Amazon Redshift ODBC Driver

Installation and Configuration Guide

Amazon Web Services Inc.

Version 1.4.8
September 13, 2019
Copyright © 2019 Amazon Web Services Inc. All Rights Reserved.

Information in this document is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. No part of this publication, or the software it describes, may be reproduced, transmitted, transcribed, stored in a retrieval system, decompiled, disassembled, reverse-engineered, or translated into any language in any form by any means for any purpose without the express written permission of Amazon Web Services Inc.

Parts of this Program and Documentation include proprietary software and content that is copyrighted and licensed by Simba Technologies Incorporated. This proprietary software and content may include one or more feature, functionality or methodology within the ODBC, JDBC, ADO.NET, OLE DB, ODBO, XMLA, SQL and/or MDX component(s).

For information about Simba's products and services, visit: www.simba.com.

Contact Us

About This Guide

Purpose

The Amazon Redshift ODBC Driver Installation and Configuration Guide explains how to install and configure the Amazon Redshift ODBC Driver. The guide also provides details related to features of the driver.

Audience

The guide is intended for end users of the Amazon Redshift ODBC Driver, as well as administrators and developers integrating the driver.

Knowledge Prerequisites

To use the Amazon Redshift ODBC Driver, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Amazon Redshift ODBC Driver
- Ability to use the data source to which the Amazon Redshift ODBC Driver is connecting
- An understanding of the role of ODBC technologies and driver managers in connecting to a data source
- Experience creating and configuring ODBC connections
- Exposure to SQL

Document Conventions

*Italics* are used when referring to book and document titles.

**Bold** is used in procedures for graphical user interface elements that a user clicks and text that a user types.

*Monospace font* indicates commands, source code, or contents of text files.

📝 *Note:*

A text box with a pencil icon indicates a short note appended to a paragraph.
Important:

A text box with an exclamation mark indicates an important comment related to the preceding paragraph.
# Amazon Redshift ODBC Driver

## Table of Contents

About the Amazon Redshift ODBC Driver .......................................................... 7

Windows Driver .................................................................................................... 8
  Windows System Requirements ................................................................. 8
  Installing the Driver on Windows ............................................................ 8
  Creating a Data Source Name on Windows ............................................. 9
  Configuring SSL Verification on Windows ............................................... 10
  Configuring Authentication on Windows ................................................. 11
  Configuring Data Type Options on Windows .......................................... 18
  Configuring Additional Options on Windows ......................................... 19
  Configuring TCP Keepalives on Windows .............................................. 20
  Configuring Logging Options on Windows ............................................. 23
  Verifying the Driver Version Number on Windows ............................... 24

macOS Driver ................................................................................................. 25
  macOS System Requirements ................................................................. 25
  Installing the Driver on macOS ............................................................... 25
  Verifying the Driver Version Number on macOS ................................. 26

Linux Driver ................................................................................................... 27
  Linux System Requirements ................................................................. 27
  Installing the Driver Using the RPM File ............................................... 27
  Verifying the Driver Version Number on Linux .................................... 28

Configuring the ODBC Driver Manager on Non-Windows Machines ......... 29
  Specifying ODBC Driver Managers on Non-Windows Machines ........... 29
  Specifying the Locations of the Driver Configuration Files .................... 30

Configuring ODBC Connections on a Non-Windows Machine ................. 32
  Creating a Data Source Name on a Non-Windows Machine .................. 32
  Configuring a DSN-less Connection on a Non-Windows Machine .......... 36
  Configuring SSL Verification on a Non-Windows Machine ..................... 38
  Configuring Authentication on a Non-Windows Machine ...................... 39
  Configuring Query Processing Modes on a Non-Windows Machine ....... 45
  Configuring a Proxy Connection on a Non-Windows Machine ............... 46
  Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine .......... 47
  Configuring TCP Keepalives on a Non-Windows Machine .................... 48
Configuring Logging Options .................................................................49
Testing the Connection .........................................................................50

Using a Connection String .....................................................................53
DSN Connection String Example ..........................................................53
DSN-less Connection String Examples ..................................................53

Features ..................................................................................................60
Query Processing Modes .........................................................................60
TCP Keepalives .......................................................................................61
Data Types ............................................................................................61
Security and Authentication .....................................................................63

Driver Configuration Options ...............................................................65
Configuration Options Appearing in the User Interface .......................65
Configuration Options Having Only Key Names .....................................90

Contact Us ............................................................................................96

Third-Party Trademarks ..........................................................................97
About the Amazon Redshift ODBC Driver

The Amazon Redshift ODBC Driver enables Business Intelligence (BI), analytics, and reporting on data that is stored in Amazon Redshift. The driver complies with the ODBC 3.80 data standard and adds important functionality such as Unicode, as well as 32- and 64-bit support for high-performance computing environments on all platforms.

ODBC is one of the most established and widely supported APIs for connecting to and working with databases. At the heart of the technology is the ODBC driver, which connects an application to the database. For complete information about the ODBC specification, see the ODBC API Reference from the Microsoft documentation: https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/odbc-api-reference.

The Amazon Redshift ODBC Driver is available for Microsoft® Windows®, Linux, and macOS platforms.

The Installation and Configuration Guide is suitable for users who are looking to access Amazon Redshift data from their desktop environment. Application developers might also find the information helpful. Refer to your application for details on connecting via ODBC.
Windows System Requirements

Install the driver on client machines where the application is installed. Before installing the driver, make sure that you have the following:

- Administrator rights on your machine.
- A machine that meets the following system requirements:
  - One of the following operating systems:
    - Windows 10, 8.1, or 7 SP1
    - Windows Server 2016, 2012, or 2008 R2 SP1
  - 100 MB of available disk space
  - Visual C++ Redistributable for Visual Studio 2013 installed (with the same bitness as the driver that you are installing).


Installing the Driver on Windows

On 64-bit Windows operating systems, you can execute both 32- and 64-bit applications. However, 64-bit applications must use 64-bit drivers, and 32-bit applications must use 32-bit drivers. Make sure that you use a driver whose bitness matches the bitness of the client application:

- AmazonRedshiftODBC32-[Version].msi for 32-bit applications
- AmazonRedshiftODBC64-[Version].msi for 64-bit applications

You can install both versions of the driver on the same machine.

To install the Amazon Redshift ODBC Driver on Windows:

1. Depending on the bitness of your client application, double-click to run AmazonRedshiftODBC32-[Version].msi or AmazonRedshiftODBC64-[Version].msi.
2. Click Next.
3. Select the check box to accept the terms of the License Agreement if you agree, and then click Next.
4. To change the installation location, click Change, then browse to the desired folder, and then click OK. To accept the installation location, click Next.
5. Click **Install**.
6. When the installation completes, click **Finish**.

### Creating a Data Source Name on Windows

Typically, after installing the Amazon Redshift ODBC Driver, you need to create a Data Source Name (DSN).

Alternatively, for information about DSN-less connections, see **Using a Connection String** on page 53.

**To create a Data Source Name on Windows:**

1. From the Start menu, go to **ODBC Data Sources**.

   **Note:**
   
   Make sure to select the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to Redshift.

2. In the ODBC Data Source Administrator, click the **Drivers** tab, and then scroll down as needed to confirm that the Amazon Redshift ODBC Driver appears in the alphabetical list of ODBC drivers that are installed on your system.

3. Choose one:
   - To create a DSN that only the user currently logged into Windows can use, click the **User DSN** tab.
   - Or, to create a DSN that all users who log into Windows can use, click the **System DSN** tab.

   **Note:**
   
   It is recommended that you create a System DSN instead of a User DSN. Some applications load the data using a different user account, and might not be able to detect User DSNs that are created under another user account.

4. Click **Add**.
5. In the Create New Data Source dialog box, select **Amazon Redshift ODBC Driver** and then click **Finish**. The Amazon Redshift ODBC Driver DSN Setup dialog box opens.
6. In the **Data Source Name** field, type a name for your DSN.
7. In the **Server** field, type the endpoint of the server hosting the database that you want to access.
8. In the **Port** field, type the number of the TCP port that the server uses to listen for client connections.

**Note:**

If you are using IAM authentication and you specify the Cluster ID and AWS Region, you do not need to specify the server, and can leave this field blank.

9. In the **Database** field, type the name of the database that you want to access.

10. In the **Authentication** area, specify the configuration options to configure standard or IAM authentication. For more information, see Configuring Authentication on Windows on page 11.

11. To configure client-server verification over SSL, click **SSL Options**. For more information, see Configuring SSL Verification on Windows on page 10.

12. To configure advanced driver options, click **Additional Options**. For more information, see Configuring Additional Options on Windows on page 19.

13. To configure logging behavior for the driver, click **Logging Options**. For more information, see Configuring Logging Options on Windows on page 23.

14. To configure how the driver returns and displays data, click **Data Type Options**. For more information, see Configuring Data Type Options on Windows on page 18.

15. To test the connection, click **Test**. Review the results as needed, and then click **OK**.

16. To save your settings and close the Amazon Redshift ODBC Driver DSN Setup dialog box, click **OK**.

17. To close the ODBC Data Source Administrator, click **OK**.

### Configuring SSL Verification on Windows

If you are connecting to a Redshift server that has Secure Sockets Layer (SSL) enabled, then you can configure the driver to connect to an SSL-enabled socket. When connecting to a server over SSL, the driver supports identity verification between the client and the server.

**To configure SSL verification on Windows:**

1. To access the SSL options for a DSN, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **SSL Options**.

2. In the **Authentication Mode** list, select the appropriate SSL mode.
Note:

For information about SSL support in Amazon Redshift, see the topic Connect Using SSL in the Amazon Redshift Management Guide at http://docs.aws.amazon.com/redshift/latest/mgmt/connecting-ssl-support.html#connect-using-ssl.

3. To use the System Trust Store for SSL certificates, select the Use System Trust Store check box.

4. If you selected Use System Trust Store, choose one of the following options:
   - To check the validity of the certificate’s trust chain, select the Check Certificate Revocation check box.
   - Or, to accept self-signed certificates, select the Allow Self-signed Server Certificate check box.

5. To specify an SSL certificate, select the Enable Custom SSL CA Root Certificate check box, and then, in the Path field, specify the full path to the certificate file.

6. To specify the minimum version of SSL to use, from the Minimum TLS drop-down list, select the minimum version of SSL.

7. To save your settings and close the dialog box, click OK.

8. To save your settings and close the Amazon Redshift ODBC Driver DSN Setup dialog box, click OK.

Configuring Authentication on Windows

Redshift databases require authentication. You can configure the driver to provide your credentials and authenticate the connection to the database, or to use a profile or credentials service.

The driver supports the following authentication methods:

- Standard authentication using your database user name and password (see Using Standard Authentication on page 12)
- IAM authentication using a profile (see Using an IAM Profile on page 12)
- IAM authentication using IAM credentials (see Using IAM Credentials on page 14)
- IAM authentication using Active Directory Federation Services (AD FS) (see Using Active Directory Federation Services (AD FS) on page 14)
- IAM authentication using PingFederate service (see Using PingFederate Service on Windows on page 16)
- IAM authentication using Okta service (see Using Okta Service on page 16)
IAM authentication using a credentials service aside from those listed above (see Using an External Credentials Service on page 44)

For more information on IAM Roles and authentication, see http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html.

To configure authentication for your connection, follow the appropriate set of steps below.

**Using Standard Authentication**

You can configure the driver to authenticate your connection using your Redshift user name and password.

**To configure standard authentication on Windows:**

1. To access the authentication options, open the ODBC Data Source Administrator where you created the DSN, select the DSN, and then click Configure.
2. If Auth Type is not already set to Standard, then from the Auth Type drop-down list, select Standard.
3. In the User field, type your user name for accessing your Redshift account.
4. In the Password field, type the password corresponding to the user name you typed.
5. Encrypt your credentials by selecting one of the following:
   - If the credentials are used only by the current Windows user, select Current User Only.
   - Or, if the credentials are used by all users on the current Windows machine, select All Users Of This Machine.
6. To save your settings and close the dialog box, click OK.

**Using an IAM Profile**

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in a chained roles profile or the Amazon EC2 instance profile.
Note:

- The default location for the credentials file that contains chained roles profiles is ~/.aws/Credentials. The AWS_SHARED_CREDENTIALS_FILE environment variable can be used to point to a different credentials file.
- If any of the information requested in the following steps is already a part of the profile you intend to use, that field can be left blank. If the default profile is configured on your local machine, you only need to set the Auth Type to AWS Profile.

To configure IAM authentication using a profile on Windows:

1. To access the authentication options, open the ODBC Data Source Administrator where you created the DSN, select the DSN, and then click Configure.
2. From the Auth Type drop-down list select AWS Profile.
3. In the User field, type the user name for accessing your IDP Server.
4. In the Password field, type the password corresponding to the user name you typed.
5. Encrypt your credentials by selecting one of the following:
   - If the credentials are used only by the current Windows user, select Current User Only.
   - Or, if the credentials are used by all users on the current Windows machine, select All Users Of This Machine.

Note:

If the Server field is used, the Cluster ID and Region fields are optional.

6. In the Cluster ID field, type the ID for the Redshift server cluster.
7. In the Region field, type the region for the Redshift server cluster.
8. In the DbUser field, type the ID you want the Redshift user to use or have.
9. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   - Select the User AutoCreate check box.
   - In the DbGroups field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
10. Specify the profile that contains your credentials:
    - To use a chained roles profile, type the name of the profile in the Profile Name field, and leave the Use Instance Profile check box clear.
    - Or, to use the Amazon EC2 instance profile, select the Use Instance Profile check box.
11. To save your settings and close the dialog box, click **OK**.

