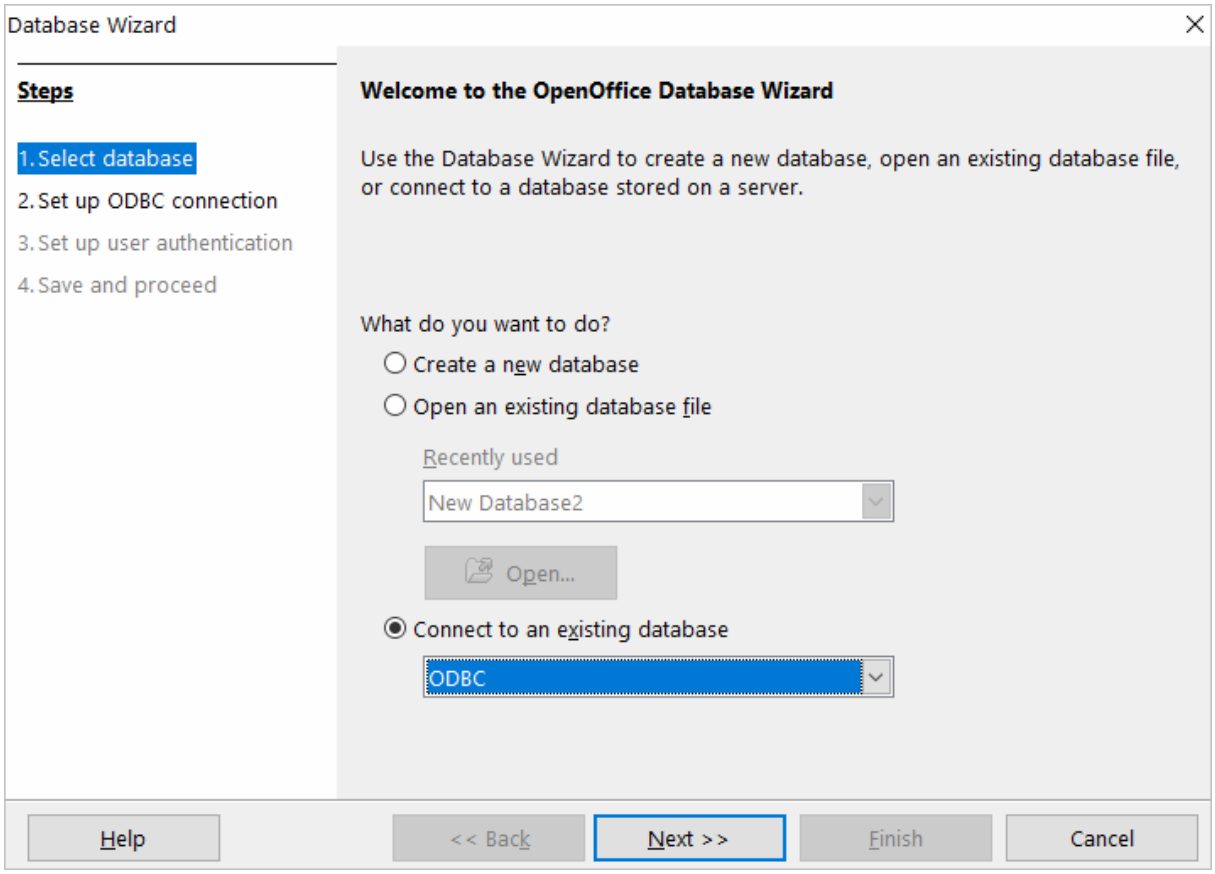
1. Start OpenOffice or LibreOffice, click **Database** to open the **Database Wizard**.



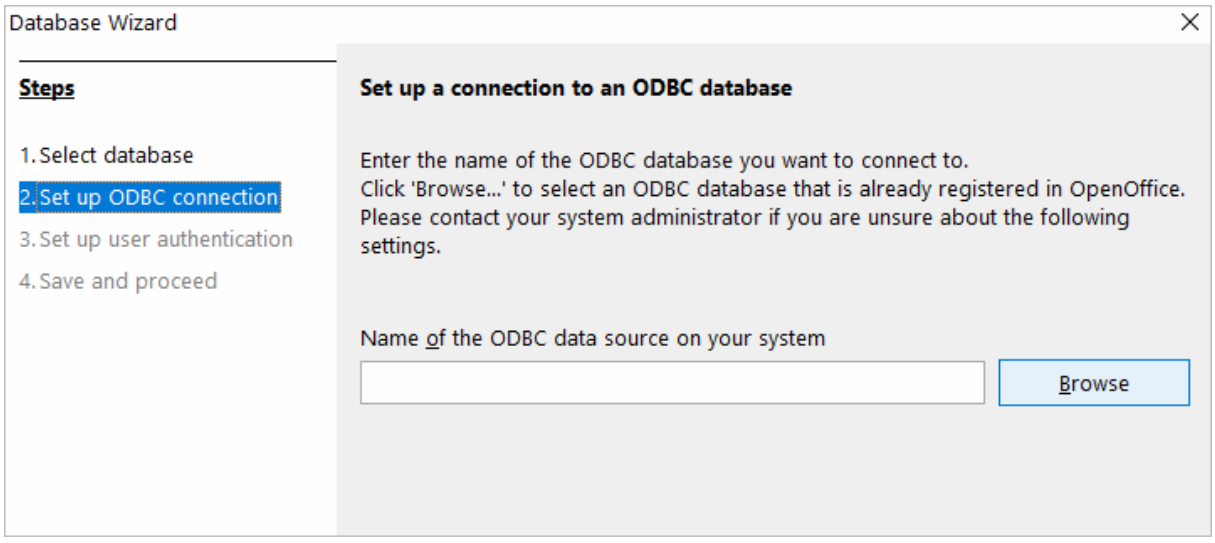
Alternatively, you can launch the **Database Wizard** from OpenOffice or LibreOffice Calc, Writer or any other tool by choosing **File > New > Database**.

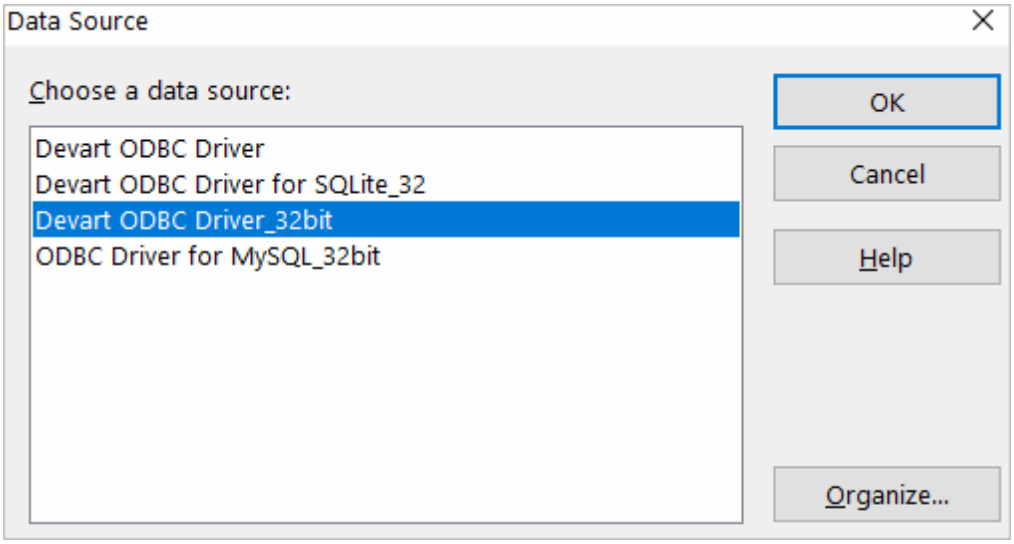


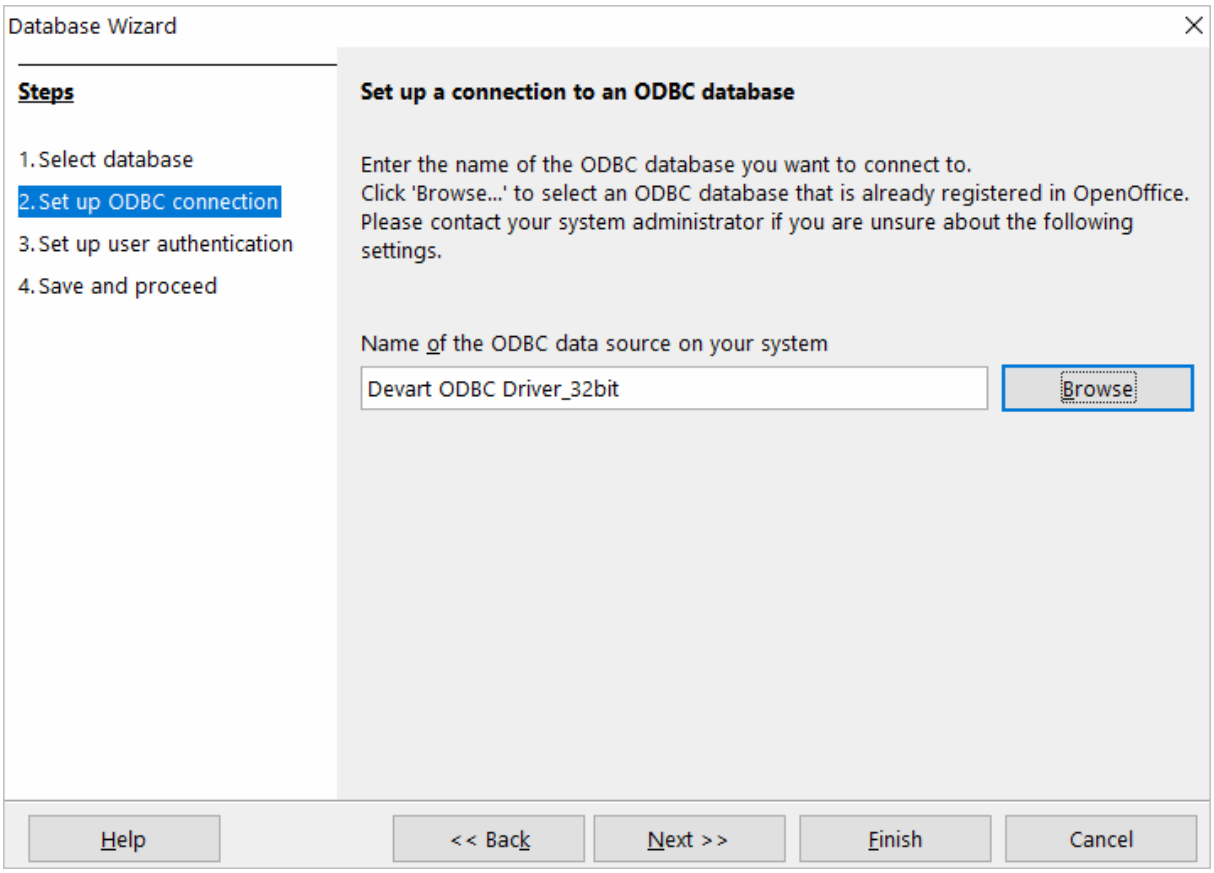
2. In the **Database Wizard dialog box**, click **Connect to an existing database**, select **ODBC** from the drop-down list, and click **Next**.



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 PostgreSQL**, or you can click **Browse**, double-click the data source you need, and then click **Next**.

