

Greenplum Database 4.3 Connectivity Tools for Windows

Greenplum provides database drivers and a C API for connecting to Greenplum Database. In this version 4.3 distribution, the following connectivity tools are provided for Windows. See the *Greenplum Database Release Notes* for the list of currently supported platforms for the Connectivity Tools.

- [psqlODBC](#)
- [PostgreSQL JDBC Interface](#)
- [libpq](#)

Note: If your Java application connects to Greenplum Database with Kerberos authentication, see “[Configuring a Client System for Kerberos Authentication.](#)”

psqlODBC

psqlODBC is the official PostgreSQL ODBC Driver. The driver is currently maintained by a number of contributors to the PostgreSQL project at <http://pgfoundry.org/projects/psqlodbc>. It is developed and supported through the psql-odbc@postgresql.org mailing list. psqlODBC is released under the Library General Public Licence, or LGPL.

PostgreSQL JDBC Interface

The PostgreSQL JDBC interface is the official PostgreSQL JDBC driver. The driver is currently maintained by a number of contributors to the PostgreSQL project at <http://jdbc.postgresql.org>. JDBC is a core API of Java 1.1 and later. It provides a standard set of interfaces to SQL-compliant databases. PostgreSQL provides a type 4 JDBC driver. Type 4 indicates that the driver is written in Pure Java, and communicates in the database system’s own network protocol. Because of this, the driver is platform independent; once compiled, the driver can be used on any system. The PostgreSQL JDBC Interface has not been modified from the original PostgreSQL distribution.

libpq

libpq is the C application programmer’s interface (API) to PostgreSQL (and Greenplum Database). libpq is a set of library functions that allow client programs to pass queries to the PostgreSQL backend server and to receive the results of these queries.

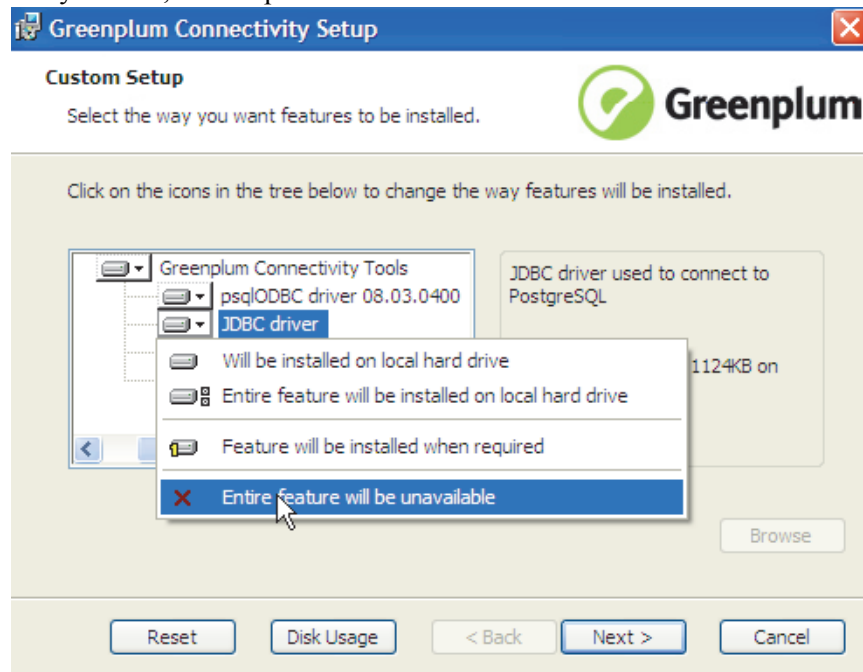
For more information on using `libpq`, see [libpq - C Library](#) in the PostgreSQL documentation.

Running the Connectivity Tools Installer

You can choose to install all connectivity tools or a subset. After installing, some connectivity tools require additional installation or configuration steps.

To install the Greenplum Database Connectivity Tools

1. Download the `greenplum-connectivity-4.3.x.x-WinXP-x86_32.msi` package from [Pivotal Network](#).
2. Double-click on the `greenplum-connectivity-4.3.x.x-WinXP-x86_32.msi` package to launch the installer.
3. Click **Next** on the Welcome screen.
4. Click **I Agree** on the License Agreement screen.
5. On the **Custom Setup** screen, deselect the components you do not want to install. By default, all components will be installed.



