

Amazon Redshift ODBC Data Connector

Installation and Configuration Guide

Version 1.6

September 2026

Copyright © 2014-2026 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.insightsoftware.com.

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.

About This Guide

Purpose

The *Amazon Redshift ODBC Data Connector Installation and Configuration Guide* explains how to install and configure the Amazon Redshift ODBC Data Connector. The guide also provides details related to features of the connector.

Audience

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

Knowledge Prerequisites

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

- Familiarity with the platform on which you are using the Amazon Redshift ODBC Connector
- Ability to use the data source to which the Amazon Redshift ODBC Connector 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 is 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.

Contents

About This Guide **3**

Purpose 3

Audience 3

Knowledge Prerequisites 3

Document Conventions 3

Contents **4**

About the Amazon Redshift ODBC Connector **7**

Platform and Data source version support **8**

Windows Connector **9**

Windows System Requirements 9

Installing the Connector in Windows 9

Creating a Data Source Name in Windows 10

Configuring SSL Verification in Windows 11

Configuring Authentication in Windows 12

Configuring Data Type Options in Windows 25

Configuring Additional Options in Windows 25

Configuring TCP Keepalives in Windows 27

Configuring Logging Options in Windows 28

Verifying the Connector Version Number in Windows 31

macOS Connector **32**

macOS System Requirements 32

Installing the Connector in macOS 32

Verifying the Connector Version Number in macOS 32

M1 Connector **34**

M1 System Requirements 34

Installing the Connector on M1 34

Verifying the Connector Version Number on M1 34

Linux Connector **36**

Linux System Requirements 36

Installing the Connector Using the RPM File 36

Verifying the Connector Version Number in Linux 37

Configuring the ODBC Driver Manager in Non-Windows Machines **38**

Specifying ODBC Driver Managers in Non-Windows Machines 38

Specifying the Locations of the Connector Configuration Files 39

Configuring ODBC Connections in Non-Windows Machine **41**

Creating a Data Source Name on a Non-Windows Machine 41

Configuring a DSN-less Connection on a Non-Windows Machine 44

Configuring SSL Verification on a Non-Windows Machine 45

Configuring Authentication on a Non-Windows Machine 46

Configuring Query Processing Modes on a Non-Windows Machine 58

Configuring a Proxy Connection on a Non-Windows Machine 59

Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine 60

Configuring TCP Keepalives on a Non-Windows Machine 61

Configuring Single Statement Mode on a Non-Windows Machine 61

Configuring Logging Options 61

Testing the Connection 63

Using a Connection String **65**

DSN Connection String Example 65

DSN-less Connection String Examples 65

Features **72**

Query Processing Modes 72

TCP Keepalives 73

Data Types 73

Security and Authentication 75

Connector Configuration Properties **76**

Configuration Options Appearing in the User Interface 76

Configuration Options Having Only Key Names 100

Contact Us **106**

Third-Party Trademarks **107**

About the Amazon Redshift ODBC Connector

The Amazon Redshift ODBC Connector enables Business Intelligence (BI), analytics, and reporting on data that is stored in Amazon Redshift. The connector 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 connector, 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 Connector 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.

Platform and Data source version support

The Amazon Redshift ODBC Connector supports Windows, macOS, M1 and Linux operating systems. For detailed information on supported operating systems and data source versions, please refer to the connector's release notes.

Windows Connector

This section provides an overview of the Connector in the Windows platform, outlining the required system specifications and the steps for installing and configuring the connector in Windows environments.

Windows System Requirements

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

- Administrator rights on your machine.
- A machine that meets the following system requirements:
  - 100 MB of available disk space

Before the connector can be used, the Visual C++ Redistributable for Visual Studio 2022 with the same bitness as the connector must also be installed. If you obtained the connector from the Simba website, then your installation of the connector automatically includes this dependency. Otherwise, you must install the redistributable manually. You can download the installation packages for the redistributable at https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170.

Installing the Connector in Windows

If you did not obtain this connector from the Simba website, you might need to follow a different installation procedure. For more information, see the *OEM ODBC Driver Installation Instructions*.

On 64-bit Windows operating systems, you can execute both 32-bit and 64-bit applications. However, 64-bit applications must use 64-bit connectors, and 32-bit applications must use 32-bit connectors. Make sure that you use a connector 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 connector on the same machine.

To install the Amazon Redshift ODBC Connector in 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 in Windows

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

Alternatively, for information about DSN-less connections, see Using a Connection String.

To create a Data Source Name in 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 connectors 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.

   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.

8. In the **Port** field, type the number of the TCP port that the server uses to listen for client connections.

   Note: The default port used by Redshift is 5439.

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 in Windows.

11. To configure client-server verification over SSL, click **SSL Options**. For more information, see Configuring SSL Verification in Windows.
12. To configure advanced connector options, click **Additional Options**. For more information, see Configuring Additional Options in Windows.
13. To configure logging behavior for the connector, click **Logging Options**. For more information, see Configuring Logging Options in Windows.
14. To configure how the connector returns and displays data, click **Data Type Options**. For more information, see Configuring Data Type Options in Windows.
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 in Windows

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

To configure SSL verification in 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 specify the minimum version of SSL to use, from the **Minimum TLS** drop-down list, select the minimum version of SSL.
4. To use the System Trust Store for SSL certificates, select the **Use System Trust Store** check box.
5. 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.
6. 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.
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 in Windows

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

The connector supports the following authentication methods:

- Standard authentication using your database user name and password (see Using Standard Authentication)
- IAM authentication using a profile (see Using an IAM Profile)
- IAM authentication using IAM credentials (see Using IAM Credentials)
- IAM authentication using Active Directory Federation Services (AD FS) (see Using Active Directory Federation Services (AD FS))
- IAM authentication using Azure AD service (see Using Azure AD Service)
- IAM authentication using a JSON Web Token (JWT) (see Using a JSON Web Token (JWT))
- IAM authentication using Okta service (see Using Okta Service)
- IAM authentication using PingFederate service (see Using PingFederate Service in Windows)
- IAM authentication using a browser plugin for Azure AD (see Using a Browser Plugin for Azure AD)
- IAM authentication using a browser plugin for Azure AD OAuth2 (see Using a Browser Plugin for Azure AD OAuth2)
- IAM authentication using a browser plugin for a SAML service (see Using a Browser Plugin for a SAML Service)
- IAM authentication using a credentials service aside from those listed above (see Using an External Credentials Service)

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 connector to authenticate your connection using your Redshift user name and password.

To configure standard authentication in 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 connector 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 in 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**.
6. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
   a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
   b. In the **Region** field, type the region for the Redshift server cluster.

7. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
9. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
10. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
11. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.
12. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
13. Optionally, to use group federation, select the **Group Federation** checkbox.
14. 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.

    Note: If you configure both options, the Use Instance Profile option takes precedence and the connector uses the Amazon EC2 instance profile.
15. To save your settings and close the dialog box, click **OK**.

Using IAM Credentials

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

To configure IAM authentication using IAM in 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**.
3. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:

   a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
   b. In the **Region** field, type the region for the Redshift server cluster.
4. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
5. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
6. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
7. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
8. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.
9. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings.
10. In the **AccessKeyID** field, type your Redshift access key ID.
11. In the **SecretAccessKey** field, type your Redshift secret key.
12. If you are using an IAM role, in the **SessionToken** field, type your temporary session token.
13. Optionally, to use group federation, select the **Group Federation** checkbox.
14. To save your settings and close the dialog box, click **OK**.

Using Active Directory Federation Services (AD FS)

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

To configure IAM authentication using AD FS in 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:
     - In the **User** field, type the user name associated with your AD FS account.
     - 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**.
5. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
   a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
   b. In the **Region** field, type the region for the Redshift server cluster.
6. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
7. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
8. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
9. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
10. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.
11. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
12. In the **IdP Host** field, type the address of the service host.
13. In the **IdP Port** field, type the port number the service listens at.
14. To skip verification of the SSL certificate of the IDP server, select the **SSL Insecure** check box.

15. 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.
16. Optionally, in the **Login To RP** field, type the relying party trust you want to use.
17. To save your settings and close the dialog box, click **OK**.

Using Azure AD Service

You can configure the connector to authenticate your connection through IAM authentication using the Azure AD service.

To configure IAM authentication using Azure AD in 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: Azure AD**.
3. In the **User** field, type the user name associated with your Redshift application on Azure AD.
4. In the **Password** field, type the password associated with your Redshift application on Azure AD.
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. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
   a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
   b. In the **Region** field, type the region for the Redshift server cluster.
7. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
9. In the **DbGroups Filter** field, type the DbGroup filter you want to use.
10. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
11. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
12. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.

13. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
14. In the **Azure Client ID** field, type the client ID associated with your Redshift application on Azure AD.
15. In the **Azure Client Secret** field, type the client secret associated with your Redshift application on Azure AD.
16. In the **Preferred Role** field, type the name or ID for the IAM role you want the user to assume when logged into Redshift.
17. In the **IdP Tenant** field, type the Azure AD tenant ID associated with your application.
18. To save your settings and close the dialog box, click **OK**.

