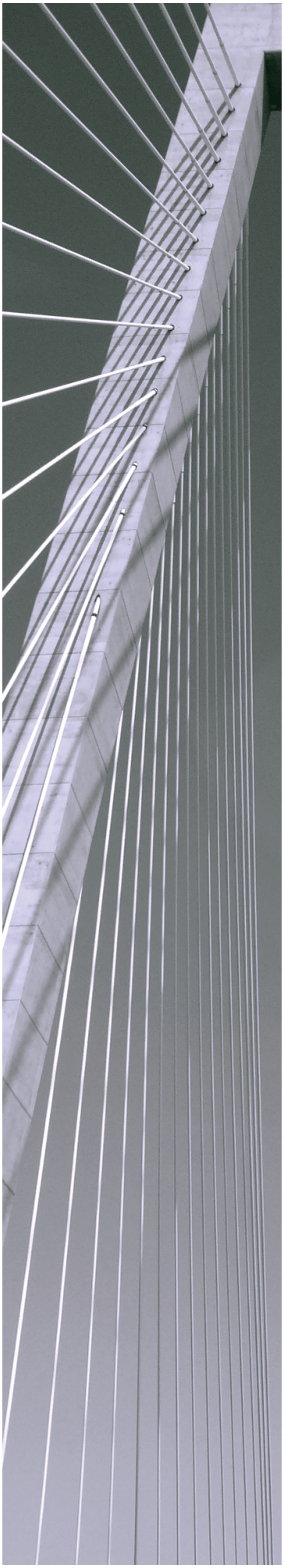

Simba JDBC Driver with SQL Connector for Google BigQuery

Installation and Configuration Guide

Simba Technologies Inc.

Version 1.1.2

September 29, 2017

Copyright © 2017 Simba Technologies 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 Simba Technologies Inc.

Trademarks

Simba, the Simba logo, SimbaEngine, and Simba Technologies are registered trademarks of Simba Technologies Inc. in Canada, United States and/or other countries. All other trademarks and/or servicemarks are the property of their respective owners.

Contact Us

Simba Technologies Inc.
938 West 8th Avenue
Vancouver, BC Canada
V5Z 1E5

Tel: +1 (604) 633-0008

Fax: +1 (604) 633-0004

www.simba.com

About This Guide

Purpose

The *Simba JDBC Driver with SQL Connector for Google BigQuery Installation and Configuration Guide* explains how to install and configure the Simba JDBC Driver with SQL Connector for Google BigQuery on all supported platforms. The guide also provides details related to features of the driver.

Audience

The guide is intended for end users of the Simba JDBC Driver for Google BigQuery.

Knowledge Prerequisites

To use the Simba JDBC Driver for Google BigQuery, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba JDBC Driver for Google BigQuery
- Ability to use the data store to which the Simba JDBC Driver for Google BigQuery is connecting
- An understanding of the role of JDBC technologies in connecting to a data store
- Experience creating and configuring JDBC connections
- Exposure to SQL

Document Conventions

Italics are used when referring to book and document titles.

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

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

> **Note:**
>
> A text box with a pencil icon indicates a short note appended to a paragraph.

! Important:

A text box with an exclamation mark indicates an important comment related to the preceding paragraph.

Table of Contents

About the Simba JDBC Driver for Google BigQuery 7

System Requirements 8

Simba JDBC Driver for Google BigQuery Files 9

Installing and Using the Simba JDBC Driver for Google BigQuery 10
- Referencing the JDBC Driver Libraries 10
- Registering the Driver Class 11
- Building the Connection URL 12

Configuring Authentication 13
- Using a Google User Account 13
- Using a Google Service Account 14
- Using Pre-Generated Access and Refresh Tokens 15
- Using Application Default Credentials 15

Configuring Logging 17

Features 19
- SQL Connector 19
- Data Types 19
- Nested and Repeated Records 20
- Arrays 21
- Security and Authentication 21
- Catalog and Schema Support 22
- Large Result Set Support 22
- Write-Back 23
- Positional Parameters 23

Driver Configuration Options 24
- AllowLargeResults 24
- Catalog (Project) 25
- LargeResultDataset 25
- LargeResultTable 26
- LogLevel 26
- LogPath 27
- MaxResults 27
- OAuthAccessToken 28

OAuthClientId 28
OAuthClientSecret 28
OAuthPvtKeyPath 29
OAuthRefreshToken 29
OAuthServiceAcctEmail 29
OAuthType 30
ProjectId 30
QueryDialect 30
Timeout 31
TimestampFallback 31
useQueryCache 32
Third-Party Trademarks 33
Third-Party Licenses 34

About the Simba JDBC Driver for Google BigQuery

The Simba JDBC Driver for Google BigQuery enables Business Intelligence (BI), analytics, and reporting on data that is stored in BigQuery. The driver complies with the JDBC 4.1 and 4.2 data standards.

JDBC 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 JDBC driver, which connects an application to the database. For more information about JDBC, see *Data Access Standards Glossary*: http://www.simba.com/resources/data-access-standards-library.

This guide is suitable for users who want to access data residing within BigQuery from their desktop environment. Application developers might also find the information helpful. Refer to your application for details on connecting via JDBC.

System Requirements

Each machine where you use the Simba JDBC Driver for Google BigQuery must have Java Runtime Environment (JRE) installed. The version of JRE that must be installed depends on the version of the JDBC API you are using with the driver. The following table lists the required version of JRE for each provided version of the JDBC API.

JDBC API Version	JRE Version
4.1	7.0 or later
4.2	8.0 or later

Simba JDBC Driver for Google BigQuery Files

The Simba JDBC Driver for Google BigQuery is delivered in the following two ZIP archives, where *[Version]* is the version number of the driver:

- `BigQueryJDBC41_[Version].zip`
- `BigQueryJDBC42_[Version].zip`

Each archive contains the driver supporting the JDBC API version indicated in the archive name, as well as release notes and third party license information.

Installing and Using the Simba JDBC Driver for Google BigQuery

To install the Simba JDBC Driver for Google BigQuery on your machine, extract the files from the appropriate ZIP archive to the directory of your choice.

! Important:

If you received a license file through email, then you must copy the file into the same directory as the `BigQueryJDBC41_[Version].jar` file or the `BigQueryJDBC42_[Version].jar` file before you can use the Simba JDBC Driver for Google BigQuery.

To access a BigQuery data store using the Simba JDBC Driver for Google BigQuery, you need to configure the following:

- The list of driver library files (see Referencing the JDBC Driver Libraries on page 10)
- The Driver or DataSource class (see Registering the Driver Class on page 11)
- The connection URL for the driver (see Building the Connection URL on page 12)

Referencing the JