Greenplum Database 4.3 Connectivity Tools for UNIX

Rev: A09

Updated: April, 2018

Overview

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

The supported platforms include RedHat Enterprise Linux, Solaris, and SUSE Linux Enterprise Server. See the *Greenplum Database Release Notes* for the list of currently supported platforms for the Connectivity Tools.

> **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 mailing list. psqlODBC is released under the Library General Public Licence, or LGPL.

PostgreSQL JDBC Interface

The PostgreSQL JDB interface is the official PostgreSQL JDBC driver. The driver is currently maintained by a number of contributors to the PostgreSQL project at *https://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 the 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* `greenplum-connectivity-4.3.x.x-PLATFORM.bin.zip` installer package for your platform from *Pivotal Network*.
2. Unzip the installer package:

   ```
   unzip greenplum-connectivity-4.3.x.x-PLATFORM.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. The default installation directory for the connectivity tools is `/usr/local/greenplum-connectivity-4.3.x.x`. If you choose to specify a different installation directory, be sure to enter an absolute path (for example, `/home/mydir/gp-drivers`).

About Your Installation

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

- `drivers/` — PostgreSQL ODBC and JDBC database drivers
- `greenplum_connectivity_path.sh` — script that sets up environment variables, including `$GPHOME_CONNECTIVITY`, which identifies the connectivity tools installation directory
- `include/` — libpq and other connectivity header files
- `lib/` — libpq and other library files
- `LICENSE, NOTICE` — license files

Configuring Greenplum Database Drivers for Unix

The PostgreSQL ODBC drivers require an ODBC driver instance, a data source definition (DSN) file, and a compatible driver manager. You must also configure Greenplum Database for ODBC by setting environment variables in `greenplum_connectivity_path.sh` to specify the driver and driver manager versions.

To use the JDBC driver, you must set an environment variable to identify the JDBC driver file. You must also add the driver's JAR files to your application `CLASSPATH`.

Configuring PostgreSQL ODBC in Greenplum Database

The `$GPHOME_CONNECTIVITY/drivers/odbc` directory contains sets of drivers and compatible driver manager program files in a directory hierarchy organized by <driver-version>/<drivermanager-version>. For example, driver files for psqlodbc version 09.02.0100 compiled with unixODBC driver manager 2.2.12 are located in the following directory:

```
$GPHOME_CONNECTIVITY/drivers/odbc/psqlodbc-09.02.0100/unixodbc-2.2.12
```

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

Perform the following procedure to configure the Greenplum Database for PostgreSQL ODBC driver:

1. Navigate to the `$GPHOME_CONNECTIVITY/drivers/odbc`, directory and locate the correct driver and driver manager. For example, if you are configuring psqlodbc-09.02.0100 with the unixODBC driver manager version 2.2.12, the correct program files are found in the following directory:

   ```
   $GPHOME_CONNECTIVITY/drivers/odbc/psqlodbc-09.02.0100/unixodbc-2.2.12
   ```

2. Open the `greenplum_connectivity_path.sh` file in the editor of your choice and set the following environment variables:

   ```
   GP_ODBC_DRIVER = psqlodbc-VERSION
   GP_ODBC_DRIVER_MANAGER = unixodbc-VERSION
   ```

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

   ```
   GP_ODBC_DRIVER = psqlodbc-09.02.0100
   GP_ODBC_DRIVER_MANAGER = unixodbc-2.2.12
   ```

3. Save the file and exit the editor.

Configuring the PostgreSQL ODBC Data Source Name

An ODBC data source name (DSN) for Greenplum Database provides the connection information necessary to access Greenplum Database tables using ODBC. User DSNs are typically located in the Greenplum user's home directory in a file named `.odbc.ini` (note the leading dot).

The first line of the DSN identifies a shortcut name for the data source.

An example PostgreSQL ODBC driver for Greenplum Database `.odbc.ini` file for the `gpadmin` user is reproduced below:

```
[Greenplum]
Description = PostgreSQL driver for Greenplum
Driver = /usr/local/greenplum-connectivity-4.3.24.0/drivers/odbc/
psqlodbc-09.02.0100/unixodbc-2.2.12/psqlodbcw.so
Trace = 1
Debug=1
Database = testdb
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
```

You must replace the `Driver`, `Database`, `Servername`, `UserName`, `Password`, and `Port` settings with those specific to your Greenplum Database deployment.

Note: The example above sets the `Driver` property value to the absolute path of the Greenplum Database PostgreSQL ODBC driver library. Alternatively, you can specify the name of an ODBC driver instance configured in the `/etc/odbcinst.ini` file.

Verifying the PostgreSQL ODBC Driver

After installing and configuring the ODBC driver, verify that it is working with a simple test program such as `isql`. If you use `isql` for verification purposes, make sure that you use a unixODBC driver manager version 2.2.14 or later, and specify the `-3` option when you invoke the command.

To verify the PostgreSQL ODBC Driver for Greenplum Database with isql:

1. If it is not already present on your system, install the `isql` command. This command resides in the operating system `unixODBC` package. You must have superuser permissions to install an operating system package. For example:

   ```
   # yum install unixODBC
   ```

2. Verify that the `isql` command uses version 2.2.14 or later of the unixODBC libraries. For example, to verify the DSN you configured in the previous section:

   ```
   isql --version
   unixODBC 2.2.14
   ```

3. Source the `greenplum_connectivity_path.sh` file to set up your environment. For example:

   ```
   source /usr/local/greenplum-connectivity-4.3.24.0/
   greenplum_connectivity_path.sh
   ```

4. Run the `isql` command, specifying the datasource shortcut name and the `-3` option. For example:

   ```
   isql Greenplum -3
   ```

   If successful, `isql` displays the following output:

   ```
   +---------------------------------------+
   | Connected!                            |
   |                                       |
   | sql-statement                         |
   | help [tablename]                      |
   | quit                                  |
   |                                       |
   +---------------------------------------+
   SQL>
   ```

5. Run the following query to display the name of the current database:

   ```
   select current_database();
   ```

   The output identifies the name of the database that you specified in the `odbc.ini` file `Database` value setting. For example:

   ```
   +------------------------------------------------------------------+
   | current_database                                                 |
   +------------------------------------------------------------------+
   | testdb                                                           |
   +------------------------------------------------------------------+
   SQLRowCount returns -1
   1 rows fetched
   ```

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`.

Note: To use a JDBC 4 driver, you use the JAR file based on the Java version being used:

- For Java 1.6, use the JDBC4 driver `postgresql-9.4-1208.jdbc4.jar`.
- For Java 1.7, use the JDBC41 driver `postgresql-9.4-1208.jdbc41.jar`.
- For Java 1.8, use the JDBC42 driver `postgresql-9.4-1208.jdbc42.jar`.

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 4.0-compliant driver and is using Java 1.6, use the following JAR file:

```
$GPHOME_CONNECTIVITY/drivers/jdbc/postgresql-9.4-1208.jdbc4.jar
```

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

```
GP_JDBC_JARFILE=postgresql-9.4-1208.jdbc4.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 includes variable settings 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 "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 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.EXAMPLE.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

Privacy Policy | *Terms of Use*

Copyright © 2018 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.