



# User Guide: NeoDB for PostgreSQL



## User Guide: NeoDB for PostgreSQL

Version 1.1.0.15

---



**Confidentiality**

This document contains confidential information and materials provided by Neosoft Technologies. The reader, acknowledge that any information present inside this document is strictly confidential and agree to not reveal those information and materials to any other person or entity

Neosoft Technologies Inc.  
4800 rue de Rouen, suite 230  
Montréal, QC H1V 3T4  
Without Fee: 1-866-636-7381  
Tel : (418) 948-8324  
Fax : (418) 948-8325  
<http://www.neosoft.ca>



## HISTORY

Version	Date	Writer	Modification
1.1.0.15	12/10/2022	Cyril Gambini	Added details about licensing
1.1	15/03/2022	Raphael Fortin	Updated Authentication methods
1.0	12/05/2020	Dave Dufour	User Guide Creation



## TABLE OF CONTENTS

<b>1</b>	<b>About .....</b>	<b>5</b>
<b>2</b>	<b>Installation Guide.....</b>	<b>6</b>
<b>3</b>	<b>Licensing .....</b>	<b>7</b>
<b>4</b>	<b>VI's Details .....</b>	<b>8</b>
<b>5</b>	<b>Custom Errors .....</b>	<b>11</b>
<b>6</b>	<b>Troubleshooting .....</b>	<b>12</b>
6.1	Cannot connect to the database .....	12
6.1.1	pg_hba.conf .....	12
6.1.2	postgres.conf.....	13



# 1 About

A database is an organized collection of data, generally stored and accessed electronically from a computer system. There are many databases available. One of the most popular is PostgreSQL. That database is an open source object-relational database. By consequent, PostgreSQL does not require any licence or investment to start working with. Also, because it is open source, it is easy to find information or documentation in books or on the Internet. Those are the reasons that led us to develop this driver.

This driver communicates with the PostgreSQL databases (v9.4 and upper) through TCP/IP. It is possible to use it on Windows (7 and later), Mac OS X (10 and later), Linux and also on NI's real-time targets (compatible with LabVIEW RT running on Pharlaps, VxWorks, Linux RT 32 and 64 bits OSes).

It supports trust, clear text, md5 and SHA256-scam encryption for user passwords.

The driver has been tested with PostgreSQL V12 on Windows 10 and PostgreSQL V9.4 on Linux RT.

This guide explains how to install the toolkit, contains a description of the different VIs present in the library, a description of the custom errors and a troubleshooting section.

Note: The library was developed with LabVIEW 2015 32-bit and is compatible with higher versions.

## 2 Installation Guide

This section contains the procedure to install the NeoDB for PostgreSQL Toolkit using VI Package Manager.

- 1- Download the most recent version of NeoDB for PostgreSQL.vip
- 2- Run VI Package Manager (VIPM) **as administrator**
- 3- Select menu file -> Open Package Files(s)
- 4- Select the NeoDB for PostgreSQL.vip downloaded
- 5- Press on Install
- 6- Follow the steps
- 7- At the end of the installation, restart LabVIEW

Upon restarting LabVIEW, an evaluation window should show and indicate that the toolkit come with 15 days of evaluation.

Two examples should have been installed (LabVIEW menu 'Help' -> 'Find Examples ...') and the *NeoDB for PostgreSQL* palette should appear in "Addons" section.



### **3 Licensing**

*NeoDB for PostgreSQL* is offered as a **development** license.

A valid license must be purchased for every computer that will be used to develop an application using the *NeoDB for PostgreSQL* code.

Once an application is compiled, it can be freely used and distributed without additional fees. This licence is perpetual for the version purchased.

## 4 VIs Details

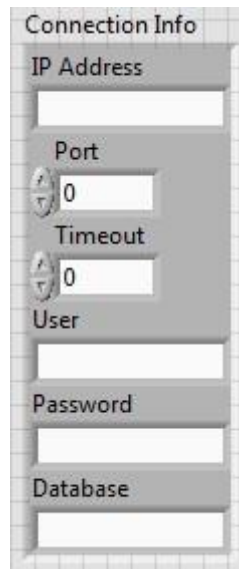
This section will show and describe the different VIs of the library.

### Constructor.vi:



#### VI Description:

This VI should be used at the beginning of the software or anytime the user wants to establish a new connection. The "Connection Info" cluster will look like the next image.



IP Address: Address of the location of the database PostgreSQL.

Port: TCP Port to use. By default, it is port 5432.

Timeout: Maximum time to connect to the database in milliseconds before generating a connection timeout. By default, it is set to 10 000ms.

User: Username to use to login in the database.

Password: User's password to use to login in the database.

Database: Name of the database the user wants to connect to.

### Connect.vi:

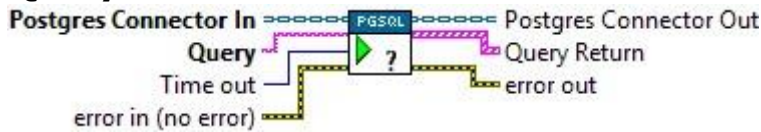


#### VI Description:

Once the information is configured using the "Constructor.vi", this VI shall be used to establish a connection with the database.