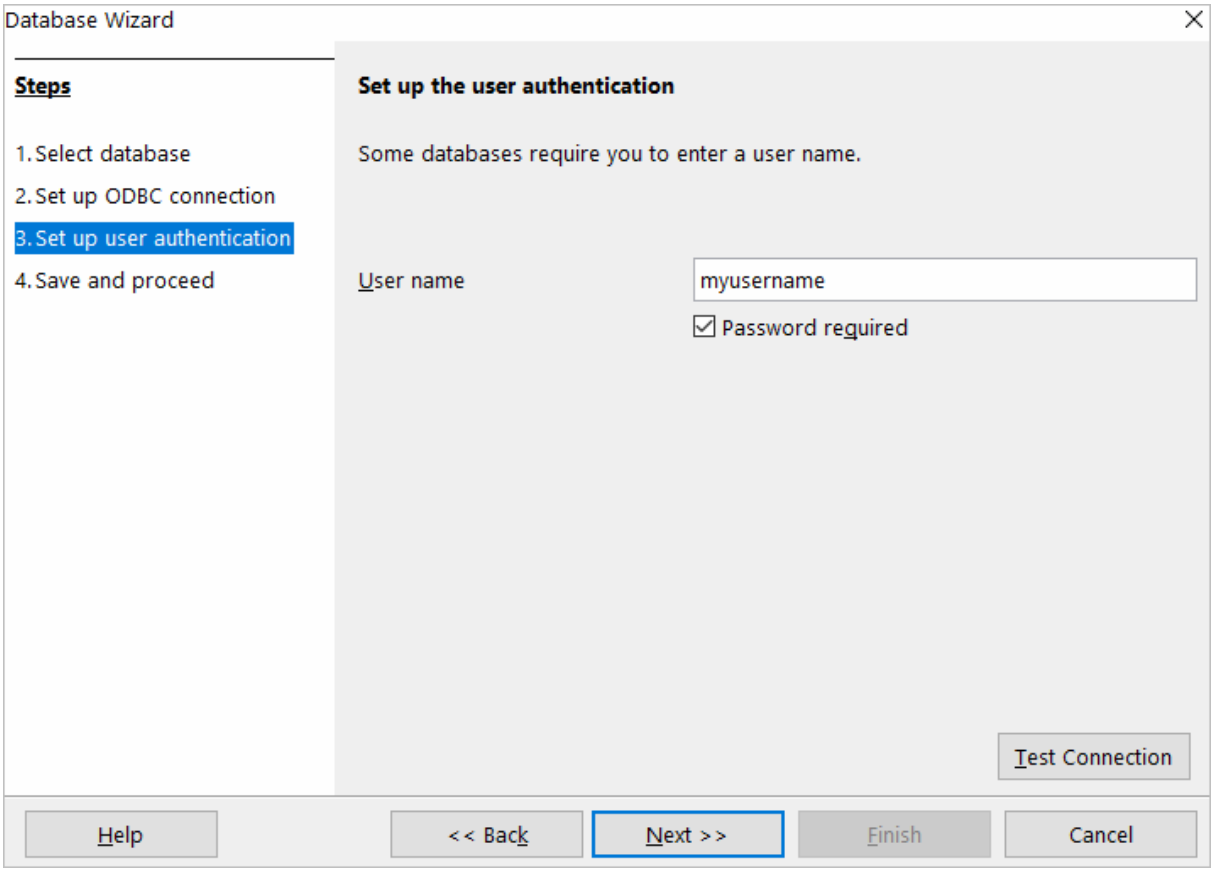




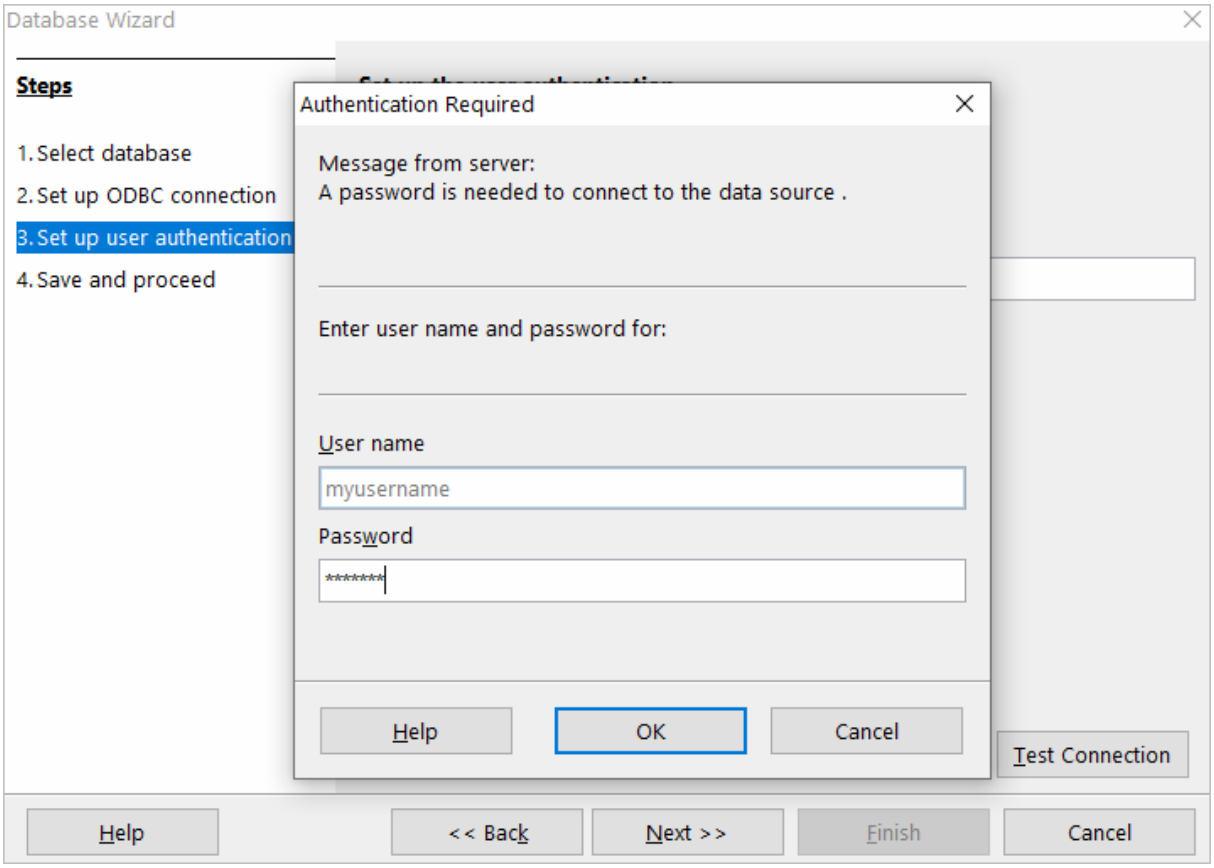


4. If your database requires a user name, type it into the **User name** field. If you are connecting to a password protected database, check the **Password required** field. Alternatively, you can specify these parameters in the data source settings of your ODBC

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

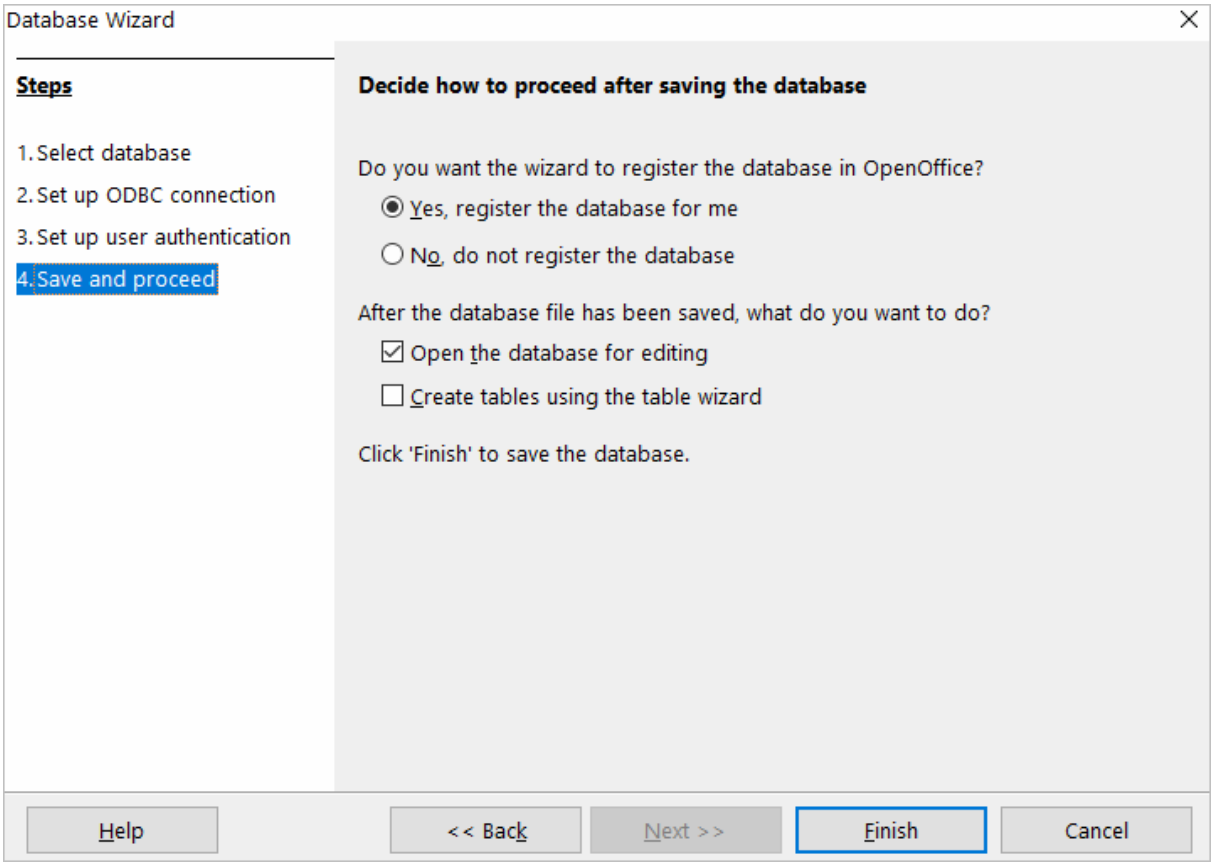


To test the connection to your data source, click **Test Connection**, input your credentials and click **OK**.

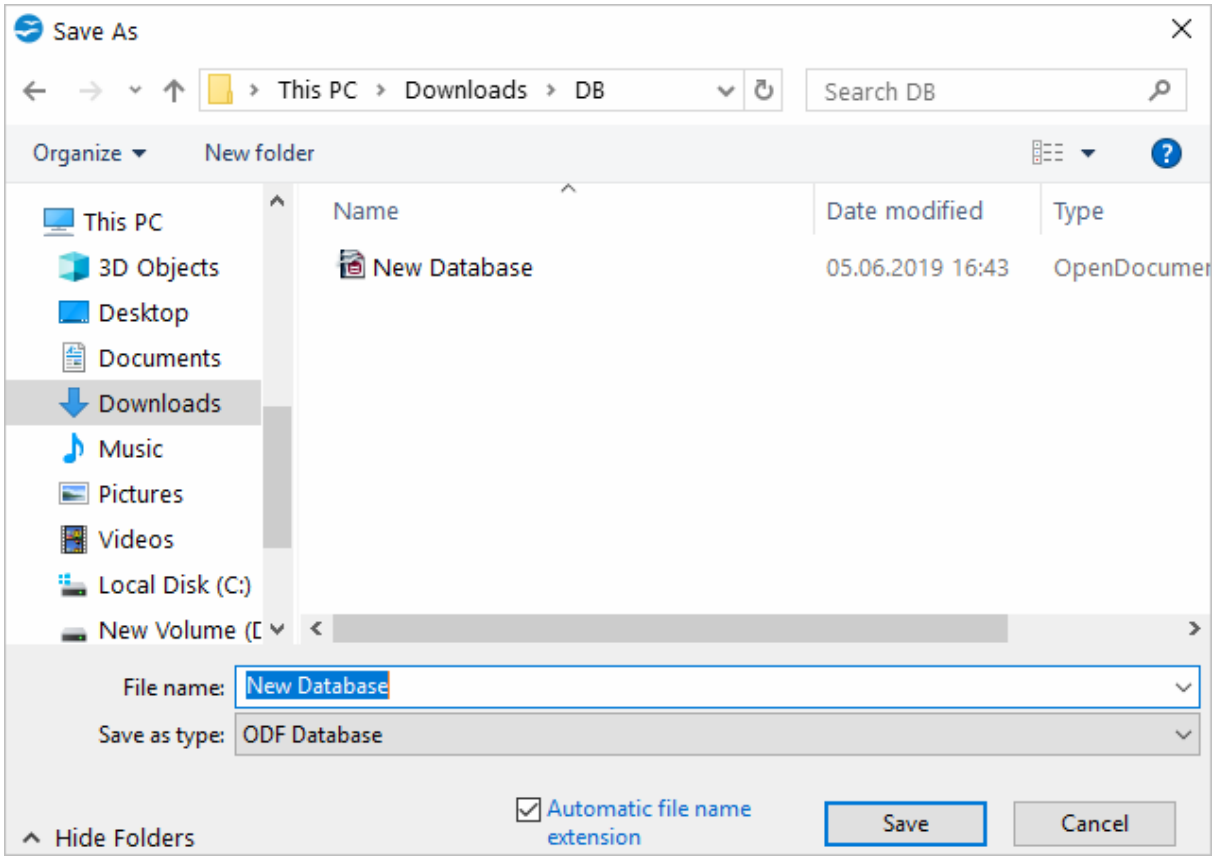


If you have entered valid credentials, you will see a success message. Click **Next** to proceed to the final step.

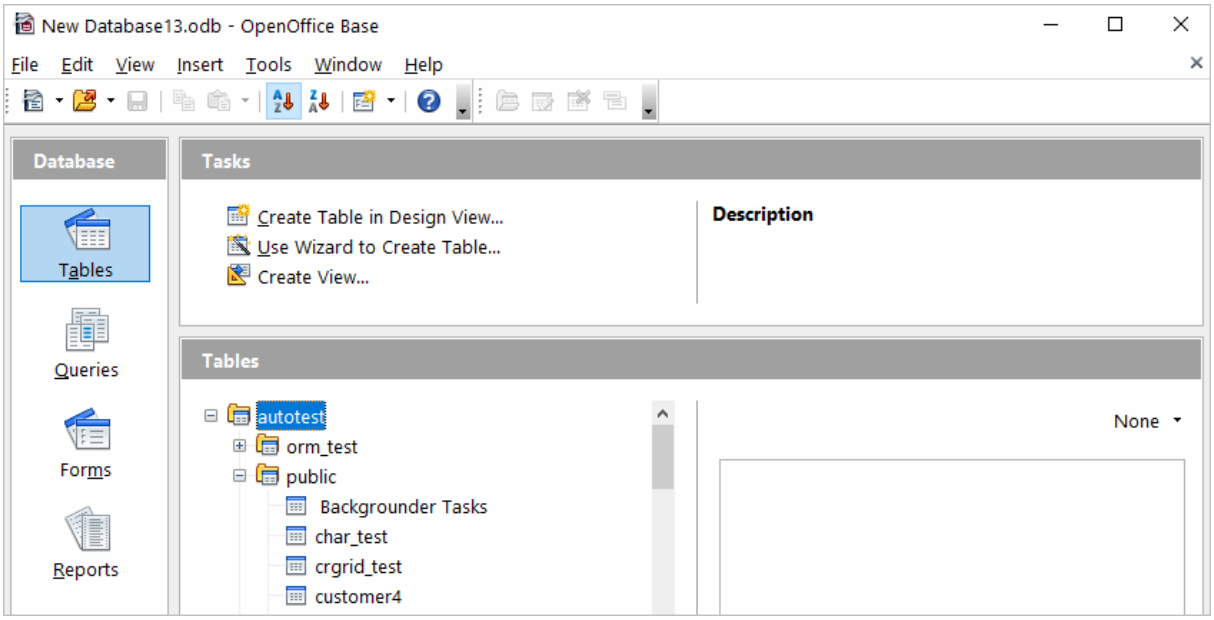
5. You can keep the default selection in this dialog box and click **Finish**.

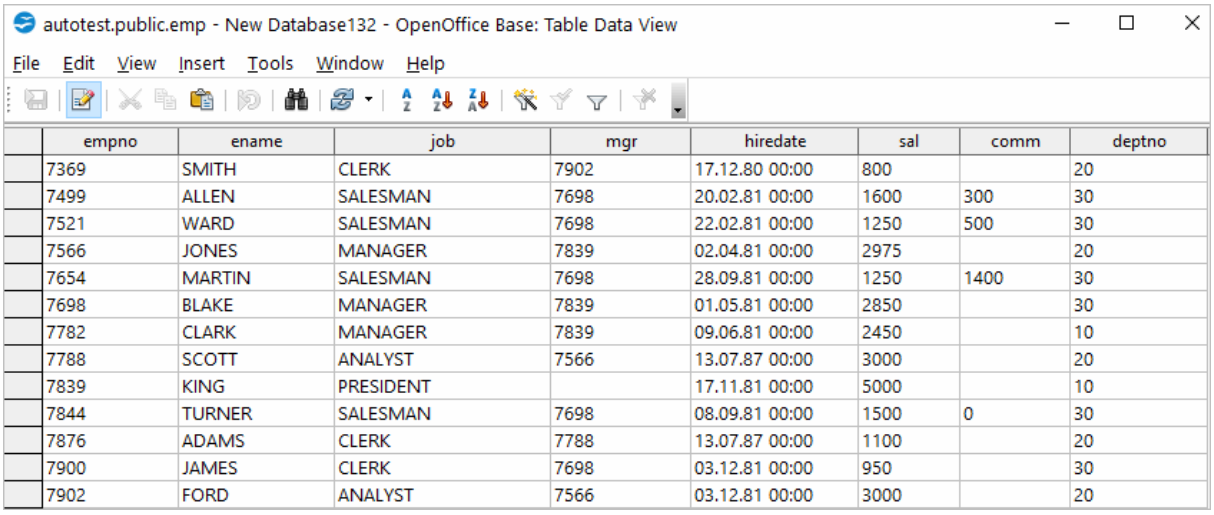


You will be prompted to give a name to your new database and select the directory where you want to store it.



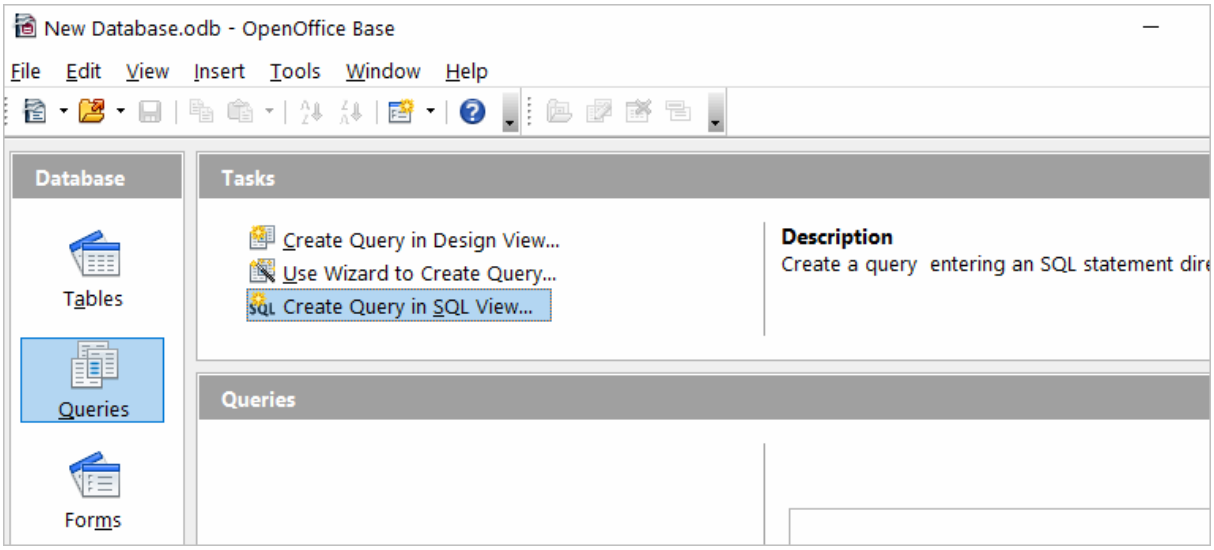
6. When the database opens, you will see the list of tables from your data source diplayed in OpenOffice or LibreOffice Base workspace. To view the data from a specific table, double-click the table name.



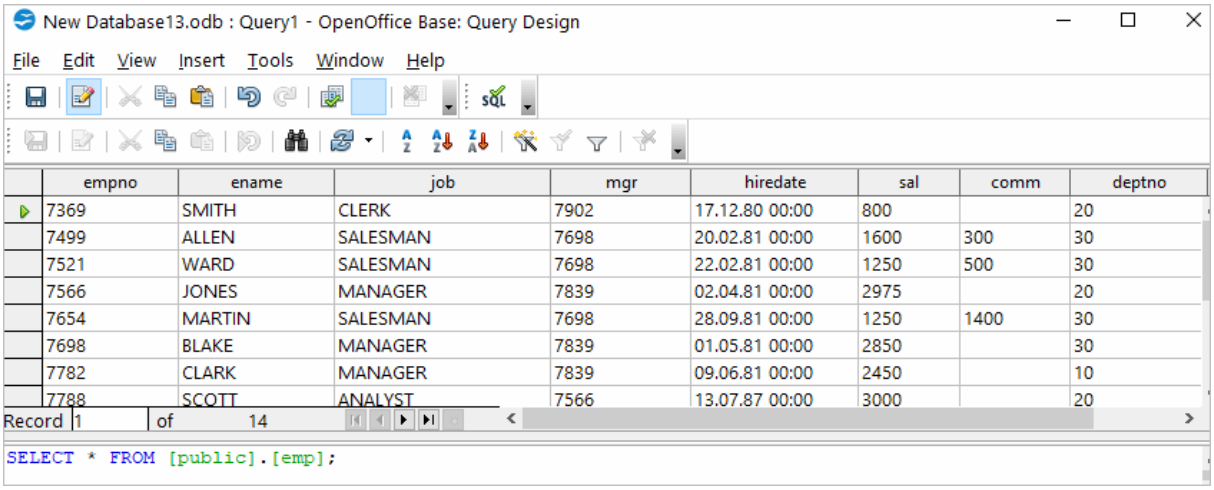


empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	17.12.80 00:00	800		20
7499	ALLEN	SALESMAN	7698	20.02.81 00:00	1600	300	30
7521	WARD	SALESMAN	7698	22.02.81 00:00	1250	500	30
7566	JONES	MANAGER	7839	02.04.81 00:00	2975		20
7654	MARTIN	SALESMAN	7698	28.09.81 00:00	1250	1400	30
7698	BLAKE	MANAGER	7839	01.05.81 00:00	2850		30
7782	CLARK	MANAGER	7839	09.06.81 00:00	2450		10
7788	SCOTT	ANALYST	7566	13.07.87 00:00	3000		20
7839	KING	PRESIDENT		17.11.81 00:00	5000		10
7844	TURNER	SALESMAN	7698	08.09.81 00:00	1500	0	30
7876	ADAMS	CLERK	7788	13.07.87 00:00	1100		20
7900	JAMES	CLERK	7698	03.12.81 00:00	950		30
7902	FORD	ANALYST	7566	03.12.81 00:00	3000		20