Using a JSON Web Token (JWT)

You can configure the connector to authenticate your connection by using a token obtained from the web identity provider.

To configure IAM authentication using a JWT in 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: JWT or JWT IAM Auth Plugin**.
3. Optionally, if the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
   a. In the Cluster ID field, type the ID for the Redshift server cluster.
   b. In the Region field, type the region for the Redshift server cluster.
4. Optionally, in the **DbUser** field, type the ID that you want to designate to the Redshift user.
5. Optionally, if the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new **DbUser** to be added to, separated by commas.
   c. Optionally, to lowercase all **DbGroups** that are received from the identity provider, select the **Force Lowercase** check box.
6. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
7. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
8. Optionally, in the **VPC Endpoint URL** field, type the endpoint used to communicate with the Redshift cluster.

9. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
10. In the **Web Identity Token** field, type the token that is provided by the identity provider.
11. Optionally, to use group federation, select the **Group Federation** checkbox.
12. Optionally, in the **Role ARN** field, type the Amazon Resource Name (ARN) of the role.
13. Optionally, in the **Role Session Name** field, type the name of the assumed role session.
14. Optionally, in the **Duration** field, type the duration of the role session, in seconds.
15. Optionally, in the **Provider Name** field, type the name of the authentication provider created from the CREATE IDENTITY PROVIDER query.
16. To save your settings and close the dialog box, click **OK**.

Using Okta Service

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

To configure IAM authentication using Okta in 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: 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**.
6. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
    a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
    b. In the **Region** field, type the region for the Redshift server cluster.
7. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:

a. Select the **User AutoCreate** check box.

b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.

c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.

9. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
10. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
11. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.
12. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
13. In the **IdP Host** field, type the address of the service host.
14. 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.
15. In the **Okta App ID** field, type the Okta-supplied ID associated with your Redshift application.
16. Optionally, in the **Okta App Name** field, type the name of your Okta application.
17. To save your settings and close the dialog box, click **OK**.

Using PingFederate Service in Windows

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

To configure IAM authentication using PingFederate service in 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: 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.
5. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
    a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
    b. In the **Region** field, type the region for the Redshift server cluster.
6. In the **DbUser** field, type the ID that you want to designate to the Redshift user.

7. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
8. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
9. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
10. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.
11. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
12. In the **IdP Host** field, type the address of the service host.
13. In the **IdP Port** field, type the port number the service listens at.
14. To skip verification of the SSL certificate of the IDP server, select the **SSL Insecure** check box.
15. 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.
16. Optionally, in the **Partner SPID** field, type a partner SPID (service provider ID) value.
17. To save your settings and close the dialog box, click **OK**.

Using a Browser Plugin for Azure AD

You can configure the connector to use a browser plugin to authenticate your connection through the Azure AD website.

To configure IAM authentication using a browser plugin for Azure AD in 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: Browser Azure AD**.
3. In the **User** field, type the user name associated with your Redshift application for Azure AD.
4. In the **Password** field, type the password associated with your Redshift application for Azure AD.

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. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:
   a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
   b. In the **Region** field, type the region for the Redshift server cluster.
7. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
9. In the **DbGroups Filter** field, type the DbGroup filter you want to use.
10. Optionally, in the **Endpoint URL** field, type the endpoint used to retrieve the Redshift cluster's credentials.
11. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
12. Optionally, in the **VPC Endpoint URL field**, type the endpoint used to communicate with the Redshift cluster.
13. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
14. In the **Azure Client ID** field, type the client ID associated with your Redshift application on Azure AD.
15. In the **Preferred Role** field, type the name or ID for the IAM role you want the user to assume when logged into Redshift.
16. In the **IdP Tenant** field, type the Azure AD tenant ID associated with your application.
17. In the **Timeout (sec)** field, type the amount of time, in seconds, that the connector waits for the auth flow to complete.
18. To save your settings and close the dialog box, click **OK**.

Using a Browser Plugin for Azure AD OAuth2

You can configure the connector to use a browser plugin to authenticate your connection through the Azure AD website.

To configure IAM authentication using a browser plugin for Azure AD OAuth2 in 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: Browser Azure AD OAuth2**.
3. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
   a. In the **AccessKeyID** field, type your Redshift access key ID.
   b. In the **SecretAccessKey** field, type your Redshift secret key.
4. In the **Azure Client ID** field, type the client ID associated with your Redshift application on Azure AD.
5. In the **Scope** field, type a space-separated list of scopes to which the user can consent.
6. In the **IdP Tenant** field, type the Azure AD tenant ID associated with your application.
7. In the **Timeout (sec)** field, type the amount of time, in seconds, that the connector waits for the auth flow to complete.
8. To save your settings and close the dialog box, click **OK**.

Using a Browser Plugin for a SAML Service

You can configure the connector to use a browser plugin to authenticate your connection through a SAML service such as Okta, Ping, or AD FS.

To configure IAM authentication using a browser plugin in 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: Browser SAML**.
3. In the **User** field, type the user name associated with your Redshift application on the identity provider.
4. In the **Password** field, type the password associated with your Redshift application on the identity provider.
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. If the ID and region of the Redshift server cluster are not already provided through the Server field, then do the following:

a. In the **Cluster ID** field, type the ID for the Redshift server cluster.
b. In the **Region** field, type the region for the Redshift server cluster.

7. In the **DbUser** field, type the ID that you want to designate to the Redshift user.
8. If the ID you entered in the DbUser field does not already exist in your Redshift account, you must create it:
   a. Select the **User AutoCreate** check box.
   b. In the **DbGroups** field, type the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all DbGroups that are received from the identity provider, select the **Force Lowercase** check box.
9. In the **DbGroups Filter** field, type the DbGroup filter you want to use.
10. Optionally, in the **Endpoint URL** field, type the endpoint to retrieve the Redshift cluster's credentials.
11. Optionally, in the **STS Endpoint URL** field, type the endpoint used to communicate with the AWS Security Token Service (AWS STS).
12. Optionally, in the **VPC Endpoint URL** field, type the endpoint used to communicate with the Redshift cluster.
13. Optionally, in the **AuthProfile** field, type the authentication profile you want to use to manage the connection settings, then do the following:
    a. In the **AccessKeyID** field, type your Redshift access key ID.
    b. In the **SecretAccessKey** field, type your Redshift secret key.
14. In the **Login URL** field, type the URL for the resource on the identity provider's website.
15. In the **Listen Port** field, type the number of the port that the connector uses to receive the SAML response from the identity provider.
16. In the **Preferred Role** field, type the name or ID for the IAM role you want the user to assume when logged into Redshift.
17. In the **Timeout (sec)** field, type the amount of time, in seconds, that the connector waits for the auth flow to complete.
18. 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, Azure AD, and Okta, the Windows version of the Amazon Redshift ODBC Connector also provides support for other credentials services. The connector can authenticate connections using any SAML-based credential provider plugin of your choice.

To configure an external credentials service in 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 connector to use this profile. The connector detects and uses the authentication settings specified in the profile.

Configuring Data Type Options in Windows

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

To configure data type options in 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 connector to return data as Unicode character types, select the **Use Unicode** check box.

   Note: When the **Use Unicode** check box is selected, the connector 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 connector to return Boolean columns as SQL_VARCHAR instead of SQL_BIT, select the **Show Boolean Column As String** check box.
4. To configure the connector to return Text columns as SQL_LONGVARCHAR instead of SQL_VARCHAR, select the **Text as LongVarChar** check box.
5. In the **Max Varchar** field, type the maximum data length for VarChar columns.
6. In the **Max LongVarChar** field, type the maximum data length for LongVarChar columns.
7. To save your settings and close the Data Type Configuration dialog box, click **OK**.

Configuring Additional Options in Windows

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

To configure additional options in 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 connector 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 connector to have more than one query, separated by a semicolon (`;`), in a single SQLExecDirect call, select **Use Multiple Statements**.
   - 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 connector to return SQL_ERROR immediately for any other queries that is executed if there is already an active query in execution under the same connection, select the **Enforce Single Statement** check box.
4. To configure the connector to recognize table type information from the data source, select the **Enable Table Types** check box. For more information, see Enable Table Types.
5. To configure the connector to enable read-only mode, select the **Enable Read Only** check box. For more information, see Enable Read Only.
6. To configure the connector to read metadata from multiple data stores, clear the **Database Metadata Current Database Only** check box. For more information, see Database Metadata Current Database Only.
7. 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.

8. To configure the connector 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.

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

Configuring TCP Keepalives in Windows

By default, the Amazon Redshift ODBC Connector is configured to use TCP keepalives to prevent connections from timing out. Settings such as how frequently the connector 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 in Windows:

1. On the Start screen, type **regedit**, and then click the **regedit** search result.
2. Select the appropriate registry key for the bitness of your connector:
   - If you are using the 32-bit connector 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 connector 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 connector sends a TCP keepalive packet.

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

   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. On the Start screen, type **regedit**, and then click the **regedit** search result.
2. Select the appropriate registry key for the bitness of your connector:
   - If you are using the 32-bit connector 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 in Windows

To help troubleshoot issues, you can enable logging. In addition to functionality provided in the Amazon Redshift ODBC Connector, 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.