6. By default, the Greenplum Database connectivity tools will be installed into `C:\Program Files\Greenplum\greenplum-drivers-4.3.x.x`. Click **Browse** to choose another location.
7. Click **Next** when you have chosen the components and install path you want.
8. Click **Install** to begin the installation.
9. Click **Finish** to exit the installer.

About Your Installation

Your Greenplum Database connectivity tools installation contains the following files and directories:

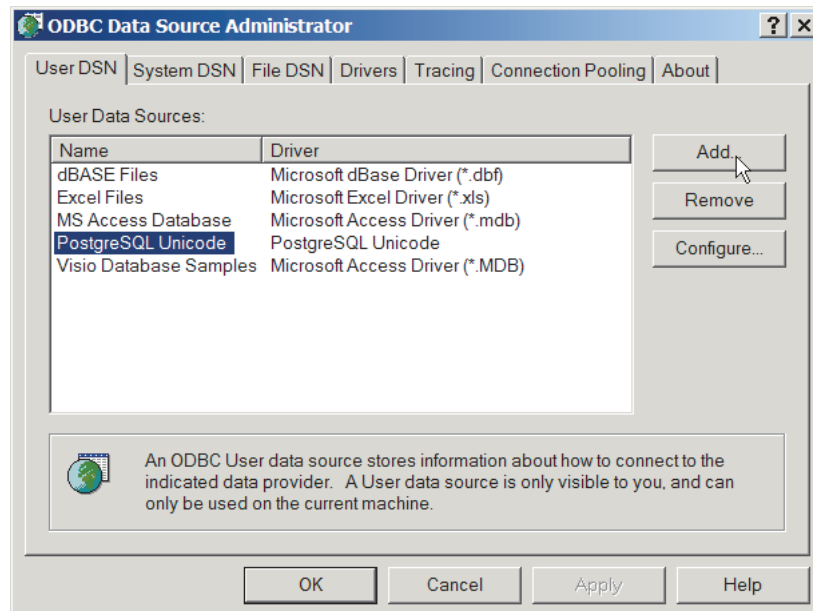
- `GP_ConnectWin.pdf` — the documentation file for connectivity tools
- `greenplum_connectivity_path.bat` — script to set environment variables. This script is run automatically as part of the installation.
- **drivers** — PostgreSQL ODBC and JDBC database drivers
- **include** — libpq C header files
- **lib** — shared object files and other library files to support the drivers

Creating an ODBC Data Source

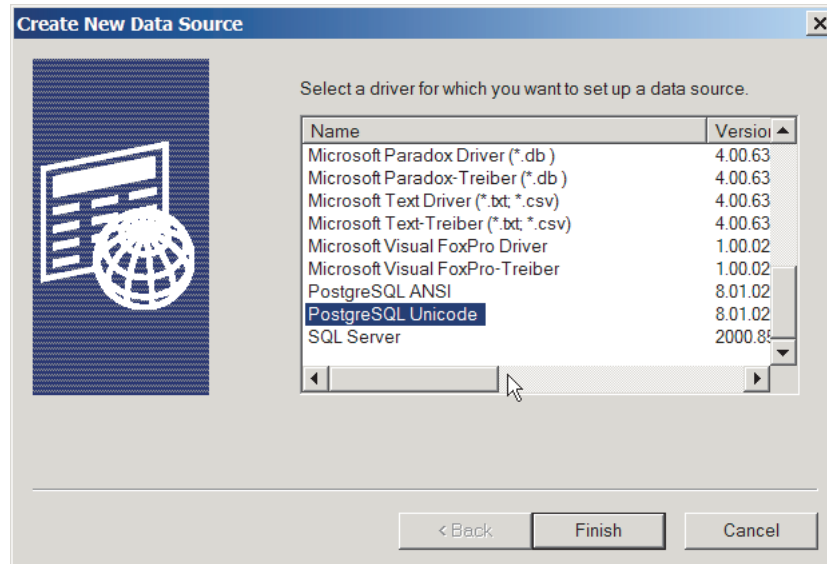
A data source configures your ODBC driver to connect to a particular database. For Greenplum Database you should configure your data source to connect to the master instance.