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 Oracle DBLink

4.8.1 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 PostgreSQL, 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, `initPostgreSQL.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=PostgreSQL
```

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=PostgreSQL)
        (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.

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

Create Database Links

To access an ODBC data source, you must create a database link using a database tool like SQL Plus or dbForge Studio for Oracle: 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.

When you create the database link in dbForge Studio for Oracle, you can see your newly created link in Database Links on the left panel. After creating the database link, you can run a query against the ODBC data source using the following syntax:

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

See also

Configuring Oracle Database Gateway for ODBC

4.8.2 Troubleshooting in Oracle DBLink

Troubleshooting SQL Queries in Oracle DBLink

When you create a SQL statement to access data in PostgreSQL using a database link, make sure to encapsulate the table names in double quotes, for example:

```
SELECT * FROM "employee"@"dblink_name"
```

Note that table names are case-sensitive.

4.9 Using in PHP

Connecting to PostgreSQL from PHP using ODBC Driver for PostgreSQL

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 PostgreSQL via ODBC. The script connects to PostgreSQL database and fetches 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
   $user = "myusername";
   $password = "mypassword";
   $ODBCConnection = odbc_connect("DRIVER={Devart ODBC Driver for PostgreSQ
```

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.

```
$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.

```
   $result = odbc_result_all($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.

4.10 Using in Power BI

Importing PostgreSQL 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 PostgreSQL, you can use a corresponding ODBC driver.

This tutorial explores how to connect to PostgreSQL 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 PostgreSQL.

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 PostgreSQL
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 PostgreSQL data into Power BI for analysis, select the needed table and click **Load**.

4.11 Using in Python

Installing the ODBC Driver for PostgreSQL

One of the most convenient methods to connect to an external database or access cloud data from Python is via ODBC. Devart has developed a range of ODBC Drivers for Python to work with databases and cloud services.

If you don't have Python installed on your machine, go to the Python official website, download the appropriate installer 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. Next, you need to download the ODBC Driver for PostgreSQL. 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 to PostgreSQL from Python using ODBC Driver for PostgreSQL

Here's an example to show you how to connect to PostgreSQL via Devart ODBC Driver in Python. 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. To execute the script, you can type the 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

```
import pyodbc
cnxn = pyodbc.connect('DRIVER={Devart ODBC Driver for PostgreSQL};Server=mys
```