Configuring Connector-wide Logging Options

The settings for logging apply to every connection that uses the Amazon Redshift ODBC Connector, so make sure to disable the feature after you are done using it. To configure logging for the current connection, see Configuring Logging for the Current Connection.

To enable connector-wide logging in 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:

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

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 Connector produces the following log files at the location you specify in the Log Path field:

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

If you enable the `UseLogPrefix` connection property, the connector 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.

To disable connector logging in 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.

Configuring Logging for the Current Connection

You can configure logging for the current connection by setting the logging configuration properties in the DSN or in a connection string. For information about the logging configuration properties, see Configuring Logging Options in Windows. Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over connector-wide settings.

Note: If the LogLevel configuration property is passed in via the connection string or DSN, the rest of the logging configurations are read from the connection string or DSN and not from the existing connector-wide logging configuration.

To configure logging properties in the DSN, you must modify the Windows registry. For information about the Windows registry, see the Microsoft Windows documentation.

Important: Editing the Windows Registry incorrectly can potentially cause serious, system-wide problems that may require re-installing Windows to correct.

To add logging configurations to a DSN in Windows:

1. On the Start screen, type **regedit**, and then click the **regedit** search result.
2. Navigate to the appropriate registry key for the bitness of your connector and your machine:
   - 32-bit System DSNs: **HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ODBC\ODBC.INI*[DSN Name]***
   - 64-bit System DSNs: **HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*[DSN Name]***
   - 32-bit and 64-bit User DSNs: **HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI*[DSN Name]***
3. For each configuration option that you want to configure for the current connection, create a value by doing the following:
   a. If the key name value does not already exist, create it. Right-click the *[DSN Name]* and then select **New > String Value**, type the key name of the configuration option, and then press **Enter**.
   b. Right-click the key name and then click **Modify**.

      To confirm the key names for each configuration option, see Connector Configuration Options.
   c. In the Edit String dialog box, in the **Value Data** field, type the value for the configuration option.
4. Close the Registry Editor.
5. Restart your ODBC application to make sure that the new settings take effect.

Verifying the Connector Version Number in Windows

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

To verify the connector version number in 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 Connector in the list of ODBC Connectors that are installed on your system. The version number is displayed in the **Version** column.

macOS Connector

This section provides an overview of the Connector in the mac OS platform, outlining the required system specifications and the steps for installing and configuring the connector in mac OS environments.

macOS System Requirements

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

- 215MB of available disk space
- One of the following ODBC driver managers installed:
  - iODBC 3.52.9 or later
  - unixODBC 2.2.14 or later

Installing the Connector in macOS

The Amazon Redshift ODBC Connector is available for macOS as a `.dmg` file named `AmazonRedshiftODBC-[Version].dmg`. The connector only supports 64-bit client applications.

To install the Amazon Redshift ODBC Connector in 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 connector files are installed in the `/opt/amazon/redshift` 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 connector. For more information, see Configuring the ODBC Driver Manager in Non-Windows Machines.

Verifying the Connector Version Number in macOS

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

To verify the connector version number in macOS:

- At the Terminal, run the command:

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

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

M1 Connector

This section provides an overview of the Connector, outlining the required system specifications and the steps for installing and configuring the connector.

M1 System Requirements

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

- 107 MB of available disk space
- One of the following ODBC driver managers installed:
  - iODBC 3.52.15 or later
  - unixODBC 2.3.9 or later

Installing the Connector on M1

The Amazon Redshift ODBC Connector is available for M1 as a `.dmg` file named `AmazonRedshiftODBC-[Version].dmg`. The connector only supports 64-bit client applications.

To install the Amazon Redshift ODBC Connector on M1:

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 connector files are installed in the `/opt/amazon/redshift` 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 connector. For more information, see Configuring the ODBC Driver Manager in Non-Windows Machines.

Verifying the Connector Version Number on M1

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

To verify the connector version number on M1:

- At the Terminal, run the following command:

  pkgutil --info com.amazon.redshiftodbc

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

Linux Connector

This section provides an overview of the Connector in the Linux platform, outlining the required system specifications and the steps for installing and configuring the connector in Linux environments.

Linux System Requirements

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

- 150MB of available disk space
- One of the following ODBC driver managers installed:
  - iODBC 3.52.9 or later
  - unixODBC 2.2.14 or later
- glibc 2.28 or later

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

Installing the Connector 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 connectors, and 32-bit applications must use 32-bit connectors. Make sure that you use a connector whose bitness matches the bitness of the client application:

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

The placeholders in the file names are defined as follows:

- *[Version]* is the version number of the connector.
- *[Release]* is the release number for this version of the connector.

To install the Amazon Redshift ODBC Connector using the RPM File:

1. Log in as the root user.
2. Navigate to the folder containing the RPM package for the connector.
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:

```
yum --nogpgcheck localinstall [RPMFileName]
```

- Or, if you are using SUSE Linux Enterprise Server, run the following command:

```
zypper install [RPMFileName]
```

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

Verifying the Connector Version Number in Linux

If you need to verify the version of the Amazon Redshift ODBC Connector that is installed on your Linux machine, you can query the version number through the command-line interface if the connector was installed using an RPM file. Alternatively, you can search the connector's binary file for version number information.

To verify the connector version number in Linux using the command-line interface:

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

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

To verify the connector version number in Linux using the binary file:

1. Navigate to the `/lib` subfolder in your connector installation directory. By default, the path to this directory is: `/opt/amazon/redshiftodbc/lib`.
2. Open the connector's `.so` binary file in a text editor, and search for the text `$driver_version_sb$:` The connector's version number is listed after this text.

Configuring the ODBC Driver Manager in Non-Windows Machines

To make sure that the ODBC Driver manager on your machine is configured to work with the Amazon Redshift ODBC Connector, 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 in Non-Windows Machines
- If the connector 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 Connector Configuration Files

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

Specifying ODBC Driver Managers in Non-Windows Machines

You need to make sure that your machine uses the correct ODBC Driver manager to load the connector. 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.

Specifying the Locations of the Connector 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 connector 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 accquired the connector 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 accquired the connector 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:

export ODBCINI=/usr/local/odbc/odbc.ini

export ODBCINSTINI=/usr/local/odbc/odbcinst.ini

export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini

For unixODBC:

export ODBCINI=/usr/local/odbc/odbc.ini

export ODBCSYSINI=/usr/local/odbc

export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini

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

1. If the AMAZONREDSHIFTODBCINI environment variable is defined, then the connector searches for the file specified by the environment variable.
2. The connector searches the directory that contains the connector library files for a file named `amazon.redshiftodbc.ini`.
3. The connector searches the current working directory of the application for a file named `amazon.redshiftodbc.ini`.
4. The connector searches the home directory for a hidden file named `.amazon.redshiftodbc.ini` (prefixed with a period).
5. The connector searches the `/etc` directory for a file named `amazon.redshiftodbc.ini`.

Configuring ODBC Connections in Non-Windows Machine

The following sections describe how to configure ODBC connections when using the Amazon Redshift ODBC Connector on non-Windows platforms:

- Creating a Data Source Name on a Non-Windows Machine
- Configuring a DSN-less Connection on a Non-Windows Machine
- Configuring Authentication on a Non-Windows Machine
- Configuring SSL Verification on a Non-Windows Machine
- Configuring Query Processing Modes on a Non-Windows Machine
- Configuring a Proxy Connection on a Non-Windows Machine
- Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine
- Configuring TCP Keepalives on a Non-Windows Machine
- Configuring Logging Options
- Testing the Connection

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 more information about configuring a DSN-less connection instead, see Configuring a DSN-less Connection on a Non-Windows Machine.

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 connector 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 connector.

   For example, on a macOS machine:

   [ODBC Data Sources]

Sample DSN=Amazon Redshift ODBC Driver

As another example, for a 32-bit connector 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 connector library file that matches the bitness of the application.

      For example, on a macOS machine:

      Driver=/opt/amazon/redshift/lib/libamazonredshiftodbc.dylib

      As another example, for a 32-bit connector 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.cbacbacba.com,

      Port=5439

      > **Note:**
      > If you are using IAM authentication and you specify the ClusterID and AWSRegion attributes, you do not need to specify the Server attribute.
   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.
   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.
   f. Optionally, modify how the connector runs queries and retrieves results into memory. For more information, see Configuring Query Processing Modes on a Non-Windows Machine.
   g. Optionally, configure the connector to connect through a proxy server. For more information, see Configuring a Proxy Connection on a Non-Windows Machine.
   h. Optionally, configure the connector 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.
   i. Optionally, modify the TCP keepalive settings that the connector uses to prevent connections from timing out. For more information, see Configuring TCP Keepalives on a Non-Windows Machine.

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 Connector, see Connector Configuration Properties.

4. Save the `odbc.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 ODBCINI environment variable specifies the location. For more information, see Specifying the Locations of the Connector Configuration Files.

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

[Sample DSN]

Driver=/opt/amazon/redshift/lib/libamazonredshiftodbc.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 connector on a Linux machine, containing a DSN that connects to Redshift:

[ODBC Data Sources]

Sample DSN=Amazon Redshift (x86)

[Sample DSN]

Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.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 connector 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 connector installation directory to the home directory, and then update the file as described below.