### Using IAM Credentials

You can configure the driver to authenticate your connection through IAM authentication using IAM credentials.

#### To configure IAM authentication using IAM on Windows:

1. To access the authentication options, open the **ODBC Data Source Administrator** where you created the DSN, select the DSN, and then click **Configure**.
2. From the **Auth Type** drop-down list, select **AWS IAM Credentials**.

   - **Note:** If the **Server** field is used, the Cluster ID and Region fields are optional.

3. In the **Cluster ID** field, type the ID for the Redshift server cluster.
4. In the **Region** field, type the region for the Redshift server cluster.
5. In the **DbUser** field, type the ID you want the Redshift user to use or have.
6. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   - Select the **User AutoCreate** check box.
   - In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
7. In the **AccessKeyId** field, type your Redshift access key ID.
8. In the **SecretAccessKey** field, type your Redshift secret key.
9. If you are using an IAM role, in the **SessionToken** field, type your temporary session token.
10. To save your settings and close the dialog box, click **OK**.

### Using Active Directory Federation Services (AD FS)

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in AD FS.
To configure IAM authentication using AD FS on Windows:

1. To access the IAM authentication options, open the **ODBC Data Source Administrator** where you created the DSN, select the DSN, and then click Configure.
2. From the **Auth Type** drop-down list, select **Identity Provider: AD FS**.
3. Choose one of the following options:
   - To log in using Windows Integrated Authentication, leave the **User** and **Password** fields blank.
   - Or, to log in without using integrated authentication:
     a. In the **User** field, type the user name associated with your AD FS account.
     b. In the **Password** field, type the password associated with your AD FS user name.
4. Encrypt your credentials by selecting one of the following:
   - If the credentials are used only by the current Windows user, select **Current User Only**.
   - Or, if the credentials are used by all users on the current Windows machine, select **All Users Of This Machine**.

**Note:**
If the Server field is used, the Cluster ID and Region fields are optional.

5. In the **Cluster ID** field, type the ID for the Redshift server cluster.
6. In the **Region** field, type the region for the Redshift server cluster.
7. In the **DbUser** field, type the ID you want the Redshift user to use or have.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   - Select the **User AutoCreate** check box.
   - In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
9. In the **IdP Host** field, type the address of the service host.
10. In the **IdP Port** field, type the port number the service listens at.
11. To skip verification of the SSL certificate of the IDP server, select the **SSL Insecure** check box.
12. In the **Preferred Role** field, type the name or ID for the IAM role you want the user to assume when logged in to Redshift.
13. To save your settings and close the dialog box, click **OK**.
Using PingFederate Service on Windows

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in the PingFederate service.

To configure IAM authentication using PingFederate service on Windows:

1. To access the IAM authentication options, open the ODBC Data Source Administrator where you created the DSN, select the DSN, and then click Configure.
2. In the Authentication area, click the Auth Type drop down and select Identity Provider: PingFederate.
3. In the User field, type the user name associated with your Ping account.
4. In the Password field, type the password associated with your Ping user name.

> Note:

If the Server field is used, the Cluster ID and Region fields are optional.

5. In the Cluster ID field, type the ID for the Redshift server cluster.
6. In the Region field, type the region for the Redshift server cluster.
7. In the DbUser field, type the ID you want the Redshift user to use or have.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   - Select the User AutoCreate check box.
   - In the DbGroups field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
9. In the IdP Host field, type the address of the service host.
10. In the IdP Port field, type the port number the service listens at.
11. To skip verification of the SSL certificate of the IDP server, select the SSL Insecure check box.
12. In the Preferred Role field, type the name or ID for the IAM role you want the user to assume when logged in to Redshift.
13. Optionally, in the Partner SPID field, type a partner SPID (service provider ID) value.
14. To save your settings and close the dialog box, click OK.

Using Okta Service

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in Okta.
To configure IAM authentication using Okta on Windows:

1. To access the IAM authentication options, open the ODBC Data Source Administrator where you created the DSN, select the DSN, and then click Configure.
2. In the Authentication area, click the Auth Type drop down and select Identity Provider: Okta.
3. In the User field, type the user name associated with your Okta account.
4. In the Password field, type the password associated with your Okta user name. If you are using a profile, this may be optional.
5. Encrypt your credentials by selecting one of the following:
   - If the credentials are used only by the current Windows user, select Current User Only.
   - Or, if the credentials are used by all users on the current Windows machine, select All Users Of This Machine.

   **Note:**
   If the Server field is used, the Cluster ID and Region fields are optional.

6. In the Cluster ID field, type the ID for the Redshift server cluster.
7. In the Region field, type the region for the Redshift server cluster.
8. In the DbUser field, type the ID you want the Redshift user to use or have.
9. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   - Select the User AutoCreate check box.
   - In the DbGroups field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
10. In the IdP Host field, type the address of the service host.
11. In the Preferred Role field, type the name or ID for the IAM role you want the user to assume when logged in to Redshift.
12. In the Okta App ID field, type the Okta-supplied ID associated with your Redshift application.
13. Optionally, in the Okta App Name field, type the name of your Okta application.
14. To save your settings and close the dialog box, click OK.

**Using an External Credentials Service**

In addition to built-in support for AD FS, PingFederate, and Okta, the Windows version of the Amazon Redshift ODBC Driver also provides support for other credentials services. The driver can authenticate connections using any SAML-based credential provider plugin of your choice.
To configure an external credentials service on Windows:

1. Create an IAM profile that specifies the credential provider plugin and other authentication parameters as needed. The profile must be ASCII-encoded, and must contain the following key-value pair, where [PluginPath] is the full path to the plugin application:

   ```plaintext
   plugin_name = [PluginPath]
   ```

   For example:

   ```plaintext
   plugin_name = C:\Users\jsmith\ApplicationInstallDir\CredServiceApplication.exe
   ```

   For information about how to create a profile, see "Using a Configuration Profile" in the Amazon Redshift Cluster Management Guide: https://docs.aws.amazon.com/redshift/latest/mgmt/options-for-providing-iam-credentials.html#using-configuration-profile.

2. Configure the driver to use this profile. For detailed instructions, see Using an IAM Profile on page 12.

The driver detects and uses the authentication settings specified in the profile.

**Configuring Data Type Options on Windows**

You can configure data type options to modify how the driver displays or returns some data types.

To configure data type options on Windows:

1. To access data type options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Data Type Options**.

2. To enable the driver to return data as Unicode character types, select the **Use Unicode** check box.

**Note:**

When the **Use Unicode** check box is selected, the driver does the following:

- Returns SQL_WCHAR instead of SQL_CHAR.
- Returns SQL_WVARCHAR instead of SQL VARCHAR.
- Returns SQL_WLONGVARCHAR instead of SQL LONGVARCHAR.
3. To configure the driver to return Boolean columns as SQL VARCHAR instead of SQL_BIT, select the **Show Boolean Column As String** check box.

4. To configure the driver to return Text columns as SQL_LONGVARCHAR instead of SQL_VARCHAR, select the **Text as LongVarChar** check box.

5. To configure the driver to return Bytea columns as SQL_LONGVARBINARY instead of SQL_VARBINARY, select the **Bytea As LongVarBinary** check box.

6. In the **Max Varchar** field, type the maximum data length for VarChar columns.

7. In the **Max LongVarChar** field, type the maximum data length for LongVarChar columns.

8. In the **Max Bytea** field, type the maximum data length for Bytea columns.

9. To save your settings and close the Data Type Configuration dialog box, click **OK**.

**Configuring Additional Options on Windows**

You can configure additional options to modify the behavior of the driver.

**To configure additional options on Windows:**

1. To access advanced options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Additional Options**.

2. Specify how the driver processes queries by doing one of the following:
   - To return query results one row at a time, select **Single Row Mode**.
   - To return a specific number of rows at a time, select **Use Declare/Fetch** and then, in the **Cache Size** field, type the number of rows.
   - To enable the driver to have multiple queries active on the same connection, select **Use Multiple Statements**. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially.
   - To return the entire query result, select **Retrieve Entire Result Into Memory**.

   **Note:**

   Use **Single Row Mode** if you plan to query large results and you do not want to retrieve the entire result into memory. Disabling **Single Row Mode** increases performance, but can result in out-of-memory errors.

3. To configure the driver to have only one active query at a time per connection, select the **Enforce Single Statement** check box.
4. To configure the driver to recognize table type information from the data source, select the **Enable Table Types** check box. For more information, see **Enable Table Types** on page 72.

5. To connect to Redshift through a proxy server, select the **Enable Proxy For Amazon Redshift Connection** check box and then do the following:
   a. In the **Proxy Server** field, type the host name or IP address of the proxy server.
   b. In the **Proxy Port** field, type the number of the TCP port that the proxy server uses to listen for client connections.
   c. If the proxy server requires authentication, then do the following:
      i. In the **Proxy Username** field, type your user name for accessing the proxy server.
      ii. In the **Proxy Password** field, type the password corresponding to the user name.

6. To configure the driver to pass IAM authentication processes through a proxy server, select the **Enable HTTPS Proxy For Federated Access** check box and then do the following:
   a. In the **HTTPS Proxy Server** field, type the host name or IP address of the proxy server.
   b. In the **HTTPS Proxy Port** field, type the number of the port that the proxy server uses to listen for client connections.
   c. If the proxy server requires authentication, then do the following:
      i. In the **HTTPS Proxy Username** field, type your user name for accessing the proxy server.
      ii. In the **HTTPS Proxy Password** field, type the password corresponding to the user name.
   d. To pass the authentication processes for identity providers through the proxy server, select the **Use HTTPS Proxy For Authentication On IdP** check box.

7. To save your settings and close the Additional Configuration dialog box, click **OK**.

8. To save your settings and close the Amazon Redshift ODBC Driver DSN Setup dialog box, click **OK**.

**Configuring TCP Keepalives on Windows**

By default, the Amazon Redshift ODBC Driver is configured to use TCP keepalives to prevent connections from timing out. Settings such as how frequently the driver sends TCP keepalive packets are based on the operating system defaults. You can configure the TCP keepalive settings or disable the feature by modifying the appropriate values in the Windows Registry.
Important:
Editing the Windows Registry incorrectly can potentially cause serious, system-wide problems that may require re-installing Windows to correct.

To configure TCP keepalives on Windows:

1. Choose one:
   - If you are using Windows 7 or earlier, click Start, then type regedit in the Search field, and then click regedit.exe in the search results.
   - Or, if you are using Windows 8 or later, on the Start screen, type regedit, and then click the regedit search result.

2. Select the appropriate registry key for the bitness of your driver:
   - If you are using the 32-bit driver on a 64-bit machine, then select the following registry key, where [YourDSN] is the DSN for which you want to configure keepalives:
     
     HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\[YourDSN]
   - Otherwise, select the following registry key, where [YourDSN] is the DSN for which you want to configure keepalives:
     
     HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\[YourDSN]

3. To specify the interval of inactivity before the driver sends a TCP keepalive packet, configure the KeepAliveIdle value by doing the following:
   a. If the KeepAliveIdle value does not already exist, create it. Select Edit > New > String Value, type KeepAliveIdle as the name of the value, and then press Enter.
   b. Select the KeepAliveIdle value, and then Select Edit > Modify.
   c. In the Edit String dialog box, in the Value Data field, type the number of seconds of inactivity before the driver sends a TCP keepalive packet.
   d. Click OK.

4. To specify the number of TCP keepalive packets that can be lost before the connection is considered broken, configure the KeepAliveCount value. To do this, follow the procedure above, but type KeepAliveCount for the value name, and in the Value Data field, type the number of keepalive packets that can be lost.
Note:
To use the system default, in the Value Data field, type 0.

5. To specify the interval of time between each retransmission of a keepalive packet, configure the KeepAliveInterval value. To do this, follow the procedure above, but type `KeepAliveInterval` for the value name, and in the Value Data field, type the number of seconds to wait between each retransmission.

Note:
To use the system default, in the Value Data field, type 0.

6. Close the Registry Editor.

To disable TCP keepalives:

1. Choose one:
   - If you are using Windows 7 or earlier, click Start, then type `regedit` in the Search field, and then click `regedit.exe` in the search results.
   - Or, if you are using Windows 8 or later, on the Start screen, type `regedit`, and then click the `regedit` search result.

2. Select the appropriate registry key for the bitness of your driver:
   - If you are using the 32-bit driver on a 64-bit machine, then select the following registry key, where `[YourDSN]` is the DSN for which you want to configure keepalives:
     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ODBC\ODBC.INI\[YourDSN]
     ```
   - Otherwise, select the following registry key, where `[YourDSN]` is the DSN for which you want to configure keepalives:
     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\[YourDSN]
     ```

3. If the `KeepAlive` value does not already exist, create it. Select Edit > New > String Value, then type `KeepAlive` as the name of the value, and then press Enter.

4. Select the `KeepAlive` value, and then click Edit > Modify.

5. In the Edit String dialog box, in the Value Data field, type 0.

6. Click OK.

7. Close the Registry Editor.

Note:
To enable TCP keepalives after disabling them, set `KeepAlive` to 1.
Configuring Logging Options on Windows

To help troubleshoot issues, you can enable logging. In addition to functionality provided in the Amazon Redshift ODBC Driver, the ODBC Data Source Administrator provides tracing functionality.

**Important:**

Only enable logging or tracing long enough to capture an issue. Logging or tracing decreases performance and can consume a large quantity of disk space.