To configure a data source

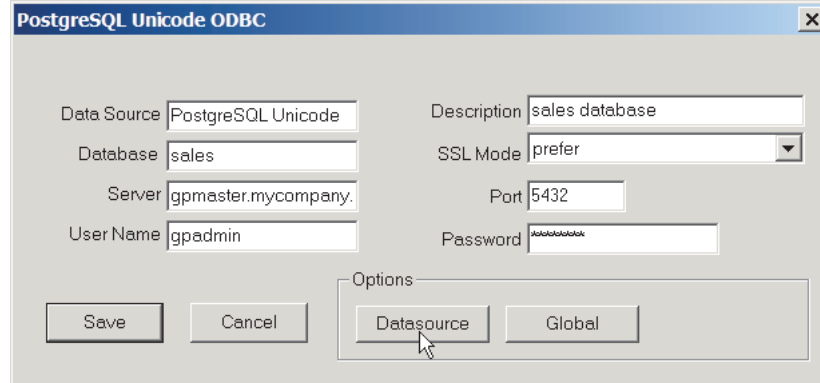
1. In Windows Explorer, go to `C:\Control Panel`.
2. Double-click the **Administrative Tools** icon.
3. Double-click **Data Sources (ODBC)** to open the ODBC Data Source Administrator.
4. Select **PostgreSQL Unicode** and click **Add** to add a new data source.



- For the driver, **PostgreSQL Unicode** is the recommended choice. Unicode (UTF-8) is the default database encoding for Greenplum Database. There is also a **PostgreSQL ANSI** driver which can handle some multi-byte character sets and LATIN character sets.



- Fill in the connection information for your database (on the Greenplum Database master instance).



7. Click **Datasource** to access the Advanced Options. the following settings are recommended for Greenplum Database.:

8. Click **Page2**. the following settings are recommended for Greenplum Database:

9. Click **OK**.
10. Click **Save**.

Configuring the PostgreSQL JDBC Driver

The PostgreSQL JDBC driver is installed by the connectivity tools installer into `C:\Program Files\Greenplum\greenplum-drivers-4.3.x.x\drivers\jdbc`. In order to use the driver, you must add its jar files to your `CLASSPATH` environment variable.

To edit the `CLASSPATH` on Windows XP

1. In Windows Explorer, go to `C:\Control Panel`.
2. Double-click the **System** icon.
3. On the **Advanced** tab, click **Environment Variables** (bottom).
4. Find the `CLASSPATH` environment variable and double-click on it to edit it (if not there, click **New** to create it).
5. Add the path to the JDBC driver jar file directory at the end of the current class path. For example:

```
C:\Program
Files\Java\jdk1.5.0_02\bin;greenplum-drivers-4.3.x.x\drivers\
jdbc\*
```
6. Click **OK**.

About `greenplum_connectivity_path.bat`

The installer automatically creates the necessary environment variables needed for the connectivity tools. As a convenience, the script `greenplum_connectivity_path.bat` is provided in your connectivity tools installation directory. This script sets the following environment variables:

GPHOME_CONNECTIVITY — The installation directory of the Greenplum Database connectivity tools.

PATH – To allow access to the connectivity tools from any directory, the `PATH` environment variable is modified to add `GPHOME_CONNECTIVITY\bin` and `GPHOME_CONNECTIVITY\lib`.

If you do not need to modify these environment variables, you do not need to run this script.

Configuring a Client System for Kerberos Authentication

If your JDBC application uses Kerberos authentication to connect to your Greenplum Database, your client system must be configured to use Kerberos authentication. If you are not using Kerberos authentication to connect to a Greenplum Database, Kerberos is not needed on your client system.

- [Requirements](#)
- [Setting Up Client System with Kerberos Authentication](#)
- [Running a Java Application](#)

For information about enabling Kerberos authentication with Greenplum Database, see the chapter “Setting Up Kerberos Authentication” in the *Greenplum Database Administrator Guide*.

Requirements

The following are requirements to connect to a Greenplum Database that is enabled with Kerberos authentication from a client system with a JDBC application.

- [Prerequisites](#)
- [Required Software on the Client Machine](#)
- [User Environment Variables](#)

Prerequisites

- Kerberos must be installed and configured on the Greenplum Database master host.