To define a connector 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 connector, 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 connector (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 connector library file that matches the bitness of the application.

      For example, on a macOS machine:

      Driver=/opt/amazon/redshift/lib/libamazonredshiftodbc.dylib

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

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

   b. Optionally, set the `Description` property to a description of the connector.

      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 Connector Configuration Files.

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

[ODBC Drivers]

Amazon Redshift ODBC Driver=Installed

Description= Amazon Redshift ODBC Driver

Driver=/opt/amazon/redshift/lib/libamazonredshiftodbc.dylib

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

[ODBC Drivers]

Amazon Redshift ODBC Driver 32-bit=Installed

Amazon Redshift 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 connector 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.

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

- Configuring Authentication on a Non-Windows Machine
- Configuring SSL Verification on a Non-Windows Machine
- Configuring a Proxy Connection on a Non-Windows Machine
- Configuring an HTTPS Proxy for IAM Authentication on a Non-Windows Machine
- Configuring Query Processing Modes on a Non-Windows Machine
- Configuring TCP Keepalives on a Non-Windows Machine

For detailed information about all the connection properties that the connector supports, see Connector Configuration Properties.

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 connector to connect to an SSL-enabled socket. When connecting to a server over SSL,

the connector 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.

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 connector 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 connector supports the following authentication methods:

- Standard authentication using your database user name and password (see Using Standard Authentication)
- IAM authentication using a profile (see Using an IAM Profile)
- IAM authentication using IAM credentials (see Using IAM Credentials)
- IAM authentication using Active Directory Federation Services (AD FS) (see Using Active Directory Federation Services (AD FS))
- IAM authentication using Azure AD service (see Using Azure AD Service)
- IAM authentication using a JSON Web Token (JWT) (see Using a JSON Web Token (JWT))
- IAM authentication using Okta service (see Using Okta Service)
- IAM authentication using PingFederate service (see Using PingFederate Service)
- IAM authentication using a browser plugin for Azure AD (see Using a Browser Plugin for Azure AD)

- IAM authentication using a browser plugin for Azure AD OAuth2 (see Using a Browser Plugin for Azure AD OAuth2)
- IAM authentication using a browser plugin for a SAML service (see Using a Browser Plugin for a SAML Service)

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 connector 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 connector 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. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
5. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
6. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:

a. Set the `AutoCreate` property to `1`.

b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.

c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.

7. Optionally, set the `EndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
8. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
9. Optionally, set the `VpcEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
10. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
11. Optionally, set the `group_federation` property to `1` to enable group federation.
12. 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`.

> **Note:**
> If both properties are set, `InstanceProfile` takes precedence and the connector uses the Amazon EC2 instance profile.

Using IAM Credentials

You can configure the connector 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. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
    a. Set the `ClusterID` property to the ID for the Redshift server cluster.
    b. Set the `Region` property to the region for the Redshift server cluster.
3. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
4. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:

a. Set the `AutoCreate` property to `1`.
b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.

5. Optionally, set the `EndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
6. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
7. Optionally, set the `VpcEndpointUrl` property to the endpoint used to communicate with the Redshift cluster
8. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings.
9. Set the `AccessKeyID` property to your Redshift access key ID.
10. Set the `SecretAccessKey` property to your Redshift secret key.
11. If you are using an IAM role, set the `SessionToken` property to your temporary session token.
12. Optionally, set the `group_federation` property to `1` to enable group federation.

Using Active Directory Federation Services (AD FS)

You can configure the connector 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:
     - Set the `UID` property to the user name associated with your AD FS account.
     - 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. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
5. Set the `DbUser` property to the ID that you want to designate to the Redshift user.

6. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to `1`.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.
7. Optionally, set the `EndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
8. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
9. Optionally, set the `VpcEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
10. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
11. Set the `IdP_Host` property to the address of the service host.
12. Set the `IdP_Port` property to the port number that the service listens at.
13. 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.
14. Optionally, set the `loginToRp` property to the the relying party trust you want to use.
15. To skip verification of the SSL certificate of the IDP server, set the `SSL_Insecure` property to `1`.

Using Azure AD Service

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

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

1. Set the `UID` property to the user name associated with your Redshift application on Azure AD.
2. Set the `PWD` property to the password associated with your Redshift application on Azure AD.
3. Set the `IAM` property to `1`.
4. Set the `plugin_name` property to `azuread`.
5. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
6. Set the `DbUser` property to the ID that you want to designate to the Redshift user.

7. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to `1`.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.
8. Set the `dbgroups_filter` property to the the DbGroup filter you want to use.
9. Optionally, set the `EndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
10. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
11. Optionally, set the `VpcEndpointUrl` property to the endpoint used to communicate with the Redshift cluster
12. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
13. Set the `IdP_Tenant` property to the Azure AD tenant ID associated with your application.
14. Set the `Client_ID` property to the client ID associated with your Redshift application on Azure AD.
15. Set the `Client_Secret` property to the client secret associated with your Redshift application on Azure AD.
16. Set the `Preferred_Role` property to the the name or ID for the IAM role you want the user to assume when logged into Redshift.

Using a JSON Web Token (JWT)

You can configure the connector to authenticate your connection by using a token generated by Microsoft Azure for authentication.

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

1. Set the `IAM` property to `1`.
2. Set the `plugin_name` property to `Jwt`.
3. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
   a. Set the `AccessKeyID` property to your Redshift access key ID.
   b. Set the `SecretAccessKey` property to your Redshift secret key.
4. Set the `web_identity_token` property to the token that is provided by the identity provider.

5. Optionally, set the `provider_name` property to the name of the authentication provider created from the CREATE IDENTITY PROVIDER query.

Using a JSON Web Token (JWT) from other services

You can configure the connector to authenticate your connection by using a token generated by any other service for authentication.

IAM authentication using a JWT IAM Auth Plugin on a non-Windows machine:

1. Set the `IAM` property to `1`.
2. Set the `plugin_name` property to `JwtIamAuthPlugin`.
3. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
4. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
5. If the ID you specified for the DbUser property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to 1.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.
6. Set the `web_identity_token` property to the token that is provided by the identity provider.
7. Set the `role_arn` property to the Amazon Resource Name (ARN) of the role.
8. Optionally, set the `EndpointUrl` property to the endpoint used to retrieve the Redshift cluster's credentials.
9. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
10. Optionally, set the `VPCEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
11. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
12. Optionally, set the `group_federation` property to 1 to enable group federation.
13. Optionally, set the `role_session_name` property to the name of the assumed role session.
14. Optionally, set the `Duration` property to the duration of the role session, in seconds.

Using Okta Service

You can configure the connector 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. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
6. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
7. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to `1`.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.
8. Optionally, set the `EndpointUrl` property to the endpoint used to retrieve the Redshift cluster's credentials.
9. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
10. Optionally, set the `VPCEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
11. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
12. Set the `IdP_Host` property to the address of the service host.
13. 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.
14. Set the `App_ID` property to the Okta-supplied ID associated with your Redshift application.
15. Optionally, set the `App_Name` property to the name of your Okta application.

Using PingFederate Service

You can configure the connector 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. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
6. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
7. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to `1`.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.
8. Optionally, set the `EndpointUrl` property to the endpoint used to retrieve the Redshift cluster's credentials.
9. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
10. Optionally, set the `VPCEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
11. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
12. Set the `IdP_Host` property to the address of the service host.
13. Set the `IdP_Port` property to the port number that the service listens at.
14. 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.
15. To skip verification of the SSL certificate of the IDP server, set the `SSL_Insecure` property to `1`.
16. Optionally, set the `partner_spid` property to a partner SPID (service provider ID) value.

Using a Browser Plugin for Azure AD

You can configure the connector to use a browser plugin to authenticate your connection through the Azure AD website.

To configure IAM authentication using a browser plugin for Azure on a non-Windows machine:

1. Set the `UID` property to the user name associated with your Redshift application on Azure AD.
2. Set the `PWD` property to the password associated with your Redshift application on Azure AD.
3. Set the `IAM` property to `1`.
4. Set the `plugin_name` property to `BrowserAzureAD`.
5. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
6. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
7. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to `1`.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.
   c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.
8. Set the `dbgroups_filter` property to the the DbGroup filter you want to use.
9. Optionally, set the `EndpointUrl` property to the endpoint used to retrieve the Redshift cluster's credentials.
10. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
11. Optionally, set the `VPCEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
12. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
13. Set the `Client_ID` property to the client ID associated with your Redshift application on Azure AD.
14. Set the `Preferred_Role` property to the the name or ID for the IAM role you want the user to assume when logged into Redshift.
15. Set the `IdP_Tenant` property to the Azure AD tenant ID associated with your application.
16. Set the `idp_response_timeout` property to the amount of time, in seconds, that the Redshift driver waits for the auth flow to complete.

Using a Browser Plugin for Azure AD OAuth2

You can configure the connector to use a browser plugin to authenticate your connection through the Azure AD website.

To configure IAM authentication using a browser plugin for Azure AD OAuth2 on a non-Windows machine:

1. Set the `IAM` property to `1`.
2. Set the `plugin_name` property to `BrowserAzureADOAuth2`.
3. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
   a. Set the `AccessKeyID` property to your Redshift access key ID.
   b. Set the `SecretAccessKey` property to your Redshift secret key.
4. Set the `Client_ID` property to the client ID associated with your Redshift application on Azure AD.
5. Set the `scope` property to a space-separated list of scopes to which the user can consent.
6. Set the `IdP_Tenant` property to the Azure AD tenant ID associated with your application.
   a. Set the `idp_response_timeout` property to the amount of time, in seconds, that the Redshift driver waits for the auth flow to complete.

Using a Browser Plugin for a SAML Service

You can configure the connector to use a browser plugin to authenticate your connection through a SAML service such as Okta, Ping, or AD FS.

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

1. Set the `UID` property to the user name associated with your Redshift application on the identity provider.
2. Set the `PWD` property to the password associated with your Redshift application on the identity provider.
3. Set the `IAM` property to `1`.
4. Set the `plugin_name` property to `BrowserSAML`.
5. If the ID and region of the Redshift server cluster are not already provided through the `Server` property, then do the following:
   a. Set the `ClusterID` property to the ID for the Redshift server cluster.
   b. Set the `Region` property to the region for the Redshift server cluster.
6. Set the `DbUser` property to the ID that you want to designate to the Redshift user.
7. If the ID you specified for the `DbUser` property does not already exist in your Redshift account, you must create it:
   a. Set the `AutoCreate` property to `1`.
   b. Set the `DbGroups` property to the names of any user groups that you want the new DbUser to be added to, separated by commas.

c. Optionally, to lowercase all `DbGroups` that are received from the identity provider, select the `Force Lowercase` check box.

8. Set the `dbgroups_filter` property to the the DbGroup filter you want to use.
9. Optionally, set the `EndpointUrl` property to the endpoint used to retrieve the Redshift cluster's credentials.
10. Optionally, set the `StsEndpointUrl` property to the endpoint used to communicate with the AWS Security Token Service (AWS STS).
11. Optionally, set the `VPCEndpointUrl` property to the endpoint used to communicate with the Redshift cluster.
12. Optionally, set the `AuthProfile` property to the authentication profile you want to use to manage the connection settings, then do the following:
    a. Set the `AccessKeyID` property to your Redshift access key ID.
    b. Set the `SecretAccessKey` property to your Redshift secret key.
13. Set the `Login_URL` property to the URL for the resource on the identity provider's website.
14. Set the `Listen_Port` property to the number of the port that the connector uses to receive the SAML response from the identity provider.
15. Set the `Preferred_Role` property to the the name or ID for the IAM role you want the user to assume when logged into Redshift.
16. Set the `idp_response_timeout` property to the amount of time, in seconds, that the Redshift driver waits for the auth flow to complete.

Configuring Query Processing Modes on a Non-Windows Machine

To optimize connector performance, you can modify how the connector runs queries and retrieves results into memory. For example, you can configure the connector 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 connector 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 connector 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 connector returns at a time.

Enabling Retrieve Entire Result Mode

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

To enable Retrieve Entire Result Mode:

- Make sure that the `SingleRowMode`, `UseDeclareFetch`, and `UseMultipleStatements` properties are set to `0` or not set.

Enabling Multiple Statements Mode

The connector can have more than one query, separated by a semicolon (`;`), in a single SQLExecDirect call. The connector 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 connector to return SQL_ERROR immediately for any other queries that is executed if there is already an active query in execution under the same 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 connector to connect to Redshift through a proxy server, so that communications between the connector and your Redshift data source are passed through the proxy server.

Note:
You can also configure the connector 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.

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 connector to pass IAM authentication processes through a proxy server.

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

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_Use_Https_Proxy` property to `1`.

Configuring TCP Keepalives on a Non-Windows Machine

By default, the Amazon Redshift ODBC Connector is configured to use TCP keepalives to prevent connections from timing out. Settings such as how frequently the connector 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 connector 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 connector 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 connector.

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

You can set the connection properties described below in a connection string, in a DSN (in the `odbc.ini` file), or as a connector-wide setting (in the `amazon.redshiftodbc.ini` file). Settings in the connection string take precedence over settings in the DSN, and settings in the DSN take precedence over connector-wide settings.

To enable logging:

1. To specify the level of information to include in log files, set the `LogLevel` property to one of the following numbers:

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

2. Set the `LogPath` key to the full path to the folder where you want to save log files.
3. Set the `LogFileCount` key to the maximum number of log files to keep.

   Note: After the maximum number of log files is reached, each time an additional file is created, the connector deletes the oldest log file.

4. Set the `LogFileSize` key to the maximum size of each log file in bytes.

   Note: After the maximum file size is reached, the connector creates a new file and continues logging.

5. Optionally, to prefix the log file name with the user name and process ID associated with the connection, set the `UseLogPrefix` property to `1`.
6. Save the `amazon.redshiftodbc.ini` configuration file.
7. Restart your ODBC application to make sure that the new settings take effect.

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

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

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 connector. Use iodbctest to test how your connector works with an ANSI application, or use iodbctestw to test how your connector 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.

   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 connector and your DSN. isql and iusql can only be used to test connections that use a DSN. Use isql to test how your connector works with an ANSI application, or use iusql to test how your connector 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 Connector Configuration Properties.

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 connector 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 connector does not encrypt your credentials.

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

- *[AzureClientID]* is the client ID associated with your Redshift application in Azure AD.
- *[AzureClientSecret]* is the secret key associated with your Redshift application in Azure AD.
- *[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 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).
- *[IDP_Tenant]* is the Azure AD tenant ID associated with your Redshift application.
- *[OktaAppID]* is the app ID assocaited 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.
- *[WebIdentityToken]* is the token that is provided by the identity provider.
- *[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]*;
UID=*[YourUserName]*;PWD=*[YourPassword]*;ProxyHost=*[PServer]*;
ProxyPort=*[PPort]*;

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;

Important:

- This example assumes that the profile contains a user name, password, and user ID. If this information is missing from the profile, then you must provide it by specifying the `UID`, `PWD`, and `DbUser` properties (respectively) in the connection string.
- If the user ID specified in your profile or connection string does not already exist, then you must configure the connector 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 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=wJalrXUtnFEMI/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 connector 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;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=adfs;UID=jsmith;PWD=amazon12345;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 connector 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 Azure AD Portal

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

Driver=Amazon Redshift ODBC Driver;Server=*[Server]*;
Port=*[PortNumber]*;Database=*[DatabaseName]*;IAM=1;
plugin_name=AzureAD;UID=*[YourUserName]*;PWD=*[YourPassword]*;DbUser=*[UserID]*;IdP_Tenant=
[IDP_Tenant];Client_ID=*[AzureClientID]*;Client_Secret=*[AzureClientSecret]*;

For example:

Driver=Amazon Redshift ODBC Driver;
Server=testserver.abcabcabcabc.us-west-
2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;
plugin_name=AzureAD;UID=jsmith;PWD=amazon12345;DbUser=Amazon;IdP_Tenant=e12x4am2-
7571-23pl-ete9-4na018221n09;Client_ID=cl007ent-66i6-4de9-1x2a-mp02le2021ss;Client_
Secret=example.E1-wC7Hiy2AwE2XAM:ple;

Important:

- If the specified user ID does not already exist, then you must configure the connector 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 a JSON Web Token (JWT)

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

Driver=Amazon Redshift ODBC Driver;Server=*[Server]*;
Port=*[PortNumber]*;Database=*[DatabaseName]*;IAM=1;plugin_name=jwt;web_identity_token=
[WebIdentityToken];

For example:

Driver=Amazon Redshift ODBC Driver;Server=testserver.abcabcabcabc.us-
west2.redshift.amazonaws.com;Port=5439;Database=TestDB;IAM=1;plugin_name=jwt;web_identity_
token=eyJhbGciOiJSUzI1NiIsImt;

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]*;

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=mQkRaOqFRNy5hAc262lW;

Important:

- If the specified user ID does not already exist, then you must configure the connector 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 connector 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 connector 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 connector 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.

Features

For more information on the features of the Amazon Redshift ODBC Connector, see the following:

- Query Processing Modes
- TCP Keepalives
- Data Types
- Security and Authentication

Query Processing Modes

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

The following query processing modes are available:

- **Single Row Mode**: The connector returns query results one row at a time.
- **Declare/Fetch Mode**: The connector returns a user-specified number of rows at a time.
- **Retrieve Entire Result Mode**: The connector returns the entire query result into memory.
- **Multiple Statements Mode:** The connector can have more than one query, separated by a semicolon (`;`), in a single SQLExecDirect call. The application calls SQLMoreResults to move to the next result set. When using this mode, the connector returns all the query results into memory.
- **Enforce Single Statement Mode**: The connector allows applications to allocate more than one statement handle and execute queries in each statement handle concurrently per connection. However, the connector allows only one active statement at a time for each connection. When using this mode, the connector returns SQL_ERROR immediately for any other queries that is executed if there is already an active query in execution under the same connection. You can use this mode in conjunction with the Single Row, Declare/Fetch, and Retrieve Entire Result modes. For more information, see Enforce Single Statement.