The settings for logging apply to every connection that uses the Amazon Redshift ODBC Driver, so make sure to disable the feature after you are done using it.

To enable driver logging on Windows:

1. To access logging options, open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click **Configure**, and then click **Logging Options**.
2. From the **Log Level** drop-down list, select the logging level corresponding to the amount of information that you want to include in log files:

<table>
<thead>
<tr>
<th>Logging Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>Disables all logging.</td>
</tr>
<tr>
<td>FATAL</td>
<td>Logs severe error events that lead the driver to abort.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Logs error events that might allow the driver to continue running.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Logs events that might result in an error if action is not taken.</td>
</tr>
<tr>
<td>INFO</td>
<td>Logs general information that describes the progress of the driver.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Logs detailed information that is useful for debugging the driver.</td>
</tr>
<tr>
<td>TRACE</td>
<td>Logs all driver activity.</td>
</tr>
</tbody>
</table>

3. In the **Log Path** field, specify the full path to the folder where you want to save log files.
4. Click OK.
5. Restart your ODBC application to make sure that the new settings take effect.

The Amazon Redshift ODBC Driver produces the following log files at the location you specify in the Log Path field:

- A `amazonredshiftodbcdriver.log` file that logs driver activity that is not specific to a connection.
- A `amazonredshiftodbcdriver_connection_{Number}.log` file for each connection made to the database, where `{Number}` is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If you enable the `UseLogPrefix` connection property, the driver prefixes the log file name with the user name associated with the connection and the process ID of the application through which the connection is made. For more information, see `UseLogPrefix` on page 94.

**To disable driver logging on Windows:**

1. Open the ODBC Data Source Administrator where you created the DSN, then select the DSN, then click `Configure`, and then click `Logging Options`.
2. From the `Log Level` drop-down list, select `LOG_OFF`.
3. Click OK.
4. Restart your ODBC application to make sure that the new settings take effect.

**Verifying the Driver Version Number on Windows**

If you need to verify the version of the Amazon Redshift ODBC Driver that is installed on your Windows machine, you can find the version number in the ODBC Data Source Administrator.

**To verify the driver version number on Windows:**

1. From the Start menu, go to **ODBC Data Sources**.

   **Note:**
   Make sure to select the ODBC Data Source Administrator that has the same bitness as the client application that you are using to connect to Redshift.

2. Click the **Drivers** tab and then find the Amazon Redshift ODBC Driver in the list of ODBC drivers that are installed on your system. The version number is displayed in the **Version** column.
macOS Driver

macOS System Requirements

Install the driver on client machines where the application is installed. Each client machine that you install the driver on must meet the following minimum system requirements:

- macOS version 10.12, 10.13, or 10.14
- 215 MB of available disk space
- iODBC 3.52.9, 3.52.10, 3.52.11, or 3.52.12

Installing the Driver on macOS

The Amazon Redshift ODBC Driver is available for macOS as a .dmg file named AmazonRedshiftODBC.dmg. The driver supports both 32- and 64-bit client applications.

To install the Amazon Redshift ODBC Driver on macOS:

1. Double-click AmazonRedshiftODBC.dmg to mount the disk image.
2. Double-click AmazonRedshiftODBC.pkg to run the installer.
3. In the installer, click Continue.
4. On the Software License Agreement screen, click Continue, and when the prompt appears, click Agree if you agree to the terms of the License Agreement.
5. Optionally, to change the installation location, click Change Install Location, then select the desired location, and then click Continue.

⚠️ Note:
By default, the driver files are installed in the /opt/amazon/redshiftodbc directory.

6. To accept the installation location and begin the installation, click Install.
7. When the installation completes, click Close.

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 29.
Verifying the Driver Version Number on macOS

If you need to verify the version of the Amazon Redshift ODBC Driver that is installed on your macOS machine, you can query the version number through the Terminal.

**To verify the driver version number on macOS:**

- At the Terminal, run the following command:

```
pkgutil --info com.amazon.redshiftodbc
```

The command returns information about the Amazon Redshift ODBC Driver that is installed on your machine, including the version number.
Linux Driver

Linux System Requirements

Install the driver on client machines where the application is installed. Each client machine that you install the driver on must meet the following minimum system requirements:

- One of the following distributions:
  - Red Hat® Enterprise Linux® (RHEL) 6 or 7
  - CentOS 6 or 7
  - SUSE Linux Enterprise Server (SLES) 11 or 12
  - Debian 7, 8, or 9
  - Ubuntu 14.04 or 16.04
  - Oracle Linux 7.5
- 150 MB of available disk space
- One of the following ODBC driver managers installed:
  - iODBC 3.52.9, 3.52.10, 3.52.11, or 3.52.12
  - unixODBC 2.3.2, 2.3.3, or 2.3.4

To install the driver, you must have root access on the machine.

Installing the Driver Using the RPM File

On 64-bit editions of Linux, you can execute both 32- and 64-bit applications. However, 64-bit applications must use 64-bit drivers, and 32-bit applications must use 32-bit drivers. Make sure that you use a driver whose bitness matches the bitness of the client application:

- `AmazonRedshiftODBC-32-bit-[Version]-[Release].i686.rpm` for the 32-bit driver
- `AmazonRedshiftODBC-64-bit-[Version]-[Release].x86_64.rpm` for the 64-bit driver

The placeholders in the file names are defined as follows:

- `[Version]` is the version number of the driver.
- `[Release]` is the release number for this version of the driver.

You can install both the 32-bit and 64-bit versions of the driver on the same machine.
To install the Amazon Redshift ODBC Driver using the RPM File:

1. Log in as the root user.
2. Navigate to the folder containing the RPM package for the driver.
3. Depending on the Linux distribution that you are using, run one of the following commands from the command line, where `[RPMFileName]` is the file name of the RPM package:
   - If you are using Red Hat Enterprise Linux or CentOS, run the following command:
     ```bash
     yum --nogpgcheck localinstall [RPMFileName]
     ```
   - Or, if you are using SUSE Linux Enterprise Server, run the following command:
     ```bash
     zypper install [RPMFileName]
     ```

The Amazon Redshift ODBC Driver files are installed in the `/opt/amazon/redshiftodbc` directory.

Next, configure the environment variables on your machine to make sure that the ODBC driver manager can work with the driver. For more information, see Configuring the ODBC Driver Manager on Non-Windows Machines on page 29.

Verifying the Driver Version Number on Linux

If you need to verify the version of the Amazon Redshift ODBC Driver that is installed on your Linux machine, you can query the version number through the command-line interface if the driver was installed using an RPM file.

To verify the driver version number on Linux using the command-line interface:

- Depending on your package manager, at the command prompt, run one of the following commands:
  - ```bash
    yum list | grep AmazonRedshiftODBC
    ```
  - ```bash
    rpm -qa | grep AmazonRedshiftODBC
    ```

The command returns information about the Amazon Redshift ODBC Driver that is installed on your machine, including the version number.
Configuring the ODBC Driver Manager on Non-Windows Machines

To make sure that the ODBC driver manager on your machine is configured to work with the Amazon Redshift ODBC Driver, do the following:

- Set the library path environment variable to make sure that your machine uses the correct ODBC driver manager. For more information, see Specifying ODBC Driver Managers on Non-Windows Machines on page 29.
- If the driver configuration files are not stored in the default locations expected by the ODBC driver manager, then set environment variables to make sure that the driver manager locates and uses those files. For more information, see Specifying the Locations of the Driver Configuration Files on page 30.

After configuring the ODBC driver manager, you can configure a connection and access your data store through the driver.

Specifying ODBC Driver Managers on Non-Windows Machines

You need to make sure that your machine uses the correct ODBC driver manager to load the driver. To do this, set the library path environment variable.

macOS

If you are using a macOS machine, then set the DYLD_LIBRARY_PATH environment variable to include the paths to the ODBC driver manager libraries. For example, if the libraries are installed in /usr/local/lib, then run the following command to set DYLD_LIBRARY_PATH for the current user session:

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
```

For information about setting an environment variable permanently, refer to the macOS shell documentation.

Linux

If you are using a Linux machine, then set the LD_LIBRARY_PATH environment variable to include the paths to the ODBC driver manager libraries. For example, if the libraries are installed in /usr/local/lib, then run the following command to set LD_LIBRARY_PATH for the current user session:
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib

For information about setting an environment variable permanently, refer to the Linux shell documentation.

Specify the Locations of the Driver Configuration Files

By default, ODBC driver managers are configured to use hidden versions of the odbc.ini and odbcinst.ini configuration files (named .odbc.ini and .odbcinst.ini) located in the home directory, as well as the amazon.redshiftodbc.ini file in the lib subfolder of the driver installation directory. If you store these configuration files elsewhere, then you must set the environment variables described below so that the driver manager can locate the files.

If you are using iODBC, do the following:

- Set ODBCINI to the full path and file name of the odbc.ini file.
- Set ODBCINSTINI to the full path and file name of the odbcinst.ini file.
- Set AMAZONREDSHIFTODBCINI to the full path and file name of the amazon.redshiftodbc.ini file.

⚠️ Note:
If you acquired the driver from a vendor other than Amazon, you need to replace AMAZON with the name of your vendor.

If you are using unixODBC, do the following:

- Set ODBCINI to the full path and file name of the odbc.ini file.
- Set ODBCSYSINI to the full path of the directory that contains the odbcinst.ini file.
- Set AMAZONREDSHIFTODBCINI to the full path and file name of the amazon.redshiftodbc.ini file.

⚠️ Note:
If you acquired the driver from a vendor other than Amazon, you need to replace AMAZON with the name of your vendor.

For example, if your odbc.ini and odbcinst.ini files are located in /usr/local/odbc and your amazon.redshiftodbc.ini file is located in /etc, then set the environment variables as follows:
For iODBC:

```bash
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCINSTINI=/usr/local/odbc/odbcinst.ini
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
```

For unixODBC:

```bash
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBC_SYSINI=/usr/local/odbc
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
```

To locate the `amazon.redshiftodbc.ini` file, the driver uses the following search order:

1. If the `AMAZONREDSHIFTODBCINI` environment variable is defined, then the driver searches for the file specified by the environment variable.
2. The driver searches the directory that contains the driver library files for a file named `amazon.redshiftodbc.ini`.
3. The driver searches the current working directory of the application for a file named `amazon.redshiftodbc.ini`.
4. The driver searches the home directory for a hidden file named `.amazon.redshiftodbc.ini` (prefixed with a period).
5. The driver searches the `/etc` directory for a file named `amazon.redshiftodbc.ini`. 
The following sections describe how to configure ODBC connections when using the Amazon Redshift ODBC Driver on non-Windows platforms:

- Creating a Data Source Name on a Non-Windows Machine on page 32
- Configuring a DSN-less Connection on a Non-Windows Machine on page 36
- Configuring Authentication on a Non-Windows Machine on page 39
- Configuring SSL Verification on a Non-Windows Machine on page 38
- Configuring Query Processing Modes on a Non-Windows Machine on page 45
- Configuring a Proxy Connection on a Non-Windows Machine on page 46
- Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine on page 47
- Configuring TCP Keepalives on a Non-Windows Machine on page 48
- Configuring Logging Options on page 49
- Testing the Connection on page 50

Creating a Data Source Name on a Non-Windows Machine

When connecting to your data store using a DSN, you only need to configure the odbc.ini file. Set the properties in the odbc.ini file to create a DSN that specifies the connection information for your data store. For information about configuring a DSN-less connection instead, see Configuring a DSN-less Connection on a Non-Windows Machine on page 36.

If your machine is already configured to use an existing odbc.ini file, then update that file by adding the settings described below. Otherwise, copy the odbc.ini file from the Setup subfolder in the driver installation directory to the home directory, and then update the file as described below.
To create a Data Source Name on a non-Windows machine:

1. In a text editor, open the `odbc.ini` configuration file.

   **Note:**
   If you are using a hidden copy of the `odbc.ini` file, you can remove the period (.) from the start of the file name to make the file visible while you are editing it.

2. In the `[ODBC Data Sources]` section, add a new entry by typing a name for the DSN, an equal sign (=), and then the name of the driver.

   For example, on a macOS machine:

   ```
   [ODBC Data Sources]
   Sample DSN=Amazon Redshift ODBC Driver
   ```

   As another example, for a 32-bit driver on a Linux machine:

   ```
   [ODBC Data Sources]
   Sample DSN=Amazon Redshift ODBC Driver 32-bit
   ```

3. Create a section that has the same name as your DSN, and then specify configuration options as key-value pairs in the section:

   a. Set the `Driver` property to the full path of the driver library file that matches the bitness of the application.

      For example, on a macOS machine:

      ```
      Driver=/opt/amazon/redshiftodbc/lib/universal/libamazonredshiftodbc.dylib
      ```

      As another example, for a 32-bit driver on a Linux machine:

      ```
      Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
      ```

   b. Set the `Server` property to a comma-delimited list of endpoint servers you want to connect to, and then set the `Port` property to the number of the TCP port that these servers use to listen for client connections.

      For example:

      ```
      Server=testserver.abcabcabcabc.com,testserver.cbacbabcba.com,
      Port=5439
      ```
c. Set the `Database` property to the name of the database that you want to access.

For example:

```
Database=TestDB
```

d. To configure authentication, specify the authentication mechanism and your credentials. For more information, see Configuring Authentication on a Non-Windows Machine on page 39.

e. To connect to the server through SSL, enable SSL and specify the certificate information. For more information, see Configuring SSL Verification on a Non-Windows Machine on page 38.

f. Optionally, modify how the driver runs queries and retrieves results into memory. For more information, see Configuring Query Processing Modes on a Non-Windows Machine on page 45.

g. Optionally, configure the driver to connect through a proxy server. For more information, see Configuring a Proxy Connection on a Non-Windows Machine on page 46.

h. Optionally, configure the driver to pass IAM authentication processes through a proxy server. For more information, see Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine on page 47.

i. Optionally, modify the TCP keepalive settings that the driver uses to prevent connections from timing out. For more information, see Configuring TCP Keepalives on a Non-Windows Machine on page 48.

j. Optionally, set additional key-value pairs as needed to specify other optional connection settings. For detailed information about all the configuration options supported by the Amazon Redshift ODBC Driver, see Driver Configuration Options on page 65.

4. Save the `odbc.ini` configuration file.
For example, the following is an odbc.ini configuration file for macOS containing a DSN that connects to Redshift:

```
[ODBC Data Sources]
Sample DSN=Amazon Redshift ODBC Driver
[Sample DSN]
Driver=/Library/amazon/redshiftodbc/lib/libredshiftodbc_sbu.dylib
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com
Port=5432
Database=TestDB
UID=amazon
PWD=amazon123
```

As another example, the following is an odbc.ini configuration file for a 32-bit driver on a Linux machine, containing a DSN that connects to Redshift:

```
[ODBC Data Sources]
Sample DSN=Amazon Redshift ODBC Driver 32-bit
[Sample DSN]
Driver=/opt/amazon/redshiftodbc/lib/32/libredshiftodbc_sbd32.so
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com
Port=5432
Database=TestDB
UID=amazon
PWD=amazon123
```

You can now use the DSN in an application to connect to the data store.
Configuring a DSN-less Connection on a Non-Windows Machine

To connect to your data store through a DSN-less connection, you need to define the driver in the odbcinst.ini file and then provide a DSN-less connection string in your application.

If your machine is already configured to use an existing odbcinst.ini file, then update that file by adding the settings described below. Otherwise, copy the odbcinst.ini file from the Setup subfolder in the driver installation directory to the home directory, and then update the file as described below.

To define a driver on a non-Windows machine:

1. In a text editor, open the odbcinst.ini configuration file.

   ![Note:]
   If you are using a hidden copy of the odbcinst.ini file, you can remove the period (.) from the start of the file name to make the file visible while you are editing it.

2. In the [ODBC Drivers] section, add a new entry by typing a name for the driver, an equal sign (=), and then Installed.

   For example:

   [ODBC Drivers]
   Amazon Redshift ODBC Driver=Installed

3. Create a section that has the same name as the driver (as specified in the previous step), and then specify the following configuration options as key-value pairs in the section:

   a. Set the Driver property to the full path of the driver library file that matches the bitness of the application.

      For example, on a macOS machine:

      ```
      Driver=/opt/amazon/redshiftodbc/lib/universal/libamazonredshiftodbc.dylib
      ```

      As another example, for a 32-bit driver on a Linux machine:

      ```
      Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
      ```
b. Optionally, set the **Description** property to a description of the driver.

For example:

```
Description=Amazon Redshift ODBC Driver
```

4. Save the `odbcinst.ini` configuration file.

**Note:**

If you are storing this file in its default location in the home directory, then prefix the file name with a period (.) so that the file becomes hidden. If you are storing this file in another location, then save it as a non-hidden file (without the prefix), and make sure that the `ODBCINSTINI` or `ODBCSYSINI` environment variable specifies the location. For more information, see **Specifying the Locations of the Driver Configuration Files** on page 30.

For example, the following is an `odbcinst.ini` configuration file for macOS:

```
[ODBC Drivers]
AmazonRedshift ODBC Driver=Installed
[AmazonRedshift ODBC Driver]
Description=Amazon Redshift ODBC Driver
Driver=/opt/amazon/redshiftodbc/lib/universal/libamazonredshiftodbc.dylib
```

As another example, the following is an `odbcinst.ini` configuration file for both the 32- and 64-bit drivers on Linux:

```
[ODBC Drivers]
Amazon Redshift ODBC Driver 32-bit=Installed
AmazonRedshift ODBC Driver 64-bit=Installed
[Amazon Redshift ODBC Driver 32-bit]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
[Amazon Redshift ODBC Driver 64-bit]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

You can now connect to your data store by providing your application with a connection string where the **Driver** property is set to the driver name specified in the `odbcinst.ini` file, and all the other necessary connection properties are also set.
For more information, see "DSN-less Connection String Examples" in **Using a Connection String** on page 53.

For instructions about configuring specific connection features, see the following:

- Configuring Authentication on a Non-Windows Machine on page 39
- Configuring SSL Verification on a Non-Windows Machine on page 38
- Configuring a Proxy Connection on a Non-Windows Machine on page 46
- Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine on page 47
- Configuring Query Processing Modes on a Non-Windows Machine on page 45
- Configuring TCP Keepalives on a Non-Windows Machine on page 48

For detailed information about all the connection properties that the driver supports, see **Driver Configuration Options** on page 65.

### Configuring SSL Verification on a Non-Windows Machine

If you are connecting to a Redshift server that has Secure Sockets Layer (SSL) enabled, then you can configure the driver to connect to an SSL-enabled socket. When connecting to a server over SSL, the driver supports identity verification between the client and the server.

You can set the connection properties described below in a connection string or in a DSN (in the odbc.ini file). Settings in the connection string take precedence over settings in the DSN.

**To configure SSL verification on a non-Windows machine:**

1. Set the **SSLMode** property to the appropriate SSL mode.

   ✍️ Note:
   
   For information about SSL support in Amazon Redshift, see the topic *Connect Using SSL* in the Amazon Redshift Management Guide at [http://docs.aws.amazon.com/redshift/latest/mgmt/connecting-ssl-support.html#connect-using-ssl](http://docs.aws.amazon.com/redshift/latest/mgmt/connecting-ssl-support.html#connect-using-ssl).

2. To specify an SSL certificate, set the **SSLCertPath** property to the full path and file name of the certificate file.

3. To specify the minimum version of SSL to use, set the **Min_TLS** property to the minimum version of SSL. Supported options include 1.0 for TLS 1.0, 1.1 for TLS 1.1, and 1.2 for TLS 1.2.
Configuring Authentication on a Non-Windows Machine

Redshift databases require authentication. You can configure the driver to provide your credentials and authenticate the connection to the database, or to use a profile or credentials service.

You can set the connection properties described below in a connection string or in a DSN (in the odbc.ini file). Settings in the connection string take precedence over settings in the DSN.

The driver supports the following authentication methods:

- Standard authentication using your database user name and password (see Using Standard Authentication on page 39)
- IAM authentication using a profile (see Using an IAM Profile on page 40)
- IAM authentication using IAM credentials (see Using IAM Credentials on page 41)
- IAM authentication using Active Directory Federation Services (AD FS) (see Using Active Directory Federation Services (AD FS) on page 41)
- IAM authentication using PingFederate service (see Using PingFederate Service on page 42)
- IAM authentication using Okta service (see Using Okta Service on page 43)

For more information on IAM Roles and authentication, see http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html.

To configure authentication for your connection, follow the appropriate set of steps below.

Using Standard Authentication

You can configure the driver to authenticate your connection using your Redshift user name and password.

To configure standard authentication on a non-Windows machine:

1. Set the UID property to an appropriate user name for accessing the Redshift server.
2. Set the PWD property to the password corresponding to the user name you provided above.
Using an IAM Profile

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in a chained roles profile or the Amazon EC2 instance profile.

**Note:**

- The default location for the credentials file that contains chained roles profiles is `~/.aws/Credentials`. The AWS_SHARED_CREDENTIALS_FILE environment variable can be used to point to a different credentials file.
- If any of the information requested in the following steps is already a part of the profile you intend to use, that property can be omitted. If the default profile is configured on your local machine, you do not need to set any of these properties.

**To configure IAM authentication using a profile on a non-Windows machine:**

1. Set the **UID** property to an appropriate user name for accessing the Redshift server.
2. Set the **PWD** property to the password corresponding to the user name you provided above.
3. Set the **IAM** property to 1.
4. Set the **ClusterID** property to the ID for the Redshift server cluster.
5. Set the **Region** property to the region for the Redshift server cluster.

**Note:**

If the **Server** property is set, the **ClusterID** and **Region** properties are optional.

6. Set the **DbUser** property to the ID you want the Redshift user to use or have.
7. If the ID you specified for the **DbUser** property does not already exist in your Redshift account, you must create it:
   - Set the **AutoCreate** property to 1.
   - Set the **DbGroups** property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
8. Specify the profile that contains your credentials:
   - To use a chained roles profile, set the **Profile** property to the name of the profile, and then either set the **InstanceProfile** property to 0 or make sure that it is not set at all.
   - Or, to use the Amazon EC2 instance profile, set the **InstanceProfile** property to 1.
Using IAM Credentials

You can configure the driver to authenticate your connection through IAM authentication using IAM credentials.

To configure IAM authentication using IAM on a non-Windows machine:

1. Set the IAM property to 1.
2. Set the ClusterID property to the ID for the Redshift server cluster.
3. Set the Region property to the region for the Redshift server cluster.

4. Set the DbUser property to the ID you want the Redshift user to use or have.
5. If the ID you specified for the DbUser property does not already exist in your Redshift account, you must create it:
   - Set the AutoCreate property to 1.
   - Set the DbGroups property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
6. Set the AccessKeyId property to your Redshift access key ID.
7. Set the SecretAccessKey property to your Redshift secret key.
8. If you are using an IAM role, set the SessionToken property to your temporary session token.

Using Active Directory Federation Services (AD FS)

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in AD FS.

To configure IAM authentication using AD FS on a non-Windows machine:

1. Choose one of the following options:
   - To log in using Windows Integrated Authentication, do not specify the UID and PWD properties.
• Or, to log in without using integrated authentication:
  a. Set the UID property to the user name associated with your AD FS account.
  b. Set the PWD property to the password associated with your AD FS user name.
 2. Set the IAM property to 1.
 3. Set the plugin_name property to adfs.
 4. Set the ClusterID property to the ID for the Redshift server cluster.
 5. Set the Region property to the region for the Redshift server cluster.

Note:
If the Server property is set, the ClusterID and Region properties are optional.

6. Set the DbUser property to the ID you want the Redshift user to use or have.
7. If the ID you specified for the DbUser property does not already exist in your Redshift account, you must create it:
   • Set the AutoCreate property to 1.
   • Set the DbGroups property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
 8. Set the IdP_Host property to the address of the service host.
 9. Set the IdP_Port property to the port number that the service listens at.
10. Set the Preferred_Role property to the name or ID for the IAM role that you want the user to assume when logged in to Redshift.
11. To skip verification of the SSL certificate of the IDP server, set the SSL_Insecure property to 1.

Using PingFederate Service

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in the PingFederate service.

To configure IAM authentication using PingFederate service on a non-Windows machine:

1. Set the UID property to the user name associated with your Ping account.
2. Set the PWD property to the password associated with your Ping user name.
3. Set the IAM property to 1.
4. Set the plugin_name property to ping.
5. Set the ClusterID property to the ID for the Redshift server cluster.
6. Set the Region property to the region for the Redshift server cluster.
7. Set the `DbUser` property to the ID you want the Redshift user to use or have.
8. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   - Set the `AutoCreate` property to 1.
   - Set the `DbGroups` property to the names of any user groups that you want the new `DbUser` to be added to, separated by commas.
9. Set the `IdP_Host` property to the address of the service host.
10. Set the `IdP_Port` property to the port number that the service listens at.
11. Set the `Preferred_Role` property to the name or ID for the IAM Role that you want the user to assume when logged in to Redshift.
12. To skip verification of the SSL certificate of the IDP server, set the `SSL_Insecure` property to 1.
13. Optionally, set the `partner_spid` property to a partner SPID (service provider ID) value.

### Using Okta Service

You can configure the driver to authenticate your connection through IAM authentication using the credentials stored in Okta.

**To configure IAM authentication using Okta on a non-Windows machine:**

1. Set the `UID` property to the user name associated with your Okta account.
2. Set the `PWD` property to the password associated with your Okta user name. If you are using a profile, this may be optional.
3. Set the `IAM` property to 1.
4. Set the `plugin_name` property to `okta`.
5. Set the `ClusterID` property to the ID for the Redshift server cluster.
6. Set the `Region` property to the region for the Redshift server cluster.

> **Note:**
> If the `Server` property is set, the `ClusterID` and `Region` properties are optional.

7. Set the `DbUser` property to the ID you want the Redshift user to use or have.
8. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
• Set the **AutoCreate** property to 1.
• Set the **DbGroups** property to the names of any user groups that you want the new **DbUser** to be added to, separated by commas.

9. Set the **IdP_Host** property to the address of the service host.
10. Set the **Preferred_Role** property to the name or ID for the IAM role that you want the user to assume when logged in to Redshift.
11. Set the **App_ID** property to the Okta-supplied ID associated with your Redshift application.
12. Optionally, set the **App_Name** property to the name of your Okta application.

### Using an External Credentials Service

In addition to built-in support for AD FS, PingFederate, and Okta, the Windows version of the Amazon Redshift ODBC Driver also provides support for other credentials services. The driver can authenticate connections using any SAML-based credential provider plugin of your choice.