Important: Greenplum Database must be configured so that a remote user can connect to Greenplum Database with Kerberos authentication. Authorization to access Greenplum Database is controlled by the `pg_hba.conf` file. For details, see “Editing the `pg_hba.conf` File” in the *Greenplum Database Administration Guide*, and also see the *Greenplum Database Security Configuration Guide*.

- The client system requires the Kerberos configuration file `krb5.conf` from the Greenplum Database master.
- The client system requires a Kerberos keytab file that contains the authentication credentials for the Greenplum Database user that is used to log into the database.
- The client machine must be able to connect to Greenplum Database master host. If necessary, add the Greenplum Database master host name and IP address to the system `hosts` file. On Windows 7 systems, the `hosts` file is in `C:\Windows\System32\drivers\etc\`.

Required Software on the Client Machine

- The Kerberos `kinit` utility is required on the client machine. The `kinit.exe` utility is available with Kerberos for Windows. Greenplum Database supports Kerberos for Windows version 3.2.2. Kerberos for Windows is available from the Kerberos web site <http://web.mit.edu/kerberos/>.

Note: When you install the Kerberos software, you can use other Kerberos utilities such as `klist` to display Kerberos ticket information.

- **For Windows or Linux connectivity:** Java JDK
Java JDK 1.7.0_21 is supported on Windows.

User Environment Variables

- `KRB5CCNAME` - A Kerberos environment variable that specifies the location of the Kerberos ticket cache. For example, in “[Setting Up Client System with Kerberos Authentication](#),” the ticket cache is `C:\Users\gpadmin\cache.txt`.
- `JAVA_HOME` is set to the installation directory of the supported Java JDK.
- Ensure that in the batch file `greenplum_connectivity_path.bat`, the `GP_JDBC_JARFILE` environment variable specifies the location of the Greenplum Database JDBC driver `postgresql-8.4-701.jdbc4.jar`.

Setting Up Client System with Kerberos Authentication

To connect to Greenplum Database with Kerberos authentication requires a Kerberos ticket. On client systems, tickets are generated from Kerberos keytab files with the `kinit` utility and are stored in a cache file.

1. Install a copy of the Kerberos configuration file `krb5.conf` from the Greenplum Database master. The file is used by the Greenplum Database client software and the Kerberos utilities.

Rename `krb5.conf` to `krb5.ini` and move it to the Windows directory. On Windows 7, the Windows directory is `C:\Windows`.

If needed, add the parameter `default_ccache_name` to the `[libdefaults]` section of the `krb5.ini` file and specify location of the Kerberos ticket cache file on the client system.

2. Obtain a Kerberos keytab file that contains the authentication credentials for the Greenplum Database user.
3. Run `kinit` specifying the keytab file to create a ticket on the client machine. For this example on a Windows system, the keytab file `gpdb-kerberos.keytab` is in the same directory as `kinit.exe`. The ticket cache file is in the Windows `gpadmin` user home directory.

```
> kinit -k -t gpdb-kerberos.keytab
   -c C:\Users\gpadmin\cache.txt
      gpadmin/kerberos-gpdb@KRB.GREENPLUM.COM
```

Running a Java Application

Accessing Greenplum Database from a Java application with Kerberos authentication uses the Java Authentication and Authorization Service (JAAS)

1. Create the file `.java.login.config` in the user home folder.

For example, on a Linux system, the home folder is similar to `/home/gpadmin`.

Add the following text to the file:

```
pgjdbc {
    com.sun.security.auth.module.Krb5LoginModule required
    doNotPrompt=true
    useTicketCache=true
    ticketCache = "C:\\Users\\gpadmin\\cache.txt"
    debug=true
    client=true;
};
```

2. Create a Java application that connects to Greenplum Database using Kerberos authentication and run the application as the user.

This example database connection URL uses a PostgreSQL JDBC driver and specifies parameters for Kerberos authentication.

```
jdbc:postgresql://kerberos-gpdb:5432/mytest?
  kerberosServerName=postgres&jaasApplicationName=pgjdbc&
  user=gpadmin/kerberos-gpdb
```


The parameter names and values specified depend on how the Java application performs Kerberos authentication.

Copyright © 2014 Pivotal Software, Inc. All rights reserved.

Pivotal Software, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." PIVOTAL SOFTWARE, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Pivotal software described in this publication requires an applicable software license.

All trademarks used herein are the property of Pivotal or their respective owners.