Step 2: Insert a row

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: Execute query

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()
cursor.close()
cnxn.close()
```

4.12 Using in QlikView

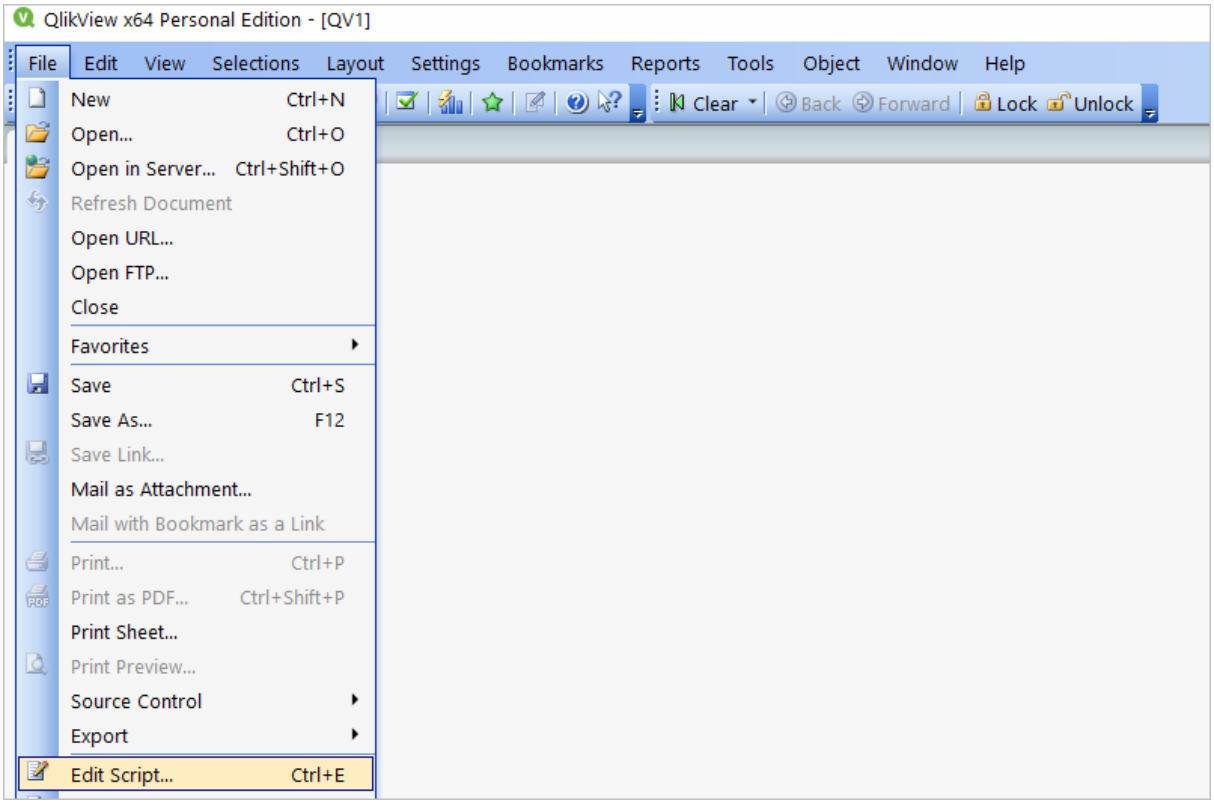
Connecting to PostgreSQL from QlikView using ODBC Driver for PostgreSQL

This tutorial describes how to connect and configure QlikView to retrieve data from PostgreSQL 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.

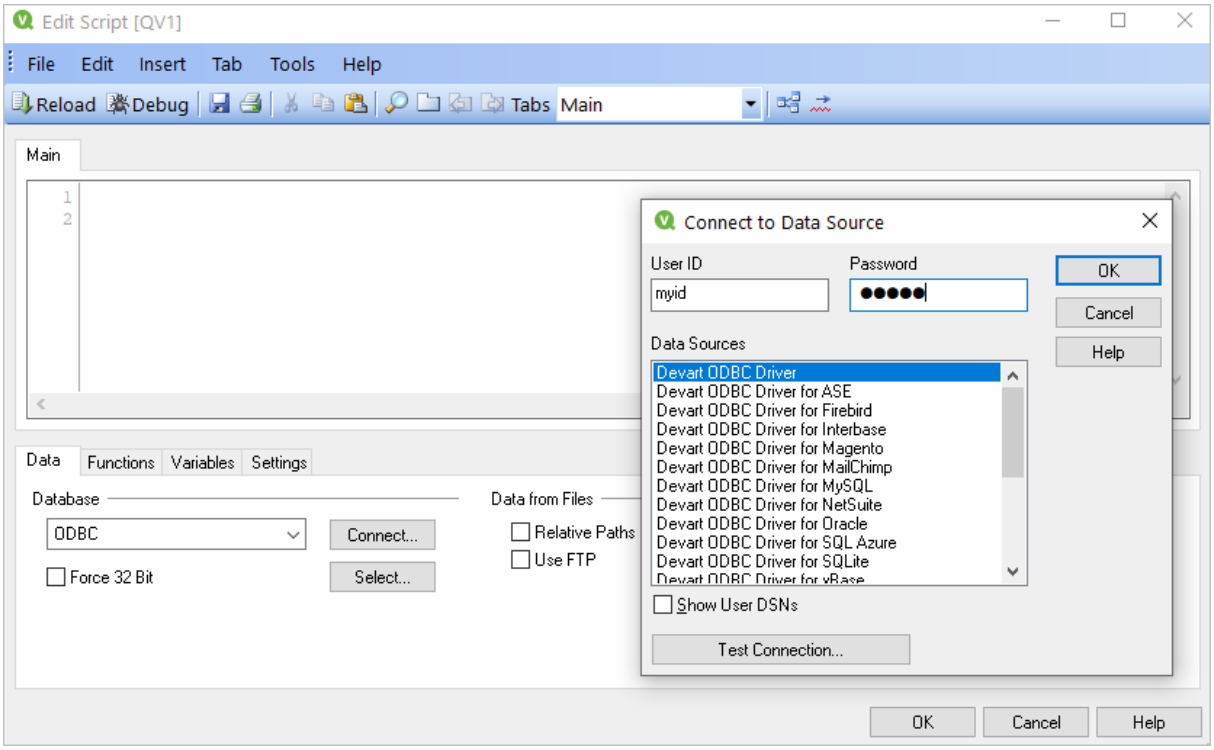
QlikView supports the ODBC connectivity interface for communication with external data sources. An ODBC data source must be configured for the database you want to access. You can create an ODBC connection using a DSN during the ODBC driver installation or later.

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

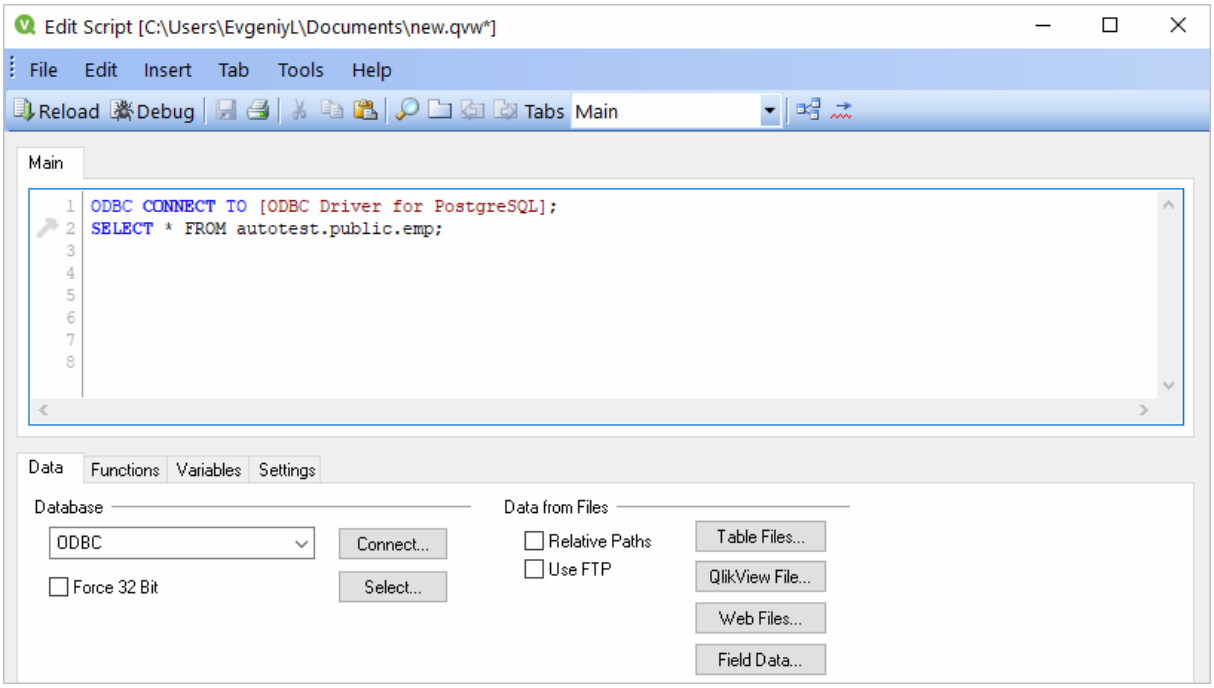
1. Open the QlikView client application and click **File > New**. Close the **Getting Started** wizard and open **File > Edit Script (CTRL+E)**.

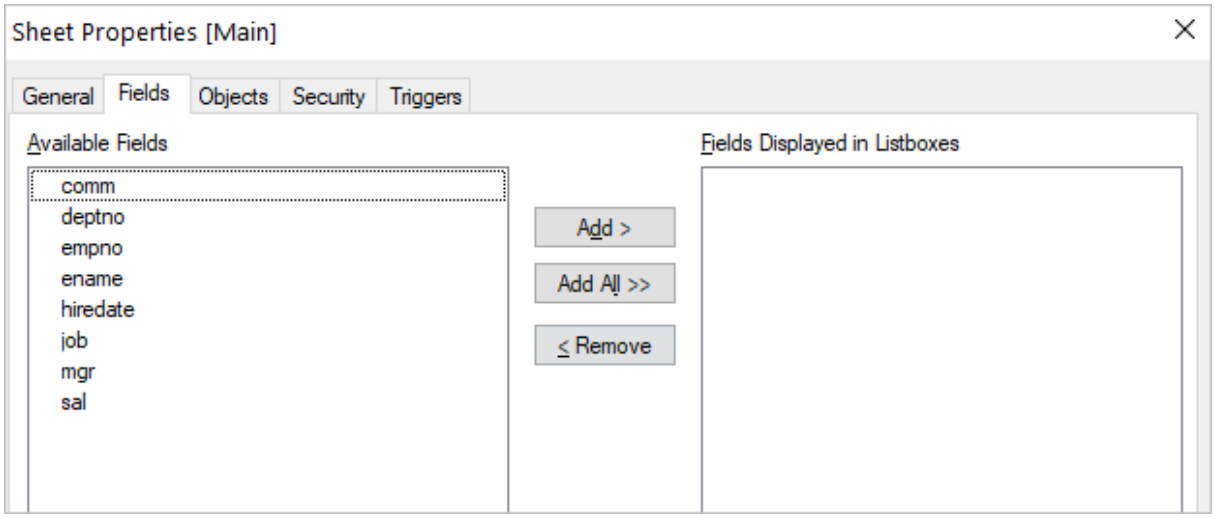


2. In the **Data** tab, choose **ODBC** from the **Database** drop-down and click **Connect**. Select the **Data Source** you created earlier, type in the **User ID** and **Password** if your database is password-protected. You can test the connection by choosing **Test Connection**. The **Connection Test succeeded** message should appear. Click **OK** to connect to your data source.

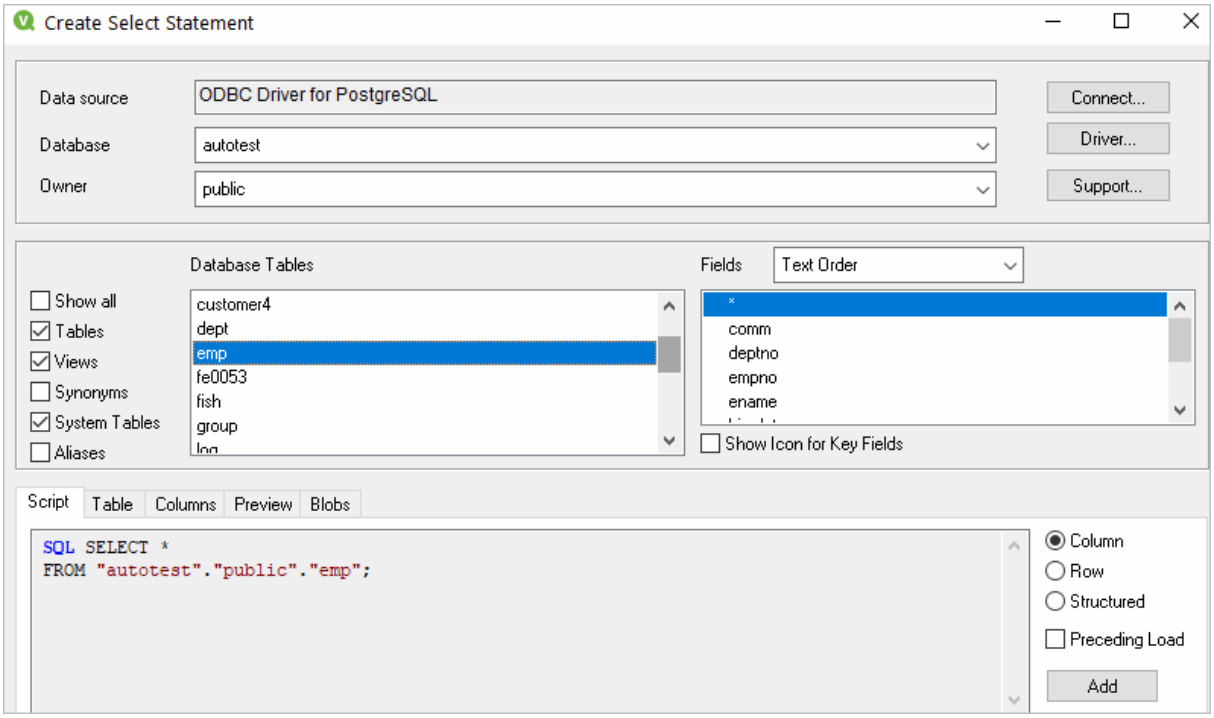


3. To retrieve the data from your data source, you can enter an SQL query and press **F5**. You will be suggested to choose fields to be displayed.

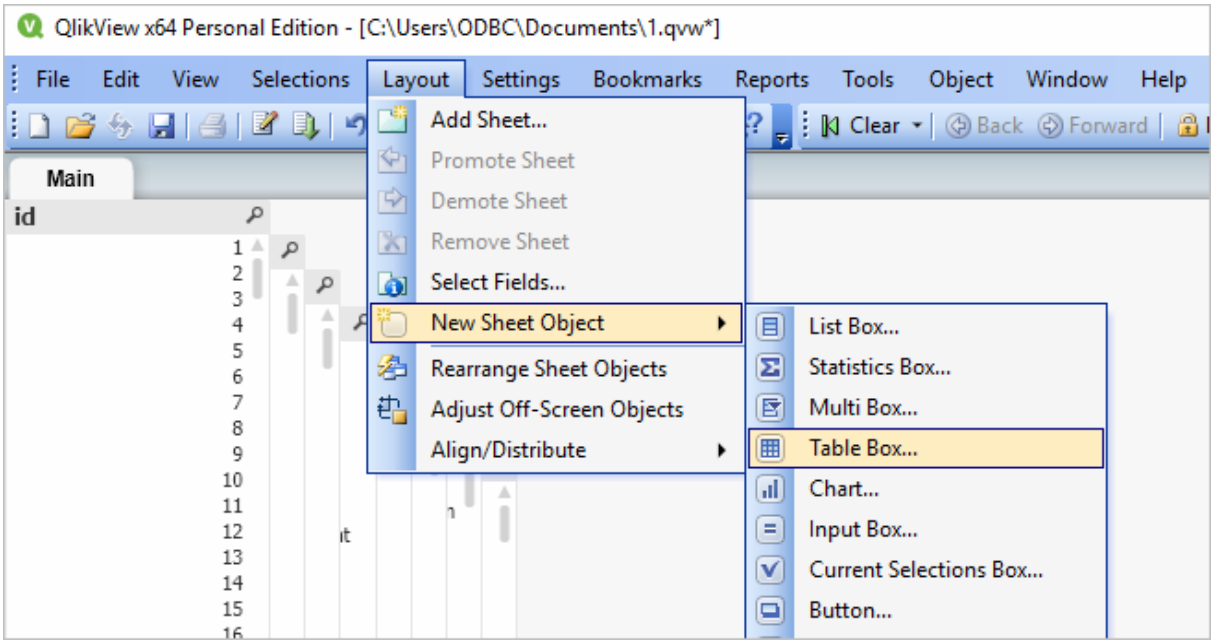


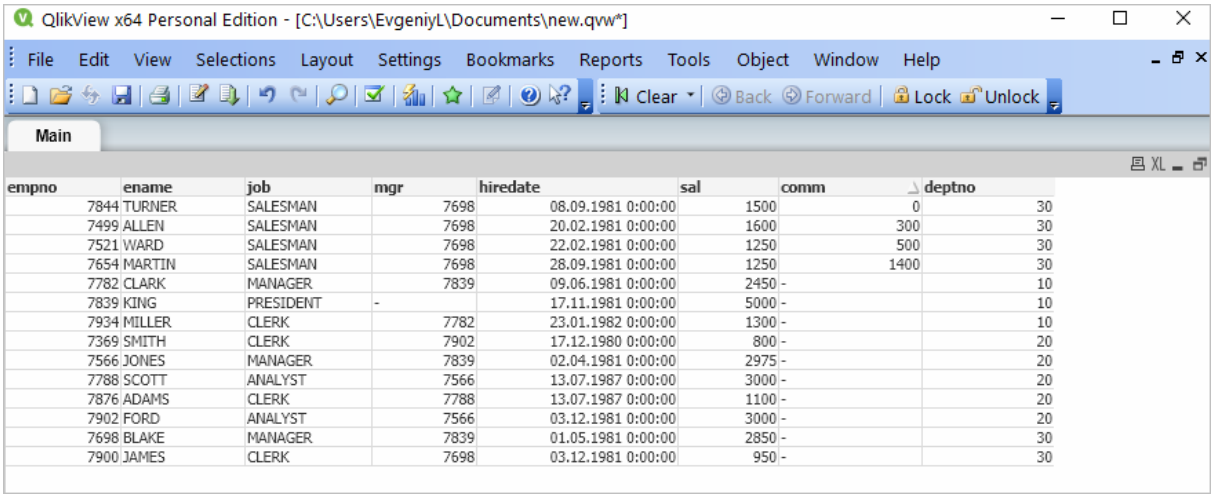


4. Alternatively, you can click **Select**, and QlikView will show you the database structure window where you can compose a SELECT statement for the data to be fetched. You can choose a different database from the database drop-down list. Select the necessary tables and fields. You can retrieve date 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**.