**To configure an external credentials service on Windows:**

1. Create an IAM profile that specifies the credential provider plugin and other authentication parameters as needed. The profile must be ASCII-encoded, and must contain the following key-value pair, where `[PluginPath]` is the full path to the plugin application:

   ```
   plugin_name = [PluginPath]
   ```

   For example:

   ```
   plugin_name = C:\Users\jsmith\ApplicationInstallDir\CredServiceApplication.exe
   ```

   For information about how to create a profile, see "Using a Configuration Profile" in the *Amazon Redshift Cluster Management Guide*: https://docs.aws.amazon.com/redshift/latest/mgmt/options-for-providing-iam-credentials.html#using-configuration-profile.

2. Configure the driver to use this profile. For detailed instructions, see **Using an IAM Profile** on page 12.

The driver detects and uses the authentication settings specified in the profile.
Configuring Query Processing Modes on a Non-Windows Machine

To optimize driver performance, you can modify how the driver runs queries and retrieves results into memory. For example, you can configure the driver to return entire query results into memory all at once, or one row at a time. Use a query processing mode that prevents queries from consuming too much memory, based on the expected result size of your queries and the specifications of your system.

**Note:**

Use Single Row Mode if you plan to query large results and you do not want to retrieve the entire result into memory. Using the other query processing modes increases performance, but can result in out-of-memory errors.

You can set the connection properties described below in a connection string or in a DSN (in the odbc.ini file). Settings in the connection string take precedence over settings in the DSN.

### Enabling Single Row Mode

You can configure the driver to return query results one row at a time.

**To enable Single Row Mode:**

1. Set the `SingleRowMode` property to 1.
2. Make sure that the `UseDeclareFetch` property is set to 0 or not set.

### Enabling Declare/Fetch Mode

You can configure the driver to return a specific number of rows at a time.

**To enable Declare/Fetch Mode:**

1. Set the `UseDeclareFetch` property to 1.
2. Set the `Fetch` property to the number of rows that the driver returns at a time.

### Enabling Retrieve Entire Result Mode

You can configure the driver to return entire query results into memory.

**To enable Retrieve Entire Result Mode:**

1. Make sure that the `SingleRowMode`, `UseDeclareFetch`, and `UseMultipleStatements` properties are set to 0 or not set.
Enabling Multiple Statements Mode
You can enable the driver to have multiple queries active on the same connection. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially. When using this mode, the driver returns all the query results into memory.

To enable Multiple Statements Mode:
1. Set the `UseMultipleStatements` property to 1.
2. Make sure that the `SingleRowMode` and `UseDeclareFetch` properties are set to 0 or not set.

Enabling Enforce Single Statement Mode
You can configure the driver to allow only one active query at a time per connection.

To enable Enforce Single Statement Mode:
1. Set the `EnforceSingleStatement` property to 1.
2. Make sure that the `UseMultipleStatements` is set to 0 or not set.

Configuring a Proxy Connection on a Non-Windows Machine
You can configure the driver to connect to Redshift through a proxy server, so that communications between the driver and your Redshift data source are passed through the proxy server.

Note:
You can also configure the driver to pass IAM authentication processes through a proxy server. For more information, see Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine on page 47.

You can set the connection properties described below in a connection string or in a DSN (in the odbc.ini file). Settings in the connection string take precedence over settings in the DSN.

To configure a proxy connection on a non-Windows machine:
1. Set the `ProxyHost` property to the host name or IP address of the proxy server.
2. Set the `ProxyPort` property the number of the TCP port that the proxy server uses to listen for client connections.
3. If the proxy server requires authentication, then do the following:
   a. Set the `ProxyUid` property to your user name for accessing the proxy server.
   b. Set the `ProxyPwd` property to the password corresponding to the user name.

**Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine**

You can configure the driver to pass IAM authentication processes through a proxy server.

> **Note:**
>
> You can also configure the driver to connect to the data source through a proxy server, so that communications between the driver and your Redshift data source are passed through a proxy server. For more information, see Configuring a Proxy Connection on a Non-Windows Machine on page 46.

You can set the connection properties described below in a connection string or in a DSN (in the `odbc.ini` file). Settings in the connection string take precedence over settings in the DSN.

To configure an HTTPS proxy for IAM authentication on a non-Windows machine:

1. Set the `Https_Proxy_Host` property to the host name or IP address of the proxy server.
2. Set the `Https_Proxy_Port` property to the number of the port that the proxy server uses to listen for client connections.
3. If the proxy server requires authentication, then do the following:
   a. Set the `Https_Proxy_Username` property to your user name for accessing the proxy server.
   b. Set the `Https_Proxy_Password` property to the password corresponding to the user name.
4. To pass the authentication processes for identity providers through the proxy server, set the `IdP_USEHttps_Proxy` property to 1.
Configuring TCP Keepalives on a Non-Windows Machine

By default, the Amazon Redshift ODBC Driver is configured to use TCP keepalives to prevent connections from timing out. Settings such as how frequently the driver sends TCP keepalive packets are based on the operating system defaults.

You can set the connection properties described below in a connection string or in a DSN (in the odbc.ini file). Settings in the connection string take precedence over settings in the DSN.

To configure TCP keepalives on a non-Windows machine:

1. Set the KeepAliveIdle property to the number of seconds of inactivity before the driver sends a TCP keepalive packet.
2. Set the KeepAliveCount property to the number of keepalive packets that can be lost before the connection is considered broken.
3. Set the KeepAliveInterval property to the number of seconds to wait before each retransmission of a keepalive packet.

**Note:**
To use the system default for KeepAliveIdle, KeepAliveCount, or KeepAliveInterval, set the property to 0.

To disable TCP keepalives:

- Set the KeepAlive property to 0.

**Note:**
To enable TCP keepalives after disabling them, remove the KeepAlive property or set it to 1.

Configuring Single Statement Mode on a Non-Windows Machine

You can configure the driver to only allow one active query on a connection at a time.

To configure Single Statement Mode on a non-Windows machine:

1. Ensure that UseMultipleStatements is set to 0.
2. Set the enforceSingleStatement property to 1.
Configuring Logging Options

To help troubleshoot issues, you can enable logging in the driver.

**Important:**

Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.

The settings for logging apply to every connection that uses the Amazon Redshift ODBC Driver, so make sure to disable the feature after you are done using it.

Logging is configured through driver-wide settings in the `amazon.redshiftodbc.ini` file, which apply to all connections that use the driver.

To enable logging:

1. Open the `amazon.redshiftodbc.ini` configuration file in a text editor.
2. To specify the level of information to include in log files, set the `LogLevel` property to one of the following numbers:

<table>
<thead>
<tr>
<th>LogLevel Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disables all logging.</td>
</tr>
<tr>
<td>1</td>
<td>Logs severe error events that lead the driver to abort.</td>
</tr>
<tr>
<td>2</td>
<td>Logs error events that might allow the driver to continue running.</td>
</tr>
<tr>
<td>3</td>
<td>Logs events that might result in an error if action is not taken.</td>
</tr>
<tr>
<td>4</td>
<td>Logs general information that describes the progress of the driver.</td>
</tr>
<tr>
<td>5</td>
<td>Logs detailed information that is useful for debugging the driver.</td>
</tr>
<tr>
<td>6</td>
<td>Logs all driver activity.</td>
</tr>
</tbody>
</table>

3. Set the `LogPath` key to the full path to the folder where you want to save log files.
4. Set the `LogFileCount` key to the maximum number of log files to keep.
5. Set the **LogFileSize** key to the maximum size of each log file in bytes.

6. Optionally, to prefix the log file name with the user name and process ID associated with the connection, set the **UseLogPrefix** property to 1.

7. Save the **amazon.redshiftodbc.ini** configuration file.

8. Restart your ODBC application to make sure that the new settings take effect.

The Amazon Redshift ODBC Driver produces the following log files at the location you specify using the **LogPath** key:

- A **amazonredshiftodbcdriver.log** file that logs driver activity that is not specific to a connection.
- A **amazonredshiftodbcdriver_connection_[Number].log** file for each connection made to the database, where **[Number]** is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If you set the **UseLogPrefix** property to 1, then each file name is prefixed with **[UserName]_[ProcessID]_**, where **[UserName]** is the user name associated with the connection and **[ProcessID]** is the process ID of the application through which the connection is made. For more information, see **UseLogPrefix** on page 94.

To disable logging:

1. Open the **amazon.redshiftodbc.ini** configuration file in a text editor.
2. Set the **LogLevel** key to 0.
3. Save the **amazon.redshiftodbc.ini** configuration file.
4. Restart your ODBC application to make sure that the new settings take effect.

### Testing the Connection

To test the connection, you can use an ODBC-enabled client application. For a basic connection test, you can also use the test utilities that are packaged with your driver manager installation. For example, the iODBC driver manager includes simple utilities
called iodbctest and iodbctestw. Similarly, the unixODBC driver manager includes simple utilities called isql and iusql.

Using the iODBC Driver Manager

You can use the iodbctest and iodbctestw utilities to establish a test connection with your driver. Use iodbctest to test how your driver works with an ANSI application, or use iodbctestw to test how your driver works with a Unicode application.

Note:

There are 32-bit and 64-bit installations of the iODBC driver manager available. If you have only one or the other installed, then the appropriate version of iodbctest (or iodbctestw) is available. However, if you have both 32- and 64-bit versions installed, then you need to make sure that you are running the version from the correct installation directory.

For more information about using the iODBC driver manager, see http://www.iodbc.org.

To test your connection using the iODBC driver manager:

1. Run iodbctest or iodbctestw.
2. Optionally, if you do not remember the DSN, then type a question mark (?) to see a list of available DSNs.
3. Type the connection string for connecting to your data store, and then press ENTER. For more information, see Using a Connection String on page 53.

If the connection is successful, then the SQL> prompt appears.

Using the unixODBC Driver Manager

You can use the isql and iusql utilities to establish a test connection with your driver and your DSN. isql and iusql can only be used to test connections that use a DSN. Use isql to test how your driver works with an ANSI application, or use iusql to test how your driver works with a Unicode application.

Note:

There are 32-bit and 64-bit installations of the unixODBC driver manager available. If you have only one or the other installed, then the appropriate version of isql (or iusql) is available. However, if you have both 32- and 64-bit versions installed, then you need to make sure that you are running the version from the correct installation directory.

For more information about using the unixODBC driver manager, see http://www.unixodbc.org.
To test your connection using the unixODBC driver manager:

- Run `isql` or `iusql` by using the corresponding syntax:
  - `isql [DataSourceName]`
  - `iusql [DataSourceName]`

`[DataSourceName]` is the DSN that you are using for the connection.

If the connection is successful, then the `SQL>` prompt appears.

📝 Note:

For information about the available options, run `isql` or `iusql` without providing a DSN.
Using a Connection String

For some applications, you might need to use a connection string to connect to your data source. For detailed information about how to use a connection string in an ODBC application, refer to the documentation for the application that you are using.

The connection strings in the following sections are examples showing the minimum set of connection attributes that you must specify to successfully connect to the data source. Depending on the configuration of the data source and the type of connection you are working with, you might need to specify additional connection attributes. For detailed information about all the attributes that you can use in the connection string, see Driver Configuration Options on page 65.

DSN Connection String Example

The following is an example of a connection string for a connection that uses a DSN:

```
DSN=[DataSourceName]
```

[DataSourceName] is the DSN that you are using for the connection.

You can set additional configuration options by appending key-value pairs to the connection string. Configuration options that are passed in using a connection string take precedence over configuration options that are set in the DSN.

DSN-less Connection String Examples

Some applications provide support for connecting to a data source using a driver without a DSN. To connect to a data source without using a DSN, use a connection string instead.

⚠️ Important:

When you connect to the data store using a DSN-less connection string, the driver does not encrypt your credentials.

The placeholders in the examples are defined as follows, in alphabetical order:

- `[DatabaseName]` is the database that you want to access.
- `[IAMRole]` is the name or ID of the IAM role that you want to assume.
- `[IDP_PortNumber]` is the number of the TCP port used by the server that is hosting the identity provider service (AD FS, Ping, or Okta).
• [IDP_Server] is the IP address or host name of the server that is hosting the the
  identity provider service (AD FS, Ping, or Okta).
• [OktaAppID] is the app ID associated with your Okta application.
• [PortNumber] is the number of the TCP port that the Redshift server uses to listen
  for client connections.
• [PPort] is the number of the TCP port that the proxy server uses to listen for client
  connection.
• [PServer] is the IP address or host name of the proxy server to which you are
  connecting.
• [Server] is the endpoint of the Redshift server to which you are connecting.
• [UserID] is the user ID that you want to associate with your Redshift account.
• [YourAccessKey] is your IAM access key.
• [YourSecretKey] is your IAM secret key.
• [YourPassword] is the password corresponding to your user name.
• [YourProfileName] is the name of the IAM profile that contains your Redshift
  credentials.
• [YourUserName] is the user name that you use to authenticate your connection to
  Redshift. Depending on the authentication method being used, this may be the
  user name associated with your Redshift, AD FS, Ping, or Okta account.