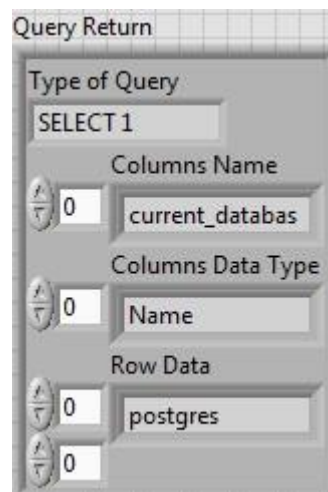
### Query.vi:



#### VI Description:

This VI is used to send a query or queries and to retrieve the information. It is possible to: SELECT, UPDATE, DELETE, UPDATE, MOVE, FETCH, COPY , CREATE TABLE and more.

After a query has been sent and processed, the server will return some information that will be stored in an array of information. Below is an image of the array and a description of each output.



**Type of Query:** Return the type of the query. For example, if the query is a "SELECT", it will be possible to find the keyword "SELECT" associated with the number of elements found.

**Columns Name:** Contain the name of the columns according to the query.

**Columns Data Type:** Return the data type of the data selected.

**Row Data:** Return the value of the data selected

**Note:** It is possible that one or more output could be empty. It depends on the type of query that has been sent to the server. For example, a pg\_sleep would only return the "Type of Query", a "Columns Name" and the "Columns Data Type" but no "Raw Data".

In the other cases, like DELETE, the "Query Return" will only send back the type of command executed.

**Note:** It is possible to enter multiple commands one after another during the same query. Just remember to add ";" at the end of a command. For example,

```
INSERT INTO public."test" ("id") VALUES (2);
SELECT * FROM public."test";
```

The timeout input is used to define the maximum time allowed for the server to send a response. By default, the timeout is 10 seconds, but can be changed by the user.

### Cancel\_Query.vi:

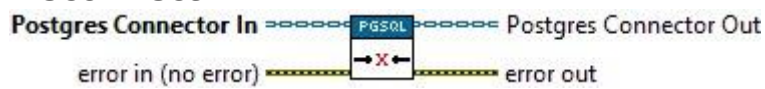


#### VI Description:

This VI is used to cancel the query currently being processed by the database server.

**Note:** It is important to have established a connection first, because this VI reuse the same Process ID and Secret Key given during the initial connection.

### Disconnect.vi:



#### VI Description:

This VI is used to close the TCP connection with the database.

## 5 Custom Errors

This section contains the information about custom errors generated by the driver.

<b>Error Number</b>	<b>Error Type</b>	<b>Possible Error</b>	<b>Possible Solutions</b>
-8000	Database Connection	<p>There is an error in the parameters of "Connection Info" cluster.</p> <p>The user doesn't have the right to connect to the database</p> <p>There is an error in the configuration files.</p>	<p>Verify and enter valid connection parameters.</p> <p>Verify that the user as the right to connect to the database</p> <p>Verify the settings in the pg_hba.conf and postgres.conf files</p>
-8001	Query	There is an error in the query	The error string should return the location of the problem. Correct it and resend the query
-8002	Query	The query is empty	Send a nonempty query.

## 6 Troubleshooting

### 6.1 Cannot connect to the database

#### 6.1.1 pg\_hba.conf

By default, the file is located in C:\Program Files\PostgreSQL\VERSION\data on Windows and in the installation folder on Linux.

To rectify the situation, it will be needed to add a line in the "IPv4 local connections section". That line should have a type name, database name, username, address, and a method.

It is possible to limit the access of the connection to certain database, user, or IP address.

The method refers to the authentication requirement to be able to connect to the server. The driver supports encrypted password (md5 or scram-sha-256), clear password (password) or no password (trust).

If the user has no password it is important to select "trust" so the server won't send back an error. Otherwise you can choose "scram-sha-256", "md5" or "password".

A more detailed description of each element can be found in the pg\_hba.conf file.

For reference, the section of the file should be similar to:

```
# TYPE DATABASE USER ADDRESS METHOD
# IPv4 local connections:
host all all 127.0.0.1/32 md5
host all all 192.168.0.65/32 md5
```

**Note:** If the database is installed locally. Be sure the address used to connect to the database and the address in the pg\_hba.conf file is the same.

Example 1: If you use the default address for localhost "127.0.0.1", there should be a line "host all all 127.0.0.1/32 md5" in the pg\_hba.conf file.

Example 2: The same thing goes if you use the real address of the machine on the network. If the local address is 192.168.1.34, the connection IP Address should be 192.168.1.34 and the line in the pg\_hba.conf file should look like "host all all 192.168.1.34/32 md5".

If you are unsure of the settings, you can add both settings to the list.



### **6.1.2 postgres.conf**

By default, the file is located in C:\Program Files\PostgreSQL\VERSION\data on Windows and in the installation folder on Linux.

To enable remote connection to the database, you need to specify the allowed requestors with the "listen\_addresses" token. By default, it is 'localhost'. It's possible to listen on one or more IP addresses using '\*' or a comma-separated list of IP addresses. Refer to the information contained in "postgres.conf" for more information.