By default, the connector does not allow more than one active query at a time, and returns the entire query result into memory. When there is an active query in execution, the connector blocks queries in other statement handles from execution until the active query finishes execution and retrieves all the data, or when the application calls SQLCloseCursor or SQLFreeHandle with a HandleType of SQL_Handle_STMT to indicate that the statement handle can be freed.

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 connector processes queries, see Configuring Additional Options in Windows if you are using the Windows version of the connector, or see Configuring Query Processing Modes on a Non-Windows Machine if you are using a non-Windows version of the connector.

TCP Keepalives

By default, the Amazon Redshift ODBC Connector 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 connector automatically sends keepalive packets to the server. If the server does not respond, then the connector returns an indication that the connection is broken.

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

Data Types

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

The table below lists the supported data type mappings.

Note:
If the Use Unicode option (the `UseUnicode` key) is enabled, then the connector returns SQL_WCHAR instead of SQL_CHAR, and SQL_WVARCHAR instead of SQL_VARCHAR.

Redshift Type	SQL Type
BIGINT	SQL_BIGINT
BOOLEAN	SQL_VARCHAR If the Show Boolean Column As String option (the `BoolsAsChar` key) is disabled, then SQL_BIT is returned instead.
CHAR	SQL_CHAR ■ If the length of the column is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_LONGVARCHAR is returned instead. ■ If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WCHAR is returned instead. ■ 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.
DATE	SQL_TYPE_DATE If you are using ODBC 2.0, the SQL type is

Redshift Type	SQL Type
	SQL_DATE.
DECIMAL	SQL_NUMERIC
DOUBLE PRECISION	SQL_DOUBLE
GEOGRAPHY	SQL_LONGVARBINARY
GEOMETRY	SQL_LONGVARBINARY
INTEGER	SQL_INTEGER
REAL	SQL_REAL
SMALLINT	SQL_SMALLINT
SUPER	SQL_LONGVARCHAR If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WLONGVARCHAR is returned instead.
TEXT	SQL_LONGVARCHAR ■ If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WLONGVARCHAR is returned instead. ■ If the Text As LongVarChar option (the `TextAsLongVarchar` key) is disabled, then SQL_VARCHAR is returned instead. ■ If Use Unicode is enabled and Text As LongVarChar is disabled at the same time, then SQL_WVARCHAR is returned instead.
TIME	SQL_TYPE_TIME If you are using ODBC 2.0, the SQL type is SQL_TIME.
TIMETZ	SQL_TYPE_TIME If you are using ODBC 2.0, the SQL type is SQL_TIME.
TIMESTAMP	SQL_TYPE_TIMESTAMP If you are using ODBC 2.0, the SQL type is SQL_TIMESTAMP.
TIMESTAMPTZ	SQL_TYPE_TIMESTAMP If you are using ODBC 2.0, the SQL type is SQL_TIMESTAMP.

Redshift Type	SQL Type
VARBYTE	SQL_LONGVARBINARY
VARCHAR	SQL_VARCHAR ■ If the length of the column is greater than the Max Varchar (`MaxVarchar`) setting, then SQL_LONGVARCHAR is returned instead. ■ If the Use Unicode option (the `UseUnicode` key) is enabled, then SQL_WVARCHAR is returned instead. ■ 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.

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 Connector 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 connector 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 connector and the server.

The connector supports authenticating your connection using your Redshift user name and password, or using IAM authentication. For detailed configuration instructions, see Configuring Authentication in Windows or Configuring Authentication on a Non-Windows Machine.

Additionally, the connector supports SSL connections with or without one-way authentication. If the server has an SSL-enabled socket, then you can configure the connector 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 in Windows or Configuring SSL Verification on a Non-Windows Machine.

Connector Configuration Properties

Connector Configuration Options lists the configuration options available in the Amazon Redshift ODBC Connector alphabetically by field or button label. Options having only key names, that is, not appearing in the user interface of the connector, 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 Connector, or via the key name when using a connection string or configuring a connection from a Linux or macOS computer:

- AccessKeyID
- Allow Self-Signed Server Certificate
- Auth Type
- AuthProfile
- Authentication Mode
- Azure Client ID
- Azure Client Secret
- Cache Size
- CheckCertificate Revocation
- Cluster ID
- Max LongVarChar
- Max Varchar
- Minimum TLS
- Okta App ID
- Okta App Name
- Partner SPID
- Password
- Preferred Role
- Port
- Profile Name

- Custom SSL Certificate Path
- Database
- Database Metadata Current Database Only
- DbGroups
- DbGroups Filter
- DbUser
- Enable HTTPS Proxy For Federated Access
- Enable Proxy For Amazon Redshift Connection
- Enable Read Only
- Enable Table Types
- EncryptPassword
- Endpoint URL
- Enforce Single Statement
- Force Lowercase
- Group Federation
- HTTPS Proxy Password
- HTTPS Proxy Port
- HTTPS Proxy Server
- HTTPS Proxy Username
- IdP Host
- IdP Port
- IdP Tenant
- Listen Port
- Provider Name
- Proxy Password
- Proxy Port
- Proxy Server
- Proxy Username
- Region
- Retrieve Entire Result Into Memory
- Scope
- Server
- SessionToken
- SecretAccessKey
- Show Boolean Column As String
- Single Row Mode
- SSL Insecure
- StsEndpointUrl
- Text As LongVarChar
- Timeout (sec)
- Use Declare/Fetch
- Use HTTPS Proxy For Authentication On IdP
- Use Instance Profile
- Use Multiple Statements
- Use System Trust Store
- Use Unicode
- User

- Log Level
- Log Path
- Login URL
- loginToRp
- User AutoCreate
- Web Identity Token

AccessKeyID

The IAM access key for the user or role. If this is specified, then SecretAccessKey must also be specified.

Key Name	Default Value	Required
`AccessKeyID`	None	Yes, if using IAM credentials for authentication or AuthProfile.

Allow Self-Signed Server Certificate

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

- Enabled (`1`): The connector authenticates the Redshift server even if the server is using a self-signed certificate.
- Disabled (`0`): The connector 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.

Key Name	Default Value	Required
`AllowSelfSignedServer Cert`	Clear (`0`)	No

AuthProfile

This option specifies the authentication profile used to manage the connection settings.

Note:
If this property is used, AccessKeyID and SecretAccessKey are required.

Key Name	Default Value	Required
`AuthProfile`	None	No

Auth Type

This option specifies the authentication mode that the connector uses when you configure a DSN using the Amazon Redshift ODBC Connector 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: Azure AD**: IAM authentication using Azure AD portal.
- **Identity Provider: Browser Azure AD**: IAM authentication using a browser plugin for Azure AD portal.
- **Identity Provider: Browser Azure AD OAuth2**: IAM authentication using a browser plugin for Azure AD OAuth2 portal.
- **Identity Provider: JWT**: IAM authentication using a JSON Web Token (JWT) generated by Microsoft Azure for authentication.
- **Identity Provider: JWT IAM Auth Plugin** :IAM authentication using a JSON Web Token (JWT) generated by any other service.
- **Identity Provider: Okta**: IAM authentication using Okta service.
- **Identity Provider: PingFederate**: IAM authentication using PingFederate 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 connector.

When you configure a connection using a connection string or a non-Windows machine, the connector 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.

Key Name	Default Value	Required
N/A	`Standard`	Yes, when you configure a DSN using the Amazon Redshift ODBC Connector DSN Setup dialog box.

Authentication Mode

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*: http://docs.aws.amazon.com/redshift/latest/mgmt/connecting-ssl-support.html#connect-using-ssl.

Key Name	Default Value	Required
SSLMode	verify-ca	No

Azure Client ID

The client ID associated with your Redshift application in Azure AD.

Key Name	Default Value	Required
Client_ID	None	Yes, if using Azure AD or Azure AD OAuth2 for authentication.

Azure Client Secret

The secret key associated with your Redshift application in Azure AD.

Key Name	Default Value	Required
Client_Secret	None	Yes, if using Azure AD for authentication.

Cache Size

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

Key Name	Default Value	Required
`Fetch`	`100`	Yes, if Declare/Fetch Mode is enabled.

CheckCertificate Revocation

This option specifies whether the connector 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).

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

Note: This option is disabled when the `AllowSelfSignedServerCert` property is set to `1`. This option is only available in Windows.

Key Name	Default Value	Required
`CheckCertRevocation`	Clear (`0`)	No

Cluster ID

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

Key Name	Default Value	Required
`ClusterID`	None	Yes, if using IAM authentication and the Cluster ID is not specified in the Server property.

Custom SSL Certificate Path

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

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

Key Name	Default Value	Required
`SSLCertPath`	The location of the connector	No

Key Name	Default Value	Required
	DLL file.	

Database

Note: To inspect your databases and determine the appropriate schema to use, at the Redshift command prompt, type `show databases`.

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

Key Name	Default Value	Required
`Database`	None	Yes

Database Metadata Current Database Only

This option specifies whether the connector returns metadata from multiple databases and clusters.