Connecting to a Redshift Server Directly

The following is the format of a DSN-less connection string for a basic connection to a
Redshift server:

Driver=Amazon Redshift ODBC Driver;Server=[Server];
Port=[PortNumber];Database=[DatabaseName];
UID=[YourUserName];PWD=[YourPassword];

For example:

Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;
UID=amazon;PWD=amazon;

Connecting to a Redshift Server Through a Proxy Server

The following is the format of a DSN-less connection string for connecting to a Redshift
server through a proxy server:

Driver=Amazon Redshift ODBC Driver;Server=[Server];
Port=[PortNumber];Database=[DatabaseName];
For example:

Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;
UID=jsmith;PWD=amazon12345;ProxyHost=192.168.222.160;
ProxyPort=8000;

**Connecting to a Redshift Server using an IAM Profile**

You can authenticate the connection using IAM credentials stored in a chained roles profile or the Amazon EC2 instance profile. The following is the format of a DSN-less connection string for connecting to a Redshift server using a chained roles profile:

Driver=Amazon Redshift ODBC Driver;Server=[Server];
Port=[PortNumber];Database=[DatabaseName];IAM=1;
Profile=[YourProfileName];

For example:

Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;
Profile=amazon_admin;

As another example, using the Amazon EC2 instance profile instead:

Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;
InstanceProfile=1;
Connecting to a Redshift Server using IAM User Credentials

The following is the format of a DSN-less connection string for connecting to a Redshift server using an access key and secret key:

```
Driver=Amazon Redshift ODBC Driver;Server=[Server];
Port=[PortNumber];Database=[DatabaseName];IAM=1;
DbUser=[YourUserID];AccessKeyId=[YourAccessKey];
SecretAccessKey=[YourSecretKey];
```

For example:

```
Driver=Amazon Redshift ODBC Driver;Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;
DbUser=Amazon;AccessKeyId=AKIAIOSFODNN7EXAMPLE;
SecretAccessKey=wJalrXUttnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY;
```
Important:

- If you are using temporary credentials associated with an IAM role, then you must also set the SessionToken property to your temporary session token.
- If the specified user ID does not already exist, then you must configure the driver to create it. To do this, set the AutoCreate property to 1, and set the DbGroups property to the database security group or groups that you want the ID to be associated with.
- When you use this authentication method, the Server property is optional. However, if you omit the Server property, then you must set the ClusterID property to the name of your Redshift cluster and set the Region property to the AWS region where the cluster is located.

Connecting to a Redshift Server using Active Directory Federation Services (AD FS)

The following is the format of a DSN-less connection string for connecting to a Redshift server using AD FS:

```
Driver=Amazon Redshift ODBC Driver;Server=[Server];Port=[PortNumber];Database=[DatabaseName];IAM=1;plugin_name=adfs;DbUser=[UserID];IdP_Host=[IDP_Server];IdP_Port=[IDP_PortNumber];Preferred_Role=[IAMRole];
```

For example:

```
Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;plugin_name=adfs;DbUser=Amazon;IdP_Host=adfs.amazon.com;IdP_Port=1234;Preferred_Role=dbAdmin;
```

Important:

- If the specified user ID does not already exist, then you must configure the driver to create it. To do this, set the AutoCreate property to 1, and set the DbGroups property to the database security group or groups that you want the ID to be associated with.
- When you use this authentication method, the Server property is optional. However, if you omit the Server property, then you must set the ClusterID property to the name of your Redshift cluster and set the Region property to the AWS region where the cluster is located.
Connecting to a Redshift Server using the PingFederate Service

The following is the format of a DSN-less connection string for connecting to a Redshift server using the PingFederate service:

Driver=Amazon Redshift ODBC Driver;Server=[Server];
Port=[PortNumber];Database=[DatabaseName];IAM=1;
plugin_name=ping;UID=[YourUserName];PWD=[YourPassword];
DbUser=[UserID];IdP_Host=[IDP_Server];
IdP_Port=[IDP_PortNumber];Preferred_Role=[IAMRole];

For example:

Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;
plugin_name=ping;UID=jsmith;PWD=amazon12345;DbUser=Amazon;
IdP_Host=ping.amazon.com;IdP_Port=1234;
Preferred_Role=dbAdmin;

⚠ Important:

- If the specified user ID does not already exist, then you must configure the driver to create it. To do this, set the AutoCreate property to 1, and set the DbGroups property to the database security group or groups that you want the ID to be associated with.

- When you use this authentication method, the Server property is optional. However, if you omit the Server property, then you must set the ClusterID property to the name of your Redshift cluster and set the Region property to the AWS region where the cluster is located.

Connecting to a Redshift Server using the Okta Service

The following is the format of a DSN-less connection string for connecting to a Redshift server using Okta:

Driver=Amazon Redshift ODBC Driver;Server=[Server];
Port=[PortNumber];Database=[DatabaseName];IAM=1;
plugin_name=okta;UID=[YourUserName];PWD=[YourPassword];
DbUser=[UserID];IdP_Host=[IDP_Server];
IdP_Port=[IDP_PortNumber];Preferred_Role=[IAMRole];App_ID=[OktaAppID];

For example:
Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;
plugin_name=okta;UID=jsmith;PWD=amazon12345;DbUser=Amazon;
IdP_Host=okta.amazon.com;Preferred_Role=dbAdmin;
App_ID=mQkRa0qFRNy5hAc262lW;

**Important:**

- If the specified user ID does not already exist, then you must configure the driver to create it. To do this, set the `AutoCreate` property to 1, and set the `DbGroups` property to the database security group or groups that you want the ID to be associated with.
- When you use this authentication method, the `Server` property is optional. However, if you omit the `Server` property, then you must set the `ClusterID` property to the name of your Redshift cluster and set the `Region` property to the AWS region where the cluster is located.

**Connecting to a Redshift Server using an External Credentials Service**

Aside from using AD FS, PingFederate, or Okta, you can also configure the Windows driver to authenticate connections using any SAML-based credential provider plugin of your choice. To do this, create a profile that specifies the plugin, and then configure the driver to use the profile. For an example of the DSN-less connection string format that you would use to configure this type of connection, see [Connecting to a Redshift Server using an IAM Profile](#) on page 55.
For more information on the features of the Amazon Redshift ODBC Driver, see the following:

- **Query Processing Modes** on page 60
- **TCP Keepalives** on page 61
- **Data Types** on page 61
- **Security and Authentication** on page 63

### Query Processing Modes

To support performance tuning, the Amazon Redshift ODBC Driver provides different query processing modes that you can configure to modify how the driver runs queries and retrieves results into memory.

The following query processing modes are available:

- **Single Row Mode**: The driver returns query results one row at a time.
- **Declare/Fetch Mode**: The driver returns a user-specified number of rows at a time.
- **Retrieve Entire Result Mode**: The driver returns the entire query result into memory.
- **Multiple Statements Mode**: The driver can have multiple queries active on the same connection. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially. When using this mode, the driver returns all the query results into memory.
- **Enforce Single Statement Mode**: The driver allows only one active statement at a time for each connection. You can use this mode in conjunction with the Single Row, Declare/Fetch, and Retrieve Entire Result modes. If you attempt to set both the Enforce Single Statement and Multiple Statements modes, Multiple Statements Mode takes precedence.

By default, the driver does not allow more than one active query at a time, and returns the entire query result into memory.

Use a query processing mode that prevents queries from consuming too much memory, considering the expected result size of your queries and the specifications of your system.

For information about configuring how the driver processes queries, see Configuring Additional Options on Windows on page 19 if you are using the Windows version of
the driver, or see Configuring Query Processing Modes on a Non-Windows Machine on page 45 if you are using a non-Windows version of the driver.

TCP Keepalives

By default, the Amazon Redshift ODBC Driver is configured to use TCP keepalives to verify the status of a connection and prevent it from timing out. After you connect to a Redshift server, the driver automatically sends keepalive packets to the server. If the server does not respond, then the driver returns an indication that the connection is broken.

For information about configuring settings for TCP keepalives when using the Windows driver, see Configuring TCP Keepalives on Windows on page 20. For information about configuring settings for TCP keepalives when using the Linux or macOS driver, see Configuring TCP Keepalives on a Non-Windows Machine on page 48.

Data Types

The Amazon Redshift ODBC Driver supports many common data formats, converting between Redshift data types and SQL data types.

The table below lists the supported data type mappings.

<table>
<thead>
<tr>
<th>Redshift Type</th>
<th>SQL Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigInt</td>
<td>SQL_BIGINT</td>
</tr>
<tr>
<td>Boolean</td>
<td>SQL_VARCHAR</td>
</tr>
<tr>
<td></td>
<td>If the Show Boolean Column As String option (the BoolsAsChar key) is disabled, then SQL_BIT is returned instead.</td>
</tr>
</tbody>
</table>

Note:

If the Use Unicode option (the UseUnicode key) is enabled, then the driver returns SQL_WCHAR instead of SQL_CHAR, and SQL_WVARCHAR instead of SQL_VARCHAR.
<table>
<thead>
<tr>
<th>Redshift Type</th>
<th>SQL Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytea (escape and hex formats)</td>
<td>SQL_VARBINARY</td>
</tr>
<tr>
<td></td>
<td>If the Bytea As LongVarBinary option (the ByteaAsLongVarBinary key) is enabled, then SQL_LONGVARBINARY is returned instead.</td>
</tr>
<tr>
<td>Char</td>
<td>SQL_CHAR</td>
</tr>
<tr>
<td></td>
<td>• If the length of the column is greater than the Max Varchar (MaxVarchar) setting, then SQL_LONGVARCHAR is returned instead.</td>
</tr>
<tr>
<td></td>
<td>• If the Use Unicode option (the UseUnicode key) is enabled, then SQL_WCHAR is returned instead.</td>
</tr>
<tr>
<td></td>
<td>• If the Use Unicode option (the UseUnicode key) is enabled and the column length is greater than the Max Varchar (MaxVarchar) setting, then SQL_WLONGVARCHAR is returned instead.</td>
</tr>
<tr>
<td>Date</td>
<td>SQL_TYPE_DATE</td>
</tr>
<tr>
<td>Decimal</td>
<td>SQL_NUMERIC</td>
</tr>
<tr>
<td>Double Precision</td>
<td>SQL_DOUBLE</td>
</tr>
<tr>
<td>Integer</td>
<td>SQL_INTEGER</td>
</tr>
<tr>
<td>Real</td>
<td>SQL_REAL</td>
</tr>
<tr>
<td>SmallInt</td>
<td>SQL_SMALLINT</td>
</tr>
</tbody>
</table>
### Redshift Type

<table>
<thead>
<tr>
<th>Redshift Type</th>
<th>SQL Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>SQL_LONGVARCHAR</td>
</tr>
<tr>
<td></td>
<td>• If the Use Unicode option (the UseUnicode key) is enabled, then SQL_WLONGVARCHAR is returned instead.</td>
</tr>
<tr>
<td></td>
<td>• If the Text As LongVarChar option (the TextAsLongVarChar key) is disabled, then SQLVARCHAR is returned instead.</td>
</tr>
<tr>
<td></td>
<td>• If Use Unicode is enabled and Text As LongVarChar is disabled at the same time, then SQL_WVARCHAR is returned instead.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>SQLTYPE_TIMESTAMP or SQL_TIMESTAMP (ODBC 2.0)</td>
</tr>
<tr>
<td>VarChar</td>
<td>SQLVARCHAR</td>
</tr>
<tr>
<td></td>
<td>• If the length of the column is greater than the Max Varchar (MaxVarchar) setting, then SQL_LONGVARCHAR is returned instead.</td>
</tr>
<tr>
<td></td>
<td>• If the Use Unicode option (the UseUnicode key) is enabled, then SQL_WVARCHAR is returned instead.</td>
</tr>
<tr>
<td></td>
<td>• If the Use Unicode option (the UseUnicode key) is enabled and the column length is greater than the Max Varchar (MaxVarchar) setting, then SQL_WLONGVARCHAR is returned instead.</td>
</tr>
</tbody>
</table>

### Security and Authentication

To protect data from unauthorized access, Redshift data stores require all connections to be authenticated using user credentials. Some data stores also require connections to be made over the Secure Sockets Layer (SSL) protocol, either with or without one-way authentication. The Amazon Redshift ODBC Driver provides full support for these authentication protocols.

**Note:**

In this documentation, "SSL" refers to both TLS (Transport Layer Security) and SSL (Secure Sockets Layer). The driver supports TLS 1.0, 1.1, and 1.2. The SSL version used for the connection is the highest version that is supported by both the driver and the server.
The driver supports authenticating your connection using your Redshift user name and password, or using IAM authentication. For detailed configuration instructions, see Configuring Authentication on Windows on page 11 or Configuring Authentication on a Non-Windows Machine on page 39.

Additionally, the driver supports SSL connections with or without one-way authentication. If the server has an SSL-enabled socket, then you can configure the driver to connect to it.

It is recommended that you enable SSL whenever you connect to a server that is configured to support it. SSL encryption protects data and credentials when they are transferred over the network, and provides stronger security than authentication alone. For information about configuring SSL settings, see Configuring SSL Verification on Windows on page 10 or Configuring SSL Verification on a Non-Windows Machine on page 38.
Driver Configuration Options

Driver Configuration Options lists the configuration options available in the Amazon Redshift ODBC Driver alphabetically by field or button label. Options having only key names, that is, not appearing in the user interface of the driver, are listed alphabetically by key name.

When creating or configuring a connection from a Windows machine, the fields and buttons described below are available in the following dialog boxes:

- Amazon Redshift ODBC Driver DSN Setup
- Additional Options
- Data Type Configuration
- SSL Options
- Logging Options

When using a connection string or configuring a connection from a non-Windows machine, use the key names provided below.

