

## Greenplum Database 4.3 Connectivity Tools for UNIX

---

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:

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

These are supported on the following platforms:

- RedHat Enterprise Linux 4.0
- RedHat Enterprise Linux 5.0 (64-bit only)
- Solaris 10 x86
- Solaris 10 for SPARC
- Solaris 9 for SPARC (32-bit only)
- Solaris 8 for SPARC (32-bit only)
- Mac OS X
- SUSE Linux Enterprise Server 10 (64-bit only)

**Note:** If your Java application on RedHat Enterprise Linux connect 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 [pgsql-odbc@postgresql.org](mailto:pgsql-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.

---

## Installing Connectivity Tools

The Greenplum Database connectivity tools installer copies the drivers and `libpq` API to your system. After installation, some connectivity tools require additional configuration steps.

### To install the Greenplum Database connectivity tools

1. Download the appropriate `greenplum-connectivity-4.3.x.x-PLATFORM.bin.zip` installer package for your platform from [EMC Download Center](#).
2. Unzip the installer package:
 

```
unzip greenplum-connectivity-4.3.x.x-PLATFORM.bin.zip
```
3. Run the installer:
 

```
/bin/bash greenplum-connectivity-4.3.x.x-PLATFORM.bin
```
4. The installer will prompt you to accept the license agreement and to provide an installation path. For the installation path, be sure to enter an absolute path if you choose not to accept the default location (for example, `/home/mydir/gp-drivers`). The connectivity tools are installed by default into `/usr/local/greenplum-connectivity-4.3.x.x`.  
This installation directory is referred to as `$GPHOME_CONNECTIVITY`.

### About Your Installation

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

- **GPConnectUnix.pdf** — documentation for the connectivity tools package
- **drivers** — PostgreSQL ODBC and JDBC database drivers
- **greenplum\_connectivity\_path.sh** — environment variables
- **include** — `libpq` C header files

- **lib** — libpq library files

---

## Configuring Greenplum Database Drivers for Unix

The PostgreSQL ODBC drivers require a data source definition (DSN) file and a compatible driver manager. Also, you must set environment variables in `greenplum_connectivity_path.sh` to specify the correct driver and driver manager files.

For JDBC, you must add the driver's JAR files to your path.

### Configuring the PostgreSQL ODBC Driver

The `$GPHOME_CONNECTIVITY/drivers/odbc` directory contains both drivers and compatible driver manager program files in subdirectories organized by driver version/driver manager. For example, driver files for `psqlodbc` version 08.02.0500 compiled with `unixODBC` driver manager 2.2.12 are located in the following directory:

```
$GPHOME_CONNECTIVITY/drivers/odbc/psqlodbc-08.02.0500/unixodbc-2.2.12
```

The `$GPHOME_CONNECTIVITY/drivers/odbc` directory contains only the driver/driver manager combinations that are supported for a given platform. If you need a different combination, contact Greenplum customer support to submit a request.

### To configure the PostgreSQL ODBC Driver

1. In the directory `$GPHOME_CONNECTIVITY/drivers/odbc`, locate the correct driver and driver manager. For example, if you are configuring `psqlodbc-08.02.0400` with the 64-bit Data Direct Driver Manager 5.2, the correct program files are found in the following directory:

```
$GPHOME_CONNECTIVITY/drivers/odbc/psqlodbc-08.02.0400/datadirect-52_64
```

2. Edit `greenplum_connectivity_path.sh` and set the following variables:

```
GP_ODBC_DRIVER = psqlodbc-VERSION
GP_ODBC_DRIVER_MANAGER = DRIVER_MANAGER_VERSION
```

To specify the versions, use the same values used in the directory naming. For example:

```
GP_ODBC_DRIVER = psqlodbc-08.02.0400
GP_ODBC_DRIVER_MANAGER = datadirect-52_64
```

3. After editing `greenplum_connectivity_path.sh`, source it as the correct user to make the changes active. For example:

```
source greenplum_connectivity_path.sh
```

4. Create a data source definition (DSN). User DSNs are usually stored in the Greenplum user's (`gpadmin`) home directory in a file named `.odbc.ini` (note the leading dot). Here is an example `.odbc.ini` file:

```
[Greenplum]
```

```

Description = PostgreSQL driver for Greenplum
Driver =
greenplum-connectivity-4.3.x.x/drivers/odbc/psqlodbc-02.08.0500
/unixodbc-2.2.14/psqlodbcw.so
Trace = 1
Debug=1
Database = template1
Servername = gpmaster_hostname
UserName = gpadmin
Password = dbpassword_for_gpadmin
Port = 5432
ReadOnly = No
RowVersioning = No
DisallowPremature = No
ShowSystemTables = Yes
ShowOidColumn = No
FakeOidIndex = No
useDeclareFetch = 1
Fetch = 4096
UpdatableCursors = No
Protocol = 7.4-1

```

### Verifying the PostgreSQL ODBC Driver

After installing and configuring the ODBC driver, you can verify its working condition with a simple test such as passing a command from iSQL. If you use iSQL for verification purposes, make sure you use a version from the unixODBC driver manager 2.2.14 or later, and specify the `-3` option. Also, you must specify the driver manager version in your `PATH` and `LD_LIBRARY_PATH` environment variables.