empno	ename	job	mgr	hiredate	sal	comm	deptno
7844	TURNER	SALESMAN	7698	08.09.1981 0:00:00	1500	0	30
7499	ALLEN	SALESMAN	7698	20.02.1981 0:00:00	1600	300	30
7521	WARD	SALESMAN	7698	22.02.1981 0:00:00	1250	500	30
7654	MARTIN	SALESMAN	7698	28.09.1981 0:00:00	1250	1400	30
7782	CLARK	MANAGER	7839	09.06.1981 0:00:00	2450	-	10
7839	KING	PRESIDENT	-	17.11.1981 0:00:00	5000	-	10
7934	MILLER	CLERK	7782	23.01.1982 0:00:00	1300	-	10
7369	SMITH	CLERK	7902	17.12.1980 0:00:00	800	-	20
7566	JONES	MANAGER	7839	02.04.1981 0:00:00	2975	-	20
7788	SCOTT	ANALYST	7566	13.07.1987 0:00:00	3000	-	20
7876	ADAMS	CLERK	7788	13.07.1987 0:00:00	1100	-	20
7902	FORD	ANALYST	7566	03.12.1981 0:00:00	3000	-	20
7698	BLAKE	MANAGER	7839	01.05.1981 0:00:00	2850	-	30
7900	JAMES	CLERK	7698	03.12.1981 0:00:00	950	-	30

4.13 Using in SQL Server Management Studio

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

- Creating a Linked Server
- Troubleshooting in SSMS

4.13.1 Creating a Linked Server

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 PostgreSQL and SQL Server must be installed on the same computer.

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

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

You can use the Microsoft SQL Server Management Studio to connect your PostgreSQL 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 PostgreSQL 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 PostgreSQL:

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 PostgreSQL

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

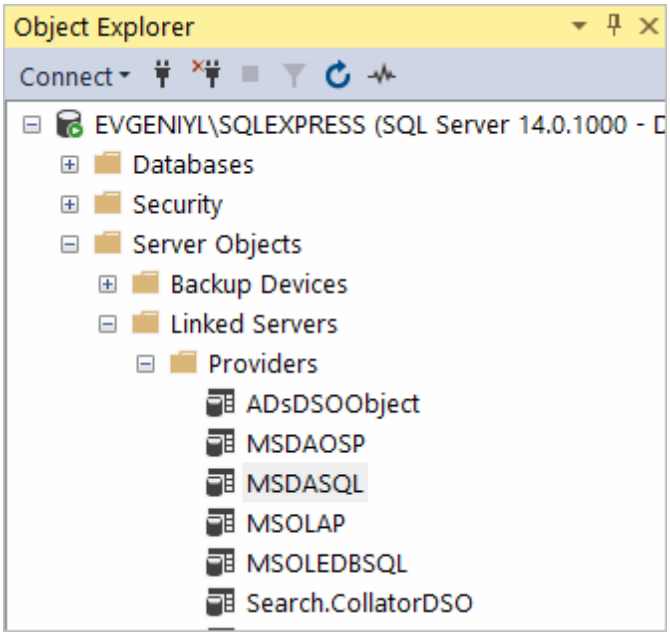
1. Start your Management Studio and choose your SQL Server instance.
2. In the **Object Explorer pane**, expand the **Server Objects**, right-click on **Linked Servers** and then click on **New Linked Server**.
3. Configure your linked server in the dialog box:
   - Give a name for your server in the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

PostgreSQL. Alternatively, you can input the ODBC Driver connection string in the **Provider** field.

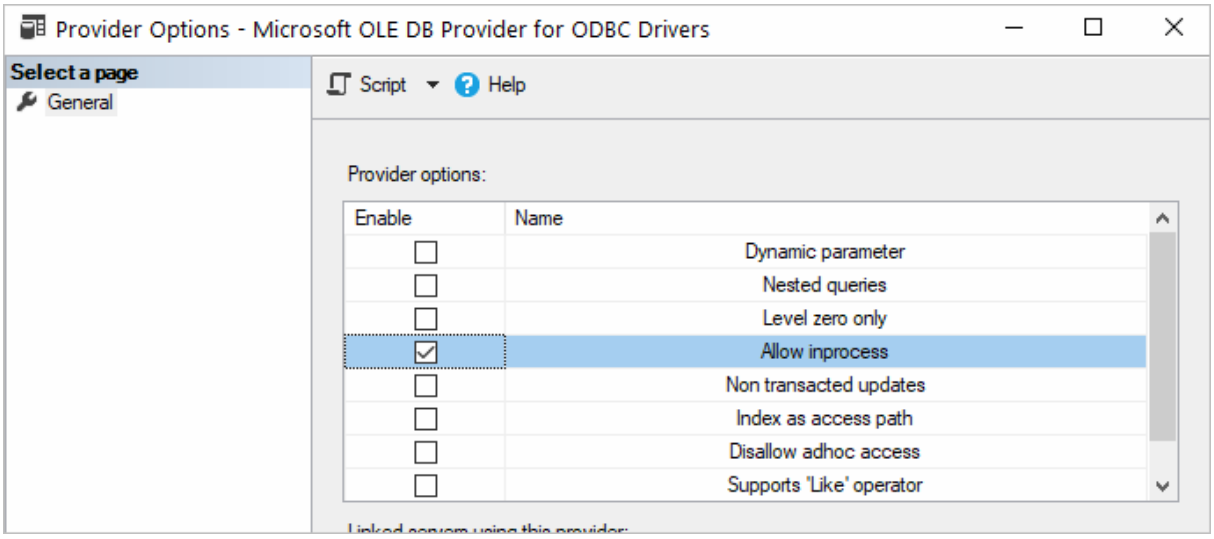
The linked server will appear under the Linked Servers in the Object Explorer Pane. You can now issue distributed queries and access PostgreSQL databases through SQL Server.

Retrieving Data From PostgreSQL

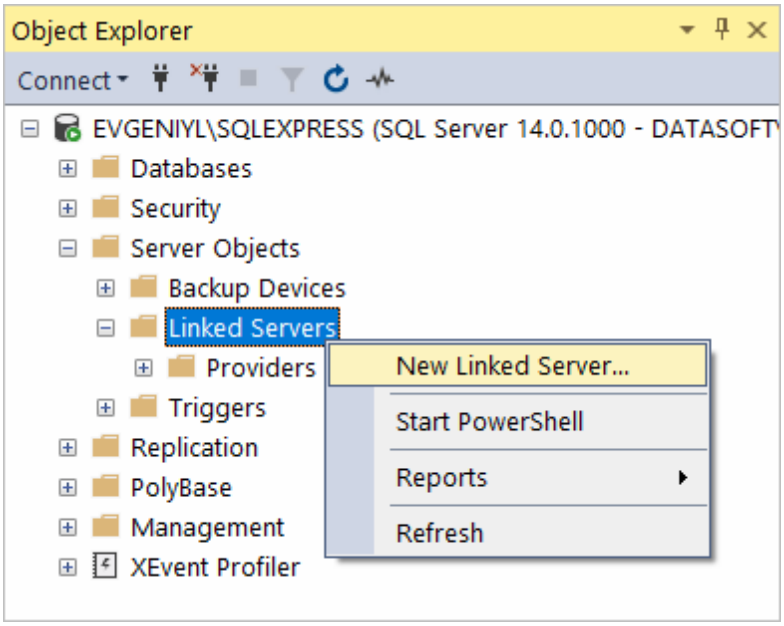
Ensure the **Allow inprocess option** of MSDASQL OLE DB Provider for ODBC Drivers is enabled. For this, find the **MSDASQL** provider in the list of Linked Servers and double-click on it



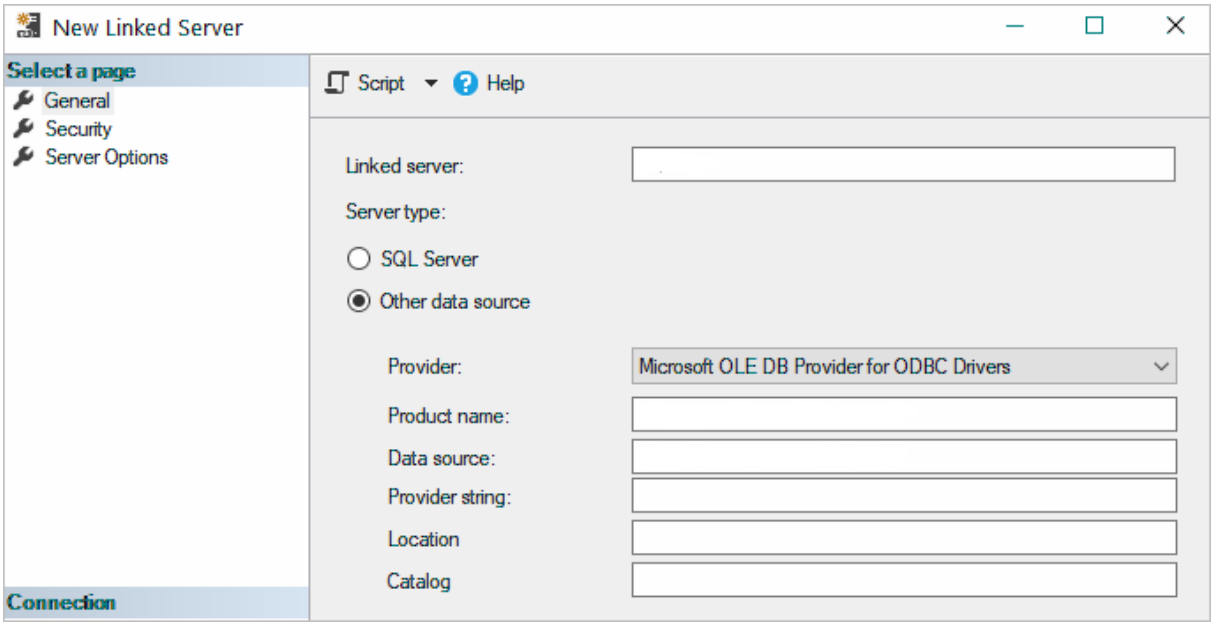
In the appeared **Provider Options** window, enable the **Allow inprocess** checkbox:



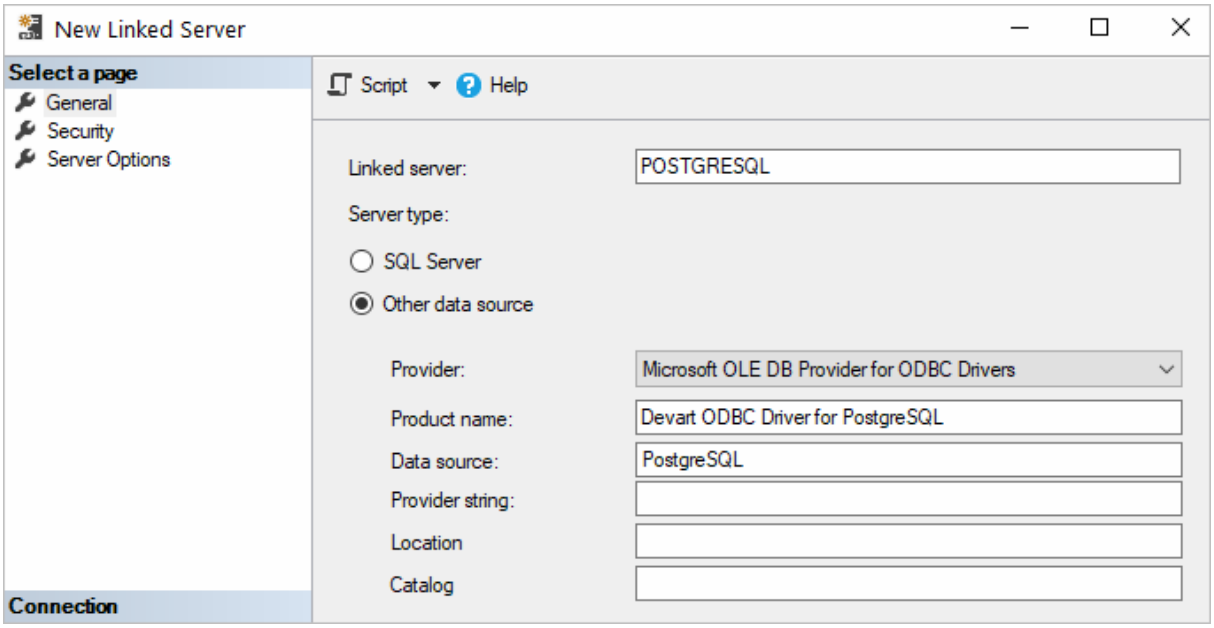
Create a new **Linked Server**



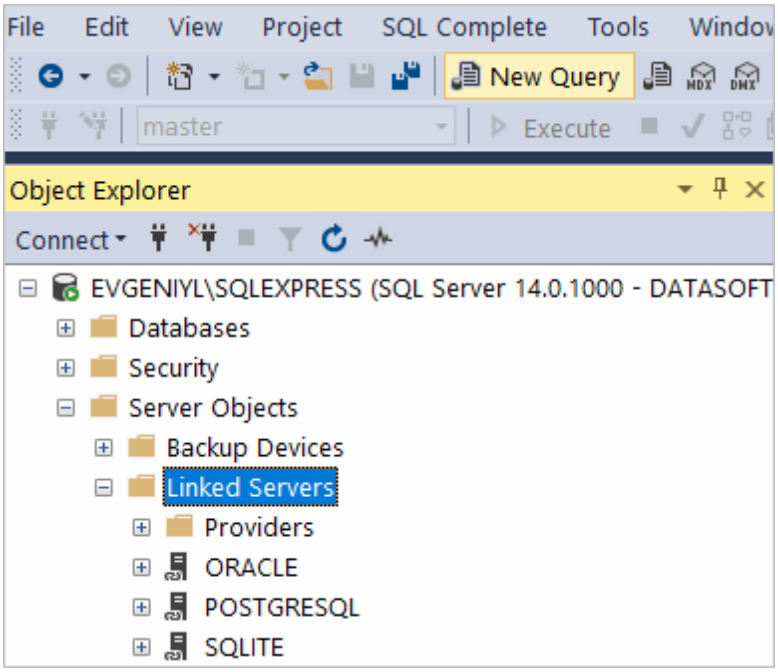
Make sure to select **Microsoft OLE DB Provider for ODBC Drivers**:



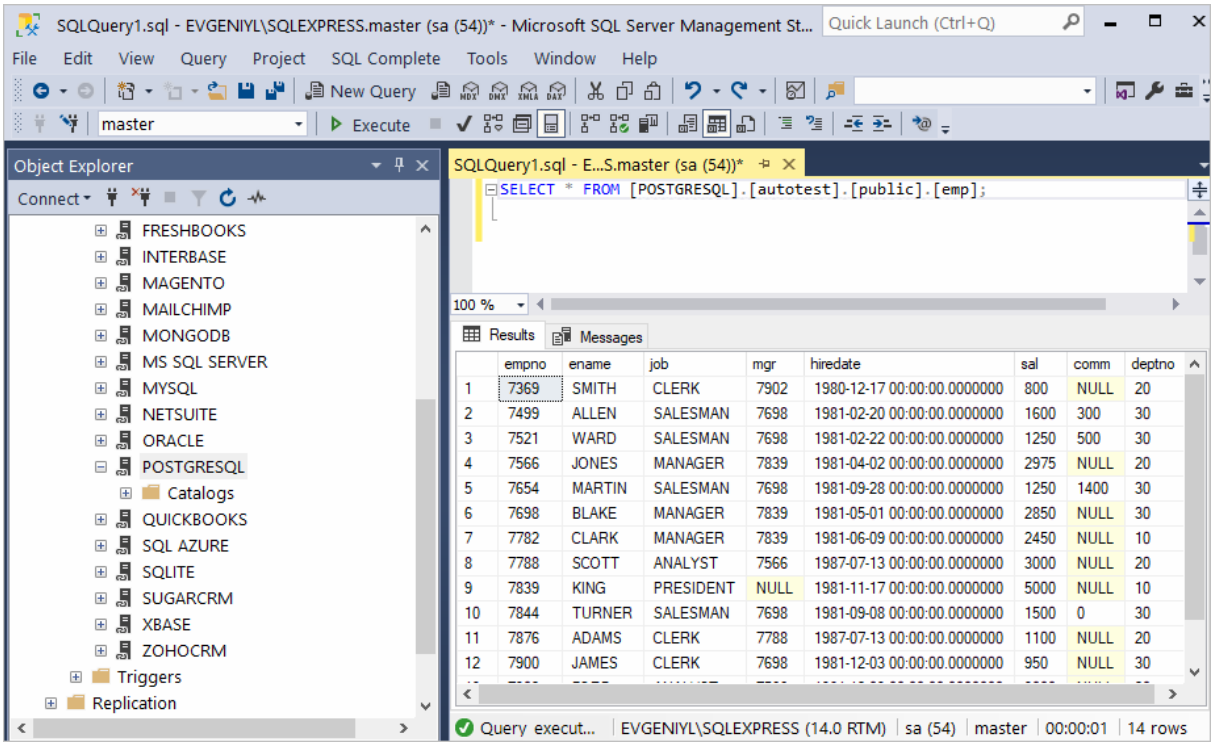
Now you need to input the Linked Server name, e.g. POSTGRESQL. In the Product Name and Data Source fields you need to indicate the System DSN that you've previously created - more info on System DSN setup can be found here.



The PostgreSQL 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

PostgreSQL account you are connected to.

See also

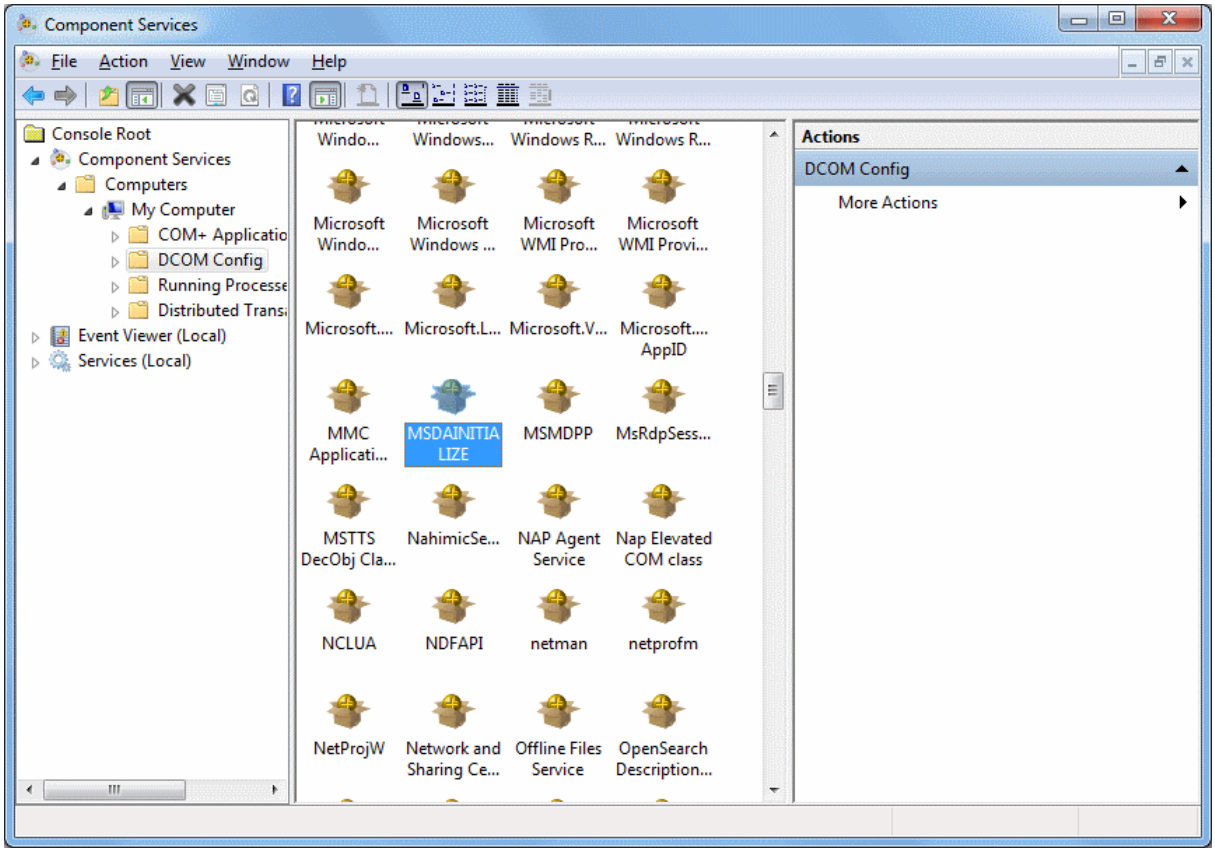
- Troubleshooting SSMS

4.13.2 Troubleshooting in SSMS

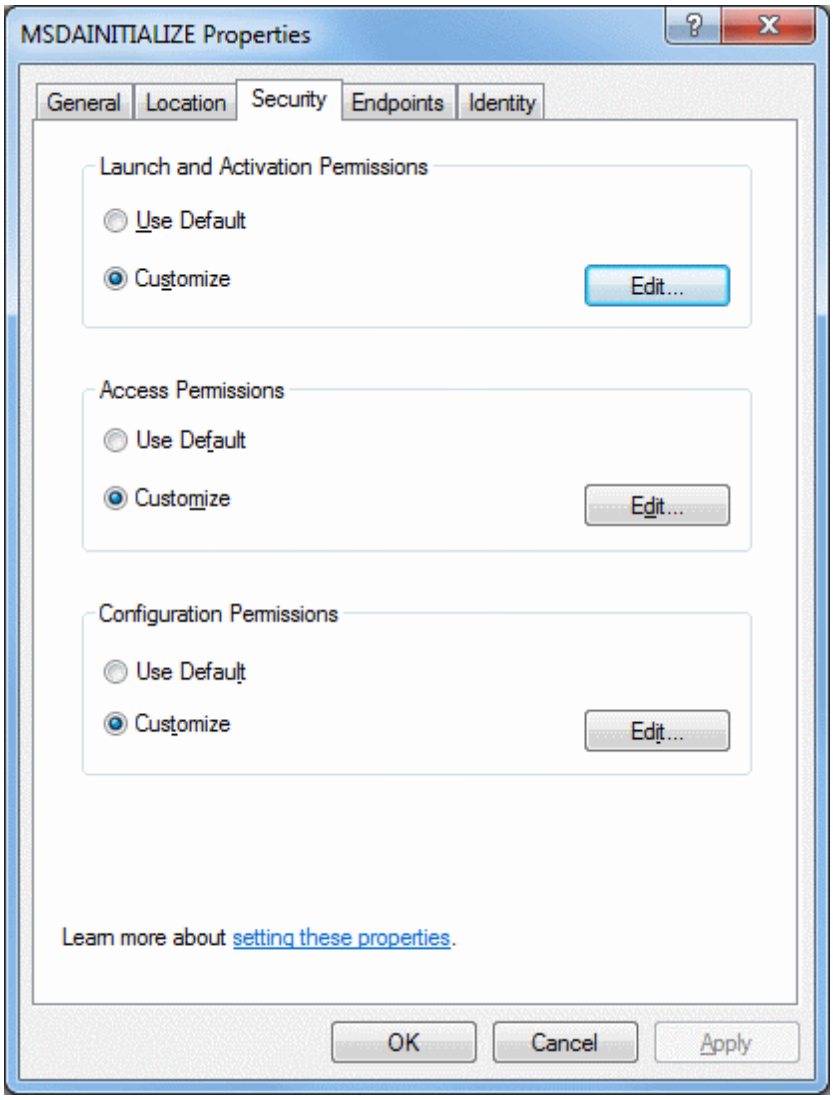
When creating a linked server in SSMS, most errors happen due to security issues with DCOM class MSDAINITIALIZE. We need to alter the DCOM Class MSDAINITIALIZE security settings to make it work.

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:



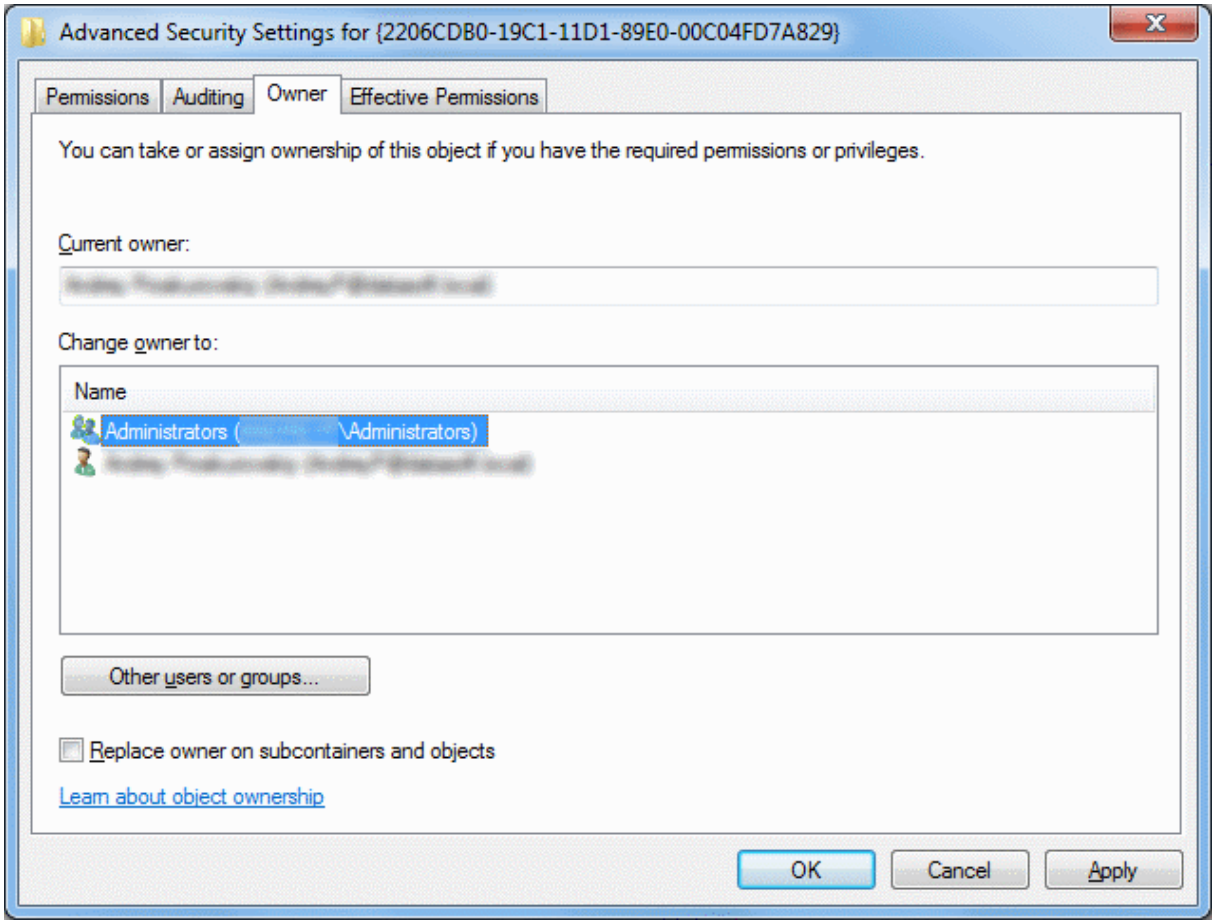