Configuration Options Appearing in the User Interface

The following configuration options are accessible via the Windows user interface for the Amazon Redshift ODBC Driver, or via the key name when using a connection string or configuring a connection from a Linux or macOS computer:

- AccessKeyId on page 66
- Allow Self-Signed Server Certificate on page 67
- Auth Type on page 67
- Authentication Mode on page 68
- Bytea As LongVarBinary on page 69
- Cache Size on page 69
- Check Certificate Revocation on page 69
- Cluster ID on page 70
- Custom SSL Certificate Path on page 70
- Minimum TLS on page 78
- Okta App ID on page 79
- Okta App Name on page 79
- Partner SPID on page 79
- Password on page 80
- Preferred Role on page 80
- Port on page 81
- Profile Name on page 80
- Proxy Password on page 81
- Proxy Port on page 81
- Proxy Server on page 82
- Proxy Username on page 82
AccessKeyId

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessKeyId</td>
<td>None</td>
<td>Yes, if using IAM credentials for authentication.</td>
</tr>
</tbody>
</table>

Description

The IAM access key for the user or role. If this is specified, then SecretAccessKey must also be specified.
Allow Self-Signed Server Certificate

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowSelfSignedServerCert</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies whether the driver allows a connection to a Redshift server that uses a self-signed certificate.

- **Enabled (1):** The driver authenticates the Redshift server even if the server is using a self-signed certificate.
- **Disabled (0):** The driver does not allow self-signed certificates from the server.

**Note:**

This setting is applicable only when SSL is enabled and the system trust store is being used. For more information, see Use System Trust Store on page 88.

Auth Type

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Standard</td>
<td>Yes, when you configure a DSN using the Amazon Redshift ODBC Driver DSN Setup dialog box.</td>
</tr>
</tbody>
</table>

**Description**

This option specifies the authentication mode that the driver uses when you configure a DSN using the Amazon Redshift ODBC Driver DSN Setup dialog box:

- **Standard**: Standard authentication using your Redshift user name and password.
- **AWS Profile**: IAM authentication using a profile.
- **AWS IAM Credentials**: IAM authentication using IAM credentials.
- **Identity Provider: AD FS**: IAM authentication using Active Directory Federation Services (AD FS).
• **Identity Provider: PingFederate**: IAM authentication using PingFederate service.

• **Identity Provider: Okta**: IAM authentication using Okta service.

**Note:**

This option is available only when you configure a DSN using the Amazon Redshift ODBC Driver DSN Setup dialog box in the Windows driver.

When you configure a connection using a connection string or a non-Windows machine, the driver automatically determines whether to use Standard, AWS Profile, or AWS IAM Credentials authentication based on your specified credentials. To use an identity provider, you must set the `plugin_name` property. For more information, see `plugin_name` on page 93.

## Authentication Mode

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSLMode</td>
<td>verify-ca</td>
<td>No</td>
</tr>
</tbody>
</table>

### Description

The SSL certificate verification mode to use when connecting to Redshift. The following values are possible:

- **verify-full**: Connect only using SSL, a trusted certificate authority, and a server name that matches the certificate.
- **verify-ca**: Connect only using SSL and a trusted certificate authority.
- **require**: Connect only using SSL.
- **prefer**: Connect using SSL if available. Otherwise, connect without using SSL.
- **allow**: By default, connect without using SSL. If the server requires SSL connections, then use SSL.
- **disable**: Connect without using SSL.

**Note:**

For information about SSL support in Amazon Redshift, see "Connect Using SSL" in the Amazon Redshift Management Guide:

Bytea As LongVarBinary

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteaAsLongVarBinary</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies the SQL data type that the driver uses to return Bytea data.

- Enabled (1): The driver returns Bytea columns as SQL_LONGVARBINARY data.
- Disabled (0): The driver returns Bytea columns as SQL_VARBINARY data.

Cache Size

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fetch</td>
<td>100</td>
<td>Yes, if Declare/Fetch Mode is enabled.</td>
</tr>
</tbody>
</table>

Description

The number of rows that the driver returns when Declare/Fetch Mode is enabled. For more information, see Use Declare/Fetch on page 86.

Check Certificate Revocation

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckCertRevocation</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver checks to see if a certificate has been revoked while retrieving a certificate chain from the Windows Trust Store.

This option is only applicable if you are using a CA certificate from the Windows Trust Store (see Use System Trust Store on page 88).

- Enabled (1): The driver checks for certificate revocation while retrieving a certificate chain from the Windows Trust Store.
- Disabled (0): The driver does not check for certificate revocation while retrieving a certificate chain from the Windows Trust Store.

**Note:**
This option is only available on Windows.

### Cluster ID

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClusterID</td>
<td>None</td>
<td>Yes, if using IAM authentication and Server is not specified.</td>
</tr>
</tbody>
</table>

**Description**

The name of the Redshift cluster you want to connect to.

### Custom SSL Certificate Path

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSLCertPath</td>
<td>The location of the driver DLL file.</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

The full path of the file containing the root certificate for verifying the server.

If this option is not set, then the driver looks in the folder that contains the driver DLL file.

### Database

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>None</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Description**

The name of the Redshift database that you want to access.
# DbGroups

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbGroups</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

A comma-separated list of existing database group names that the DbUser joins for the current session. If not specified, defaults to PUBLIC.

# DbUser

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbUser</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

The user ID to use with your Redshift account. You can use an ID that does not currently exist if you have enabled the User Auto Create option (the `AutoCreate` property).

# Enable HTTPS Proxy For Federated Access

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Clear</td>
<td>Yes, if using the Additional Configuration dialog box to configure the driver to pass IAM authentication processes through a proxy.</td>
</tr>
</tbody>
</table>

**Description**

*Note:*

This option is used only when you configure proxy connections using the Additional Configuration dialog box.
This option specifies whether the driver passes the IAM authentication processes through a proxy server.

- **Enabled**: The driver passes IAM authentication processes through a proxy server.
- **Disabled**: The driver does not pass IAM authentication processes through a proxy server.

For information about how to specify the proxy server information, see Configuring Additional Options on Windows on page 19 and Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine on page 47.

**Enable Proxy For Amazon Redshift Connection**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Clear</td>
<td>Yes, if using the Additional Configuration dialog box to configure a proxy connection.</td>
</tr>
</tbody>
</table>

**Description**

*Note:*

This option is used only when you configure proxy connections using the Additional Configuration dialog box.

This option specifies whether the driver passes the connection to Redshift through a proxy server.

- **Enabled**: The driver passes the connection through a proxy server.
- **Disabled**: The driver does not pass the connection through a proxy server.

For information about configuring proxy connections, see Configuring Additional Options on Windows on page 19 and Configuring a Proxy Connection on a Non-Windows Machine on page 46.

**Enable Table Types**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableTableTypes</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>
Description

This option specifies whether the driver recognizes table type information from the data source. By default, the driver only recognizes a single, generic table type.

- **Enabled (1):** The driver recognizes the following table types: TABLE, VIEW, SYSTEM TABLE, EXTERNAL TABLE, and LOCAL TEMPORARY.
- **Disabled (0):** All tables returned from the data source have the generic type TABLE.

Encrypt Password

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>All Users Of This Machine</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies how the driver encrypts the credentials that are saved in the DSN:

- **Current User Only:** The credentials are encrypted, and can only be used by the current Windows user.
- **All Users Of This Machine:** The credentials are encrypted, but can be used by any user on the current Windows machine.

ℹ️ Important:

This option is available only when you configure a DSN using the Amazon Redshift ODBC Driver DSN Setup dialog box in the Windows driver. When you connect to the data store using a connection string, the driver does not encrypt your credentials.

Enforce Single Statement

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnforceSingleStatement</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver can have more than one active query at a time per connection.
- **Enabled (1):** The driver can have only one active query at a time.
- **Disabled (0):** The driver can have multiple active queries if Use Multiple Statements is enabled. For more information, see [Use Multiple Statements](#) on page 88.

**Note:**
If Enforce Single Statement and Use Multiple Statements are both enabled, Multiple Statements Mode takes precedence.

### HTTPS Proxy Password

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Https_Proxy_Password</td>
<td>None</td>
<td>Yes, if passing IAM authentication processes through a proxy server that requires authentication.</td>
</tr>
</tbody>
</table>

**Description**
The password that you use to access the proxy server.

### HTTPS Proxy Port

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Https_Proxy_Port</td>
<td>None</td>
<td>Yes, if passing IAM authentication processes through a proxy server.</td>
</tr>
</tbody>
</table>

**Description**
The number of the port that the proxy server uses to listen for client connections.
## HTTPS Proxy Server

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Https_Proxy_Host</td>
<td>None</td>
<td>Yes, if passing IAM authentication processes through a proxy server.</td>
</tr>
</tbody>
</table>

**Description**

The host name or IP address of a proxy server through which you want to pass IAM authentication processes.

## HTTPS Proxy Username

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Https_Proxy_Username</td>
<td>None</td>
<td>Yes, if passing IAM authentication processes through a proxy server that requires authentication.</td>
</tr>
</tbody>
</table>

**Description**

The user name that you use to access the proxy server.

## IdP Host

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdP_Host</td>
<td>None</td>
<td>Yes, if using a credentials service for authentication.</td>
</tr>
</tbody>
</table>

**Description**

The IdP (identity provider) host you are using to authenticate into Redshift.
IdP Port

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdP_Port</td>
<td>None</td>
<td>Yes, if using a credentials service for authentication.</td>
</tr>
</tbody>
</table>

Description

The port for an IdP (identity provider).

Log Level

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogLevel</td>
<td>OFF (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

Use this property to enable or disable logging in the driver and to specify the amount of detail included in log files.

⚠️ Important:

- Only enable logging long enough to capture an issue. Logging decreases performance and can consume a large quantity of disk space.
- The settings for logging apply to every connection that uses the Amazon Redshift ODBC Driver, so make sure to disable the feature after you are done using it.
- This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the `amazon.redshiftodbc.ini` file.

Set the property to one of the following values:

- OFF (0): Disable all logging.
- FATAL (1): Logs severe error events that lead the driver to abort.
- ERROR (2): Logs error events that might allow the driver to continue running.
- WARNING (3): Logs events that might result in an error if action is not taken.
- INFO (4): Logs general information that describes the progress of the driver.
• DEBUG (5): Logs detailed information that is useful for debugging the driver.
• TRACE (6): Logs all driver activity.

When logging is enabled, the driver produces the following log files at the location you specify in the Log Path (LogPath) property:

• A `amazonredshiftodbcdriver.log` file that logs driver activity that is not specific to a connection.
• A `amazonredshiftodbcdriver_connection_[Number].log` file for each connection made to the database, where `[Number]` is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If you enable the UseLogPrefix connection property, the driver prefixes the log file name with the user name associated with the connection and the process ID of the application through which the connection is made. For more information, see UseLogPrefix on page 94.

### Log Path

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogPath</td>
<td>None</td>
<td>Yes, if logging is enabled.</td>
</tr>
</tbody>
</table>

**Description**

The full path to the folder where the driver saves log files when logging is enabled.

**Important:**

This option is not supported in connection strings. To configure logging for the Windows driver, you must use the Logging Options dialog box. To configure logging for a non-Windows driver, you must use the `amazon.redshiftodbc.ini` file.

### Max Bytea

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxBytea</td>
<td>255</td>
<td>No</td>
</tr>
</tbody>
</table>
**Description**
The maximum data length for Bytea columns.

**Max LongVarChar**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxLongVarChar</td>
<td>8190</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**
The maximum data length for LongVarChar columns.

**Max Varchar**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxVarchar</td>
<td>255</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**
The maximum data length for VarChar columns.

**Minimum TLS**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min_TLS</td>
<td>TLS 1.0 (1.0)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**
The minimum version of TLS/SSL that the driver allows the data store to use for encrypting connections. For example, if TLS 1.1 is specified, TLS 1.0 cannot be used to encrypt connections.

- TLS 1.0 (1.0): The connection must use at least TLS 1.0.
- TLS 1.1 (1.1): The connection must use at least TLS 1.1.
- TLS 1.2 (1.2): The connection must use at least TLS 1.2.
Okta App ID

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>App_ID</td>
<td>None</td>
<td>Yes, if authenticating through the Okta service.</td>
</tr>
</tbody>
</table>

Description

The Okta-provided unique ID associated with your Redshift application.

Okta App Name

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>App_Name</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

The name of the Okta application that you use to authenticate the connection to Redshift.

Partner SPID

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>partner_spid</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

The partner SPID (service provider ID) value to use when authenticating the connection using the PingFederate service.
Password

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWD OR Password</td>
<td>None</td>
<td>Yes, if User has been set.</td>
</tr>
</tbody>
</table>

Description

The password corresponding to the user name that you provided in the User field (the Username or UID key).

Preferred Role

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred_Role</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

The role you want to assume during the connection to Redshift.

Profile Name

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

The name of the user profile used to authenticate into Redshift.
Note:

- If the Use Instance Profile option (the InstanceProfile property) is enabled, that setting takes precedence and the driver uses the Amazon EC2 instance profile instead.
- The default location for the credentials file that contains profiles is `~/.aws/Credentials`. The AWS_SHARED_CREDENTIALS_FILE environment variable can be used to point to a different credentials file.

Port

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>5439</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Description

The number of the TCP port that the Redshift server uses to listen for client connections.

Proxy Password

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyPwd</td>
<td>None</td>
<td>Yes, if connecting to a proxy server that requires authentication.</td>
</tr>
</tbody>
</table>

Description

The password that you use to access the proxy server.

Proxy Port

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyPort</td>
<td>None</td>
<td>Yes, if connecting through a proxy server.</td>
</tr>
</tbody>
</table>
Description

The number of the port that the proxy server uses to listen for client connections.