#### To verify the PostgreSQL ODBC Driver with iSQL:

1. Open `greenplum_connectivity_path.sh` for edit.
2. To the `PATH` variable, add `<path_to_unixodbc-2.2.14>/bin`. For example:  
`PATH=${GPHOME_CONNECTIVITY}/bin:${GPHOME_CONNECTIVITY}/drivers/odbc/psqlodbc-08.02.0500/unixodbc-2.2.14/bin:${PATH}`
3. To the `LD_LIBRARY_PATH` variable (`DYLD_LIBRARY_PATH` on OSX), add `<path_to_unixodbc-2.2.14>/ext/lib`. For example:  
`LD_LIBRARY_PATH=${GPHOME_CONNECTIVITY}/drivers/odbc/${GP_ODBC_DRIVER}/${GP_ODBC_DRIVER_MANAGER}:/:${GPHOME_CONNECTIVITY}/lib:${GPHOME_CONNECTIVITY}/drivers/odbc/psqlodbc-08.02.0500/unixodbc-2.2.14/ext/lib:${LD_LIBRARY_PATH}`
4. After editing `greenplum_connectivity_path.sh`, source it as the correct user to make the changes active. For example:  
`source greenplum_connectivity_path.sh`

5. Run `isql`, specifying the database name and the `-3` option. For example:
 

```
isql -3 <database_name>
```

### Configuring the PostgreSQL JDBC Driver

The PostgreSQL JDBC drivers are installed by the client tools installer into `greenplum-connectivity-4.3.x.x/drivers/jdbc`. In order to use a driver, you must specify the correct JAR file in the `GP_JDBC_DRIVER` variable provided in `greenplum_connectivity_path.sh`.

#### To configure the PostgreSQL JDBC Driver

1. In the directory `$GPHOME_CONNECTIVITY/drivers/jdbc`, locate the correct JAR file. For example, if your application requires a JDBC 3.0-compliant driver, use the following JAR file:

```
$GPHOME_CONNECTIVITY/drivers/jdbc/postgresql-8.1-407.jdbc3.jar
```

2. Edit `greenplum_connectivity_path.sh` and set `GP_JDBC_DRIVER` to the correct JAR file name:

```
GP_JDBC_JARFILE=postgresql-8.1-407.jdbc3.jar
```

3. After editing `greenplum_connectivity_path.sh`, source it as the correct user to make the changes active. For example:

```
source greenplum_connectivity_path.sh
```

### Setting Environment Variables

The `greenplum_connectivity_path.sh` file is provided in your connectivity tools installation directory. It has the following environment variable settings:

**GPHOME\_CONNECTIVITY** — The installation directory of the Greenplum Database connectivity tools.

**PATH** — The path to additional library files needed for the drivers.

**PYTHONPATH** — The path to Python library files needed for ODBC drivers.

**CLASSPATH** — The path to the selected JAR file for the JDBC driver.

This file also stores the variables to specify the selected ODBC and JDBC driver files:

**GP\_ODBC\_DRIVER** — Set to the name of the selected ODBC driver (default is unset).

**GP\_ODBC\_DRIVER\_MANAGER** — Set to the name of the selected driver manager (default is unset).

**GP\_JDBC\_JARFILE** — Set to the name of the JAR file for the selected JDBC driver (default is unset).

You can source this file in your user's startup shell profile (such as `.bashrc` or `.bash_profile`).

For example, you could add a line similar to the following to your chosen profile files (making sure the right install path is used):

```
source
greenplum-connectivity-4.3.x.x/greenplum_connectivity_path.sh
```

After editing the chosen profile file, source it as the correct user to make the changes active. For example:

```
source ~/.bashrc
```

---

## Configuring a Client System for Kerberos Authentication

If your JDBC application on RedHat Enterprise Linux uses Kerberos authentication when it connects 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 “Kerberos Authentication” in the *Greenplum Database System 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 information about managing authorization privileges, see the *Greenplum Database Database Administrator Guide*. For information about the `pg_hba.conf` file, see the Postgres documentation: <http://www.postgresql.org/docs/8.4/static/auth-pg-hba-conf.html>

- 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 Linux systems, the `hosts` file is in `/etc`.

### Required Software on the Client Machine

- The Kerberos `kinit` utility is required on the client machine. The `kinit` utility is available when you install the Kerberos packages:
  - `krb5-libs`
  - `krb5-workstation`

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

- Java JDK  
Java JDK 1.7.0\_17 is supported on Red Hat Enterprise Linux 6.x.  
Java JDK 1.6.0\_21 is supported on Red Hat Enterprise Linux 5.x.

### User Environment Variables

- Ensure that `JAVA_HOME` is set to the installation directory of the supported Java JDK.
- Ensure `greenplum_connectivity_path.sh` and set `GP_JDBC_DRIVER` to the correct JAR file name:  
`GP_JDBC_JARFILE=postgresql-8.1-407.jdbc4.jar`  
Source the file as the user running the Java application to make the changes active.

### 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.

Install `krb5.conf` in the directory `/etc`.

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, the keytab file `gpdb-kerberos.keytab` is in the the current directory. The ticket cache file is in the `gpadmin` user home directory.

```
> kinit -k -t gpdb-kerberos.keytab
  -c /home/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 = "/home/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 © 2013 GoPivotal, Inc. All rights reserved.**

GoPivotal, 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." GOPIVOTAL, 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.