5. Add the Domain User who is accessing the linked server and ‘Allow’ all the permissions available (Local Launch, Remote Launch, Local Activation, Remote Activation). If you are connecting to SQL server using SQL account, you need to provide this permission to the account under which the SQL service 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-

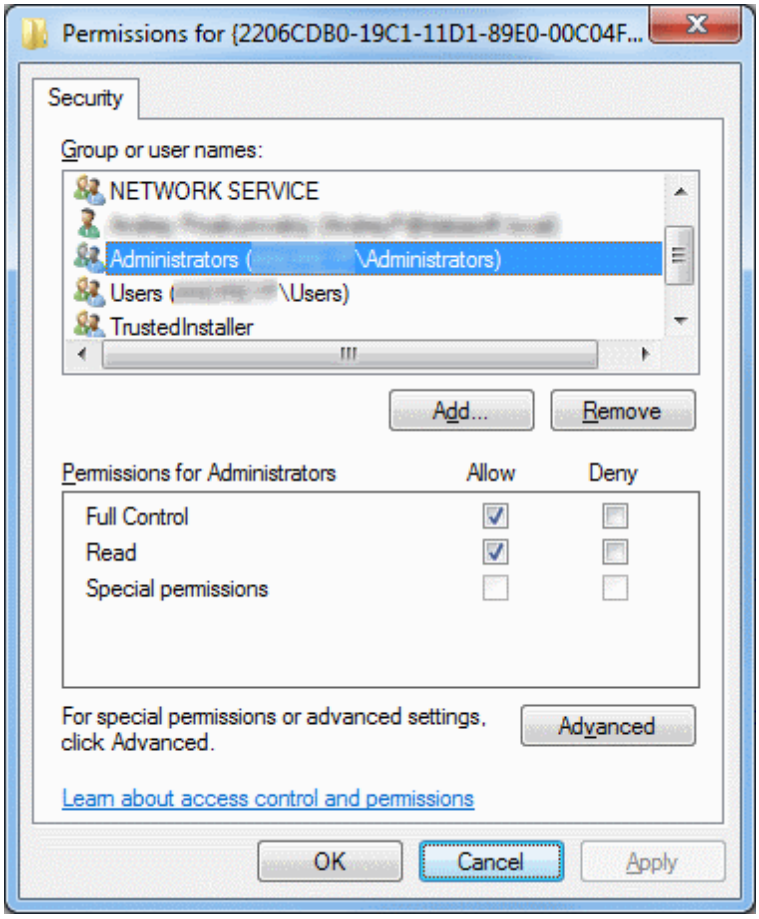
11D1-89E0-00C04FD7A829}

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

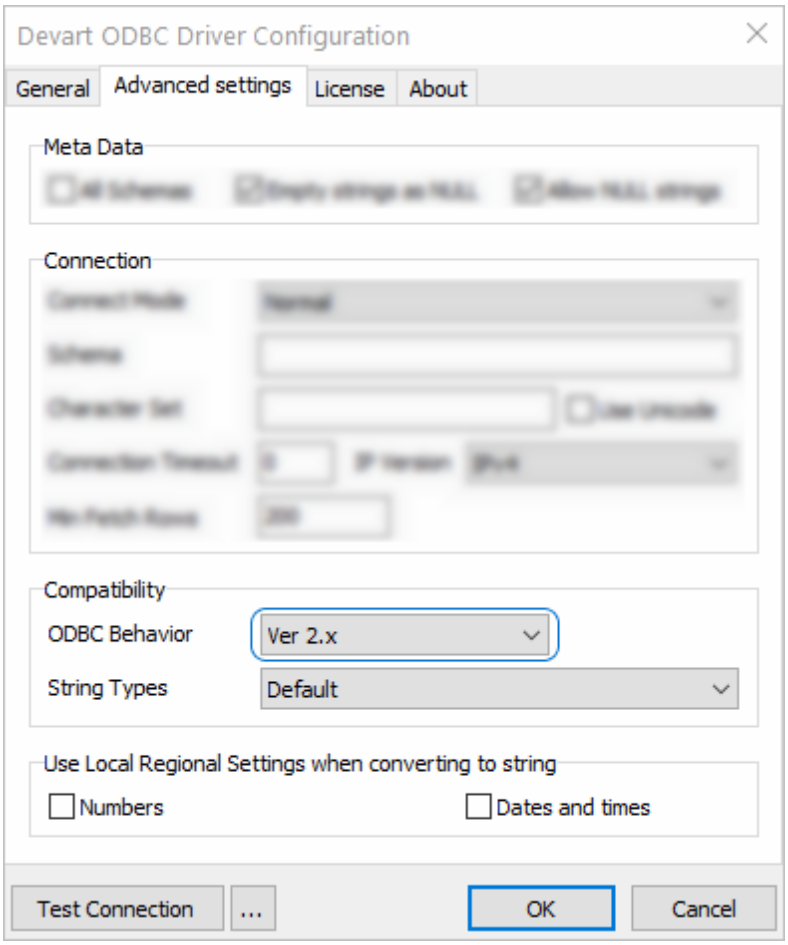
- Error message when you try to create an instance of an OLE DB provider in SQL Server: "Cannot create an instance of OLE DB provider"

4.14 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 PostgreSQL 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.



4.15 Using in Tableau

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

- Using in Tableau
- Troubleshooting in Tableau on macOS

4.15.1 Using in Tableau

Importing PostgreSQL Data Into Tableau Through an ODBC Connection

This article explains to establish and ODBC connection to PostgreSQL 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 PostgreSQL. Alternatively, if you have not created a DSN, you can choose the **Driver** option and select Devart ODBC Driver for PostgreSQL from the drop-down.
5. Click **Connect**.
6. After a successful connection, click **Sign in**.
7. Select the needed database and schema in PostgreSQL.
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.

4.15.2 Troubleshooting in Tableau on macOS

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, create a Tableau Datasource Customization (`.tdc`) file in '*Users\[your name]\Documents\My Tableau Repository*

Datasources\' — for example, *devart-postgresql.tdc*, and add the following capabilities to the file:

```
<?xml version='1.0' encoding='utf-8' ?>
<connection-customization class='genericodbc' enabled='true' version='1.0'>
<vendor name='PostgreSQL' />
<driver name='Devart ODBC Driver for PostgreSQL' />
    <customizations>
          <customization name='CAP_ODBC_BIND_SUPPRESS_WIDE_CHAR' value='yes'
    </customizations>
</connection-customization>
```