Proxy Server

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyHost</td>
<td>None</td>
<td>Yes, if connecting through a proxy server.</td>
</tr>
</tbody>
</table>

Description

The host name or IP address of a proxy server that you want to connect through.

Proxy Username

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyUid</td>
<td>None</td>
<td>Yes, if connecting to a proxy server that requires authentication.</td>
</tr>
</tbody>
</table>

Description

The user name that you use to access the proxy server.

Region

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>None</td>
<td>Yes, if using IAM authentication and Server is not specified.</td>
</tr>
</tbody>
</table>

Description

The AWS region that your cluster is in.
Retrieve Entire Result Into Memory

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Selected (1)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies whether the driver returns the entire query result into memory.

- Enabled (1): The driver returns the entire query result into memory.
- Disabled (0): The driver returns the query result in chunks or single rows.

When using keys to set driver options, you can enable this option by setting the `SingleRowMode`, `UseDeclareFetch`, and `UseMultipleStatements` keys to 0.

**Note:**

When using connection attributes to set driver options, you can enable this option by setting the `SingleRowMode`, `UseDeclareFetch`, and `UseMultipleStatements` attributes to 0.

**SecretAccessKey**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecretAccessKey</td>
<td>None</td>
<td>Yes, if using IAM credentials for authentication.</td>
</tr>
</tbody>
</table>

**Description**

The IAM secret key for the user or role. If this is specified, AccessKeyId must also be specified.

**SessionToken**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionToken</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>
**Description**

The temporary IAM session token associated with the IAM role you are using to authenticate.

**Server**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>None</td>
<td>Yes, unless AWS Region and Cluster ID are specified.</td>
</tr>
</tbody>
</table>

**Description**

A comma-delimited list of endpoint servers. The driver attempts to connect to each server in the order specified until it finds a valid server or the list has been exhausted. If a valid server cannot be found the driver alerts the user.

**Note:**

If you are using IAM authentication you can only specify one server, not a list.

**Show Boolean Column As String**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoolsAsChar</td>
<td>Selected (1)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies the SQL data type that the driver uses to return Boolean data.

- **Enabled (1):** The driver returns Boolean columns as SQL_VARCHAR data with a length of 5.
- **Disabled (0):** The driver returns Boolean columns as SQL_BIT data.
Single Row Mode

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SingleRowMode</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver uses Single Row Mode and returns query results one row at a time. Enable this option if you plan to query large results and do not want to retrieve the entire result into memory.

- Enabled (1): The driver returns query results one row at a time.
- Disabled (0): The driver returns all query results at once.

When using connection attributes to set driver options, make note of the following:

- If `SingleRowMode` and `UseDeclareFetch` are both set to 0, then the driver retrieves the entire query result into memory.
- If `UseDeclareFetch` is set to 1, then it takes precedence over `SingleRowMode`.
- If `SingleRowMode` is set to 1 and `UseDeclareFetch` is set to 0, then `SingleRowMode` takes precedence over `UseMultipleStatements`.

SSL Insecure

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL_Insecure</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver checks the authenticity of the IdP server certificate.

- Enabled (1): The driver does not check the authenticity of the IdP server certificate.
- Disabled (0): The driver checks the authenticity of the IdP server certificate.
**Text As LongVarChar**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextAsLongVarChar</td>
<td>Selected (1)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies the SQL data type that the driver uses to return Text data. The returned data type is also affected by the Use Unicode option (the `UseUnicode` key). For more information, see **Use Unicode** on page 89.

- **Enabled (1):** The driver returns Text columns as SQL_LONGVARCHAR data. If the Use Unicode option (the `UseUnicode` key) is also enabled, then the driver returns SQL_WLONGVARCHAR data instead.
- **Disabled (0):** The driver returns Text columns as SQL_VARCHAR data. If the Use Unicode option (the `UseUnicode` key) is also enabled, then the driver returns SQL_WVARCHAR data instead.

**Use Declare/Fetch**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UseDeclareFetch</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies whether the driver uses Declare/Fetch Mode and returns a specific number of rows at a time.

- **Enabled (1):** The driver uses Declare/Fetch Mode and returns a specific number of rows at a time. To specify the number of rows, configure the Cache Size option (the `Fetch` attribute).
- **Disabled (0):** The driver returns all rows at once.

When using keys to set driver options, make note of the following:

- If `UseDeclareFetch` is set to 1, then it takes precedence over `SingleRowMode` and `UseMultipleStatements`.
- If `UseDeclareFetch` is set to 0 and `SingleRowMode` is set to 1, then the driver returns query results one row at a time.
- If `UseDeclareFetch` and `SingleRowMode` are both set to 0, then the driver retrieves the entire query result into memory.
Use HTTPS Proxy For Authentication On IdP

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>IdP_Use_Https_Proxy</td>
<td>Clear (0)</td>
<td>Yes, if authenticating through an identity provider that can only be reached through a proxy connection.</td>
</tr>
</tbody>
</table>

**Description**

This option specifies whether the driver passes the authentication processes for identity providers (IdP) through a proxy server.

- **Enabled (1):** The driver passes IdP authentication processes through a proxy server.
- **Disabled (0):** The driver does not pass IdP authentication processes through a proxy server.

For information about how to specify the proxy server information, see Configuring Additional Options on Windows on page 19 and Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine on page 47.

Use Instance Profile

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceProfile</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies whether the driver uses the Amazon EC2 instance profile, when configured to use a profile for authentication.

- **Enabled (1):** The driver uses the Amazon EC2 instance profile.
- **Disabled (0):** The driver uses the chained roles profile specified by the Profile Name option (the Profile property) instead. For more information, see Profile Name on page 80.
Use Multiple Statements

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UseMultipleStatements</td>
<td>Enabled (1)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver can have more than one active query at a time per connection.

- **Enabled (1):** The driver can have multiple queries active on the same connection. The ODBC application may interleave calls to ODBC statements, but all queries are still sent and executed sequentially. The driver returns all the query results into memory.
- **Disabled (0):** The driver executes queries one at a time.

When using connection attributes to set driver options, make note of the following:

- If **UseDeclareFetch** is set to 1, then it takes precedence over **UseMultipleStatements**.
- If **UseDeclareFetch** is set to 0 and **SingleRowMode** is set to 1, then **SingleRowMode** takes precedence over **UseMultipleStatements**.

Use System Trust Store

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UseSystemTrustStore</td>
<td>Selected (1)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether to use a CA certificate from the system trust store, or from a specified .pem file.

- **Enabled (1):** The driver verifies the connection using a certificate in the system trust store.
- **Disabled (0):** The driver verifies the connection using a specified .pem file. For information about specifying a .pem file, see Custom SSL Certificate Path on page 70.
Use Unicode

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UseUnicode</td>
<td>Selected (1)</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver returns Redshift data as Unicode or regular SQL types.

- **Enabled (1):** The driver returns data as Unicode character types:
  - SQL_WCHAR is returned instead of SQL_CHAR.
  - SQL_WVARCHAR is returned instead of SQL_VARCHAR.
  - SQL_WLONGVARCHAR is returned instead of SQL_LONGVARCHAR.
- **Disabled (0):** The driver returns data as regular SQL types:
  - SQL_CHAR is returned instead of SQL_WCHAR.
  - SQL_VARCHAR is returned instead of SQL_WVARCHAR.
  - SQL_LONGVARCHAR is returned instead of SQL_WLONGVARCHAR.

For detailed information about how the driver returns Redshift data as SQL types, see Data Types on page 61.

User

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>OR</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Description

The user name that you use to access the Redshift server.

If you are using keys to set driver options, **UID** takes precedence over **Username**.
If you are using IAM authentication, can be used in the following ways:

- If the connection uses a credential provider plugin, this will be the user name for the idp_host server. In this case the information can be included in a user profile and may not be required for the connection URL.
- If your connection does not use a credential provider, this is used as the user name for your data source or UID.

If this value is defined in multiple places, the preference order will be: DbUser > user > UID.

**User AutoCreate**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCreate</td>
<td>Clear (0)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

This option specifies whether the driver causes a new user to be created when the specified user does not exist.

- **Enabled (1):** If the user specified by either DbUser or UID does not exist, a new user with that name is created.
- **Disabled (0):** The driver does not cause new users to be created. If the specified user does not exist, the authentication fails.

**Configuration Options Having Only Key Names**

The following configuration options do not appear in the Windows user interface for the Amazon Redshift ODBC Driver. They are accessible only when you use a connection string or configure a connection on macOS or Linux.

- **cafile** on page 91
- **Driver** on page 91
- **IAM** on page 92
- **KeepAlive** on page 92
- **KeepAliveCount** on page 92
- **KeepAliveInterval** on page 93
- **KeepAliveTime** on page 93
- **Locale** on page 93
- **plugin_name** on page 93
The `UseLogPrefix` property must be configured as a Windows Registry key value, or as a driver-wide property in the `amazon.redshiftodbc.ini` file for macOS or Linux.

- `UseLogPrefix` on page 94

**cafile**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>cafile</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

The file path to the CA certificate file used for some forms of IAM authentication.

⚠️ **Note:**

This option is only available on macOS and Linux.

**Driver**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>Amazon Redshift ODBC Driver when installed on Windows, or the absolute path of the driver shared object file when installed on a non-Windows machine.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Description**

On Windows, the name of the installed driver (Amazon Redshift ODBC Driver).

On other platforms, the name of the installed driver as specified in `odbcinst.ini`, or the absolute path of the driver shared object file.
IAM

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This property specifies whether the driver uses an IAM authentication method to authenticate the connection.

- 0: The driver uses standard authentication (using your database user name and password).
- 1: The driver uses one of the IAM authentication methods (using an access key and secret key pair, or a profile, or a credentials service).

KeepAlive

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeepAlive</td>
<td>1</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

When this option is enabled (1), the driver uses TCP keepalives to prevent connections from timing out.

When this option is disabled (0), the driver does not use TCP keepalives.

KeepAliveCount

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeepAliveCount</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

The number of TCP keepalive packets that can be lost before the connection is considered broken.

When this key is set to 0, the driver uses the system default for this setting.
**KeepAliveTime**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeepAliveTime</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

The number of seconds of inactivity before the driver sends a TCP keepalive packet. When this key is set to 0, the driver uses the system default for this setting.

**KeepAliveInterval**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeepAliveInterval</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

The number of seconds between each TCP keepalive retransmission. When this key is set to 0, the driver uses the system default for this setting.

**Locale**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locale</td>
<td>en-US</td>
<td>No</td>
</tr>
</tbody>
</table>

**Description**

The locale to use for error messages.

**plugin_name**

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>plugin_name</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>
Description

A string indicating the credentials provider plugin class that you want to use for authentication. The following values are supported:

- `adfs`: Use Active Directory Federation Services for authentication.
- `ping`: Use the PingFederate service for authentication.
- `okta`: Use the Okta service for authentication.

On Windows, you can use other SAML-based credential provider plugins by setting this property to the full path to the plugin application. For more information, see Using an External Credentials Service on page 44.

⚠️ Note:

This property is applicable only when you configure a connection using a connection string or a non-Windows machine.

When you configure a connection using the Amazon Redshift ODBC Driver DSN Setup dialog box in the Windows driver, the Auth Type option is used instead. For more information, see Auth Type on page 67.

UseLogPrefix

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UseLogPrefix</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Description

This option specifies whether the driver includes a prefix in the names of log files so that the files can be distinguished by user and application.

Set the property to one of the following values:

- `1`: The driver prefixes log file names with the user name and process ID associated with the connection that is being logged.

  For example, if you are connecting as a user named "jdoe" and using the driver in an application with process ID 7836, the generated log files would be named `jdoe_7836_amazonredshiftodbcdriver.log` and `jdoe_7836_amazonredshiftodbcdriver_connection_[Number].log`, where `[Number]` is a number that identifies each connection-specific log file.

- `0`: The driver does not include the prefix in log file names.
To configure this option for the Windows driver, you create a value for it in one of the following registry keys:

- For a 32-bit driver installed on a 64-bit machine: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Amazon\Amazon Redshift ODBC Driver\Driver
- Otherwise: HKEY_LOCAL_MACHINE\SOFTWARE\Amazon\Amazon Redshift ODBC Driver\Driver

Use UseLogPrefix as the value name, and either 0 or 1 as the value data.

To configure this option for a non-Windows driver, you must use the amazon.redshiftodbc.ini file.
Contact Us

For support, check the EMR Forum at https://forums.aws.amazon.com/forum.jspa?forumID=52 or open a support case using the AWS Support Center at https://aws.amazon.com/support
Third-Party Trademarks

Simba, the Simba logo, SimbaEngine, SimbaEngine C/S, SimbaExpress and SimbaLib are registered trademarks of Simba Technologies Inc. All other trademarks and/or servicemarks are the property of their respective owners.

Linux is the registered trademark of Linus Torvalds in Canada, United States and/or other countries.

Mac, macOS, Mac OS, and OS X are trademarks or registered trademarks of Apple, Inc. or its subsidiaries in Canada, United States and/or other countries.

Microsoft, MSDN, Windows, Windows Server, Windows Vista, and the Windows start button are trademarks or registered trademarks of Microsoft Corporation or its subsidiaries in Canada, United States and/or other countries.

Red Hat, Red Hat Enterprise Linux, and CentOS are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in Canada, United States and/or other countries.

SUSE is a trademark or registered trademark of SUSE LLC or its subsidiaries in Canada, United States and/or other countries.

All other trademarks are trademarks of their respective owners.