- Enabled (`1`): The connector only returns metadata from the current database.
- Disabled (`0`): The connector returns metadata across multiple Redshift databases and clusters.

Key Name	Default Value	Required
`DatabaseMetadata CurrentDbOnly`	Selected (`1`)	No

DbGroups

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

Key Name	Default Value	Required
`DbGroups`	None	No

DbGroups Filter

The regular expression you can specify to filter DbGroups that are received from the SAML response to Redshift when using Azure, Browser Azure, and Browser SAML authentication types.

Key Name	Default Value	Required
`dbgroups_filter`	None	No

DbUser

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).

Key Name	Default Value	Required
`DbUser`	None	No

Duration

The duration, in seconds, of the role session.

Key Name	Default Value	Required
`duration`	`0`	No

Enable HTTPS Proxy For Federated Access

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

This option specifies whether the connector passes the IAM authentication processes through a proxy server.

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

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

Key Name	Default Value	Required
N/A	Clear	Yes, if using the Additional Configuration dialog box to configure the connector to pass IAM authentication processes through a proxy.

Enable Proxy For Amazon Redshift Connection

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

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

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

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

Key Name	Default Value	Required
N/A	Clear	Yes, if using the Additional Configuration dialog box to configure a proxy connection.

Enable Read Only

This option controls whether the connector is in read-only mode.

- Enabled (`1`): The connection is in read-only mode, and cannot write to the data store.
- Disabled (`0`): The connection is not in read-only mode, and can write to the data store.

Key Name	Default Value	Required
`ReadOnly`	Clear (`0`)	No

Enable Table Types

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

- Enabled (`1`): The connector 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.

Key Name	Default Value	Required
`EnableTableTypes`	Clear (`0`)	No

Encrypt Password

This option specifies how the connector 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.

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

Key Name	Default Value	Required
N/A	All Users Of This Machine	No

Endpoint URL

This option specifies the overriding endpoint used to communicate with the Redshift cluster.

Key Name	Default Value	Required
`EndpointUrl`	None	No

Enforce Single Statement

This option specifies whether the connector returns SQL_ERROR immediately for any other queries that is executed if there is already an active query in execution under the same connection. The connector allows applications to allocate more than one statement handles and execute queries in each statement handle concurrently per connection. However, the connector allows only one active statement at a time for each connection.

- Enabled (`1`): The connector allows one active query to be executed at a time. If there is already an active query in execution under the same connection, the connector returns SQL_ERROR immediately for any other queries that is executed if there is already an active query in execution under the same connection.
- Disabled (`0`): The connector allows more than one queries to be executed at a time, but all queries are still sent and executed sequentially. If there is already an active query in execution under the same connection, the connector blocks queries in other statement handles from execution until the active query finishes execution and retrieves all the data, or when the application calls SQLCloseCursor or SQLFreeHandle with a HandleType of SQL_HANDLE_STMT to indicate that the statement handle can be freed.

Note:

- If Enforce Single Statement and Use Multiple Statements are both enabled, Use Multiple Statements Mode takes precedence.
- The connector only allows multiple queries to be execute sequentially when the statement handles are allocated in different threads. If there is already an active query in execution under the same connection and the queries to be executed belong to statement handles that are allocated within the same thread, the connector returns SQL_ERROR immediately. For more information, see Query Processing Modes.

Key Name	Default Value	Required
EnforceSingleStatement	Clear (0)	No

Force Lowercase

This option specifies whether the connector lowercases all DbGroups sent from the identity provider to Redshift when using SSO authentication.

- True: The connector lowercases all DbGroups that are sent from the identity provider.
- False: The connector does not alter DbGroups.

Key Name	Default Value	Required
ForceLowercase	False	No

Group Federation

This property specifies whether the connector uses group federation, when configured to use AWS IAM Credentials, AWS Profile or JWT IAM Auth Plugin for authentication.

- Enabled (1): The connector uses group federation, when configured to use AWS IAM Credentials or AWS Profile for authentication.The connector uses group federation, when configured to use AWS IAM Credentials, AWS Profile or JWT IAM Auth Plugin for authentication.
  - Disabled (0): The connector does not use group federation.

Key Name	Default Value	Required
group_federation	Clear (0)	No

HTTPS Proxy Password

The password that you use to access the proxy server.

Key Name	Default Value	Required
Https_Proxy_Password	None	Yes, if passing IAM authentication processes through a proxy server that requires authentication.

HTTPS Proxy Port

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

Key Name	Default Value	Required
Https_Proxy_Port	None	Yes, if passing IAM authentication processes through a proxy server.

HTTPS Proxy Server

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

Key Name	Default Value	Required
Https_Proxy_Host	None	Yes, if passing IAM authentication processes through a proxy server.

HTTPS Proxy Username

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

Key Name	Default Value	Required
Https_Proxy_Username	None	Yes, if passing IAM authentication processes through a proxy server that requires authentication.

IdP Host

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

Key Name	Default Value	Required
IdP_Host	None	Yes, if using a credentials service for authentication.

IdP Port

The port for an IdP (identity provider).

Key Name	Default Value	Required
IdP_Port	None	Yes, if using a credentials service for authentication.

IdP Tenant

The Azure AD tenant ID associated with your Redshift application.

Key Name	Default Value	Required
IdP_Tenant	None	Yes, if using Azure AD or Azure AD OAuth2 for authentication.

Listen Port

The port that the connector uses to receive the SAML response from the identity provider when using the SAML or Azure AD services through a browser plugin.

Key Name	Default Value	Required
Listen_Port	7890	No

Log Level

Use this property to enable or disable logging in the connector 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.
- When logging with connection strings and DSNs, this option only applies to per-connection logs.

Set the property to one of the following values:

- OFF (0): Disable all logging.
- FATAL (1): Logs severe error events that lead the connector to abort.
- ERROR (2): Logs error events that might allow the connector 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 connector.
- DEBUG (5): Logs detailed information that is useful for debugging the connector.
- TRACE (6): Logs all connector activity.

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

- A amazonredshiftodbcdriver.log file that logs connector 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 connector activity that is specific to the connection.

If you enable the `UseLogPrefix` connection property, the connector 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.

Key Name	Default Value	Required
`LogLevel`	OFF (`0`)	No

Log Path

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

Important: When logging with connection strings and DSNs, this option only applies to per-connection logs.

Key Name	Default Value	Required
`LogPath`	None	Yes, if logging is enabled.

Login URL

The URL for the resource on the identity provider's website when using the SAML or Azure AD services through a browser plugin.

Key Name	Default Value	Required
`Login_Url`	None	Yes, if authenticating with the SAML or Azure AD services through a browser plugin.

loginToRp

The relying party trust you want to use for the AD FS authentication type.

Key Name	Default Value	Required
`loginToRp`	`urn:amazon:webservices`	No

Max LongVarChar

The maximum data length for LongVarChar columns.

- If the column is of type WVARCHAR, the length is in Unicode characters.
- Otherwise, the length is in UTF-8 code units.

Key Name	Default Value	Required
MaxLongVarChar	8190	No

Max Varchar

The maximum data length for VARCHAR columns.

- If the column is of type WVARCHAR, the length is in Unicode characters.
- Otherwise, the length is in UTF-8 code units.

Key Name	Default Value	Required
MaxVarchar	255	No

Minimum TLS

The minimum version of TLS/SSL that the connector 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.

Key Name	Default Value	Required
Min_TLS	TLS 1.2 (1.2)	No

Okta App ID

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

Key Name	Default Value	Required
App_ID	None	Yes, if authenticating through the Okta service.

Okta App Name

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

Key Name	Default Value	Required
App_Name	None	No

Partner SPID

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

Key Name	Default Value	Required
partner_spid	None	No

Password

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

Important: This option should not be explicitly set in the DSN properties of the Windows Registry. in Windows, if both PWD and ENCRYPTED_PWD are set, the connector always uses the value for PWD.

Key Name	Default Value	Required
PWD OR Password	None	Yes, if User has been set.

Port

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

Key Name	Default Value	Required
Port	5439	Yes

Preferred Role

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

Key Name	Default Value	Required
Preferred_Role	None	No

Profile Name

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 connector 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.

Key Name	Default Value	Required
`Profile`	None	No

Provider Name

The authentication provider created by user using the CREATE IDENTITY PROVIDER query.

Key Name	Default Value	Required
`provider_name`	None	No

Proxy Password

The password that you use to access the proxy server.

Key Name	Default Value	Required
`ProxyPwd`	None	Yes, if connecting to a proxy server that requires authentication.

Proxy Port

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

Key Name	Default Value	Required
`ProxyPort`	None	Yes, if connecting through a proxy server.

Proxy Server

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

Key Name	Default Value	Required
`ProxyHost`	None	Yes, if connecting through a

Key Name	Default Value	Required
		proxy server.

Proxy Username

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

Key Name	Default Value	Required
`ProxyUid`	None	Yes, if connecting to a proxy server that requires authentication.

Region

The AWS region that your cluster is in.

Key Name	Default Value	Required
`Region`	None	Yes, if using IAM authentication and the region is not specified in the Server property.

Retrieve Entire Result Into Memory

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

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

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

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

Key Name	Default Value	Required
N/A	Selected (`1`)	No

Role ARN

The Amazon Resource Name (ARN) of the role.

Key Name	Default Value	Required
role_arn	None	Yes, if authenticating using JwtIamAuthPlugin authentication.

Role Session Name

The name of the assumed role session.

Key Name	Default Value	Required
role_session_name	jwt_redshift_session	No

Scope

A space-separated list of scopes to which the user can consent.

Key Name	Default Value	Required
scope	None	Yes, if using Azure AD OAuth2 for authentication.

SecretAccessKey

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

Key Name	Default Value	Required
SecretAccessKey	None	Yes, if using IAM credentials for authentication or AuthProfile.

SessionToken

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

Key Name	Default Value	Required
SessionToken	None	No

Server

A comma-delimited list of endpoint servers. The connector 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 connector alerts the user.

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

Key Name	Default Value	Required
Server	None	Yes, unless AWS Region and Cluster ID are specified.

Show Boolean Column As String

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

- Enabled (1): The connector returns Boolean columns as SQL_VARCHAR data with a length of 5.
- Disabled (0): The connector returns Boolean columns as SQL_BIT data.

Key Name	Default Value	Required
BoolsAsChar	Selected (1)	No

Single Row Mode

This option specifies whether the connector 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 connector returns query results one row at a time.
- Disabled (0): The connector returns all query results at once.

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

- If SingleRowMode and UseDeclareFetch are both set to 0, then the connector 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.

Key Name	Default Value	Required
SingleRowMode	Clear (0)	No

SSL Insecure

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

- Enabled (1): The connector does not check the authenticity of the IdP server certificate.
- Disabled (0): The connector checks the authenticity of the IdP server certificate.

Key Name	Default Value	Required
SSL_Insecure	Clear (0)	No

StsEndpointUrl

This option specifies the overriding endpoint used to communicate with the AWS Security Token Service (AWS STS).

Key Name	Default Value	Required
StsEndpointUrl	None	No

Text As LongVarChar

This option specifies the SQL data type that the connector 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.

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

Key Name	Default Value	Required
TextAsLongVarchar	Selected (1)	No

Timeout (sec)

The amount of time, in seconds, that the connector waits for the auth flow to complete.

Key Name	Default Value	Required
IdP_Response_Timeout	120	No

Use Declare/Fetch

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

- Enabled (1): The connector 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 connector returns all rows at once.

When using keys to set connector 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 connector returns query results one row at a time.
- If `UseDeclareFetch` and `SingleRowMode` are both set to `0`, then the connector retrieves the entire query result into memory.

Key Name	Default Value	Required
`UseDeclareFetch`	Clear (`0`)	No

Use HTTPS Proxy For Authentication On IdP

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

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

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

Key Name	Default Value	Required
`IdP_Use_Https_Proxy`	Clear (`0`)	Yes, if authenticating through an identity provider that can only be reached through a proxy connection.

Use Instance Profile

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

- Enabled (`1`): The connector uses the Amazon EC2 instance profile.
- Disabled (`0`): The connector uses the chained roles profile specified by the Profile Name option (the `Profile` property) instead. For more information, see Profile Name.

Key Name	Default Value	Required
`InstanceProfile`	Clear (`0`)	No

Use Multiple Statements

This option specifies whether the connector can have more than one query, separated by a semicolon (;), in a single SQLExecDirect call.

- Enabled (1): The connector can have more than one query, separated by semicolon (;), in a single SQLExecDirect call. The connector returns all the query results into memory.
- Disabled (0): The connector executes one query at a time in SQLExecDirect.

When using connection attributes to set connector 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.

Key Name	Default Value	Required
UseMultipleStatements	Disabled (0)	No

Use System Trust Store

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

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

Note: This option is only available in Windows.

Key Name	Default Value	Required
UseSystemTrustStore	Selected (1)	No

Use Unicode

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

- Enabled (1): The connector 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 connector 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 connector returns Redshift data as SQL types, see Data Types.

Key Name	Default Value	Required
`UseUnicode`	Selected (`1`)	No

User

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

If you are using keys to set connector 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.

Key Name	Default Value	Required

`UID`
OR
`User`
	None	Yes, if using Standard authentication.

User AutoCreate

This option specifies whether the connector 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 connector does not cause new users to be created. If the specified user does not exist, the authentication fails.

Key Name	Default Value	Required
`AutoCreate`	Clear (`0`)	No

VPC Endpoint URL

This option specifies the Redshift-managed VPC endpoint when IAM authentication is used.

Key Name	Default Value	Required
VpcEndpointUrl	None	No

Web Identity Token

The token that is provided by the identity provider.

Key Name	Default Value	Required
web_identity_token	None	Yes, if authenticating using a JWT or JwtIamAuthPlugin authentication.

Configuration Options Having Only Key Names

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

- ApplicationName
- cafile
- ConnectionTimeout
- Driver
- EnableAwsSdkLogs
- IAM
- KeepAlive
- KeepAliveCount
- KeepAliveInterval
- KeepAliveTime
- Locale
- plugin_name
- StsConnectionTimeout

The `UseLogPrefix` property must be configured as a Windows Registry key value, or as a connector-wide property in the `amazon.redshiftodbc.ini` file for macOS or Linux.

- UseLogPrefix

ApplicationName

This property sets the name of the current application on the server. If not set, `ApplicationName` is set to the connector name and version upon connection.

Key Name	Default Value	Required
`ApplicationName`	None	No

cafile

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

Note:
This option is only available in macOS and Linux.

Key Name	Default Value	Required
`cafile`	None	No

ConnectionTimeout

This property specifies the maximum wait time for the connection, in seconds. If set to `0` or not specified, the connector waits indefinitely. The minimum allowed value is `2`.

Key Name	Default Value	Required
`ConnectionTimeout`	`0`	No

Driver

In Windows, the name of the installed connector for (`Amazon Redshift ODBC Driver(x64)` for 64-bit connector).

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

Key Name	Default Value	Required
`Driver`	`Amazon Redshift ODBC Driver` when installed in Windows, or the absolute path of the connector shared object file when installed on a non-Windows machine.	Yes

EnableAwsSdkLogs

This option specifies whether the connector enables AWS SDK logging at the TRACE level.

- `1`: The connector enables AWS SDK logging at the TRACE level. Log files will be generated in the executable directory.
- `0`: The connector does not enable AWS SDK logging at the TRACE level.

Key Name	Default Value	Required
`EnableAwsSdkLogs`	`0`	No

IAM

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

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

Key Name	Default Value	Required
`IAM`	`0`	No

KeepAlive

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

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

Key Name	Default Value	Required
`KeepAlive`	`1`	No

KeepAliveCount

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

When this key is set to `0`, the connector uses the system default for this setting.

Key Name	Default Value	Required
`KeepAliveCount`	`0`	No

KeepAliveTime

The number of seconds of inactivity before the connector sends a TCP keepalive packet.

When this key is set to `0`, the connector uses the system default for this setting.

Key Name	Default Value	Required
KeepAliveTime	0	No

KeepAliveInterval

The number of seconds between each TCP keepalive retransmission.

When this key is set to 0, the connector uses the system default for this setting.

Key Name	Default Value	Required
KeepAliveInterval	0	No

Locale

The locale to use for error messages.

Key Name	Default Value	Required
Locale	en-US	No

plugin_name

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.
- AzureAD: Use Microsoft Azure Active Directory (AD) Service for authentication.
- BrowserAzureAD: Use a browser plugin for the Microsoft Azure Active Directory (AD) Service for authentication.
- BrowserAzureADOAuth2: Use a browser plugin for the Microsoft Azure Active Directory (AD) OAuth2 Service for authentication.
- BrowserSAML: Use a browser plugin for SAML services such as Okta or Ping for authentication.
- jwt: Use a JSON Web Token (JWT) generated by Microsoft Azure for authentication.
- JwtIamAuthPlugin: Use a JSON Web Token (JWT) generated by any other service for authentication.
- ping: Use the PingFederate service for authentication.
- okta: Use the Okta service for authentication.

In 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.

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 Connector DSN Setup dialog box in the Windows connector, the Auth Type option is used instead. For more information, see Auth Type.

Key Name	Default Value	Required
`plugin_name`	None	No

StsConnectionTimeout

This property specifies the maximum wait time for IAM connections, in seconds. If set to `0` or not specified, the connector waits 60 seconds for each STS call.

Key Name	Default Value	Required
`StsConnectionTimeout`	`0`	No

UseLogPrefix

This option specifies whether the connector 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 connector 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 connector 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 connector does not include the prefix in log file names.

  To configure this option for the Windows connector, you create a value for it in one of the following registry keys:
  - For a 32-bit connector installed on a 64-bit machine: **HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Amazon\Amazon Redshift ODBC Driver\Driver**
  - Otherwise: **HKEY_LOCAL_MACHINE\SOFTWARE\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 connector, you must use the `amazon.redshiftodbc.ini` file.

Key Name	Default Value	Required
UseLogPrefix	0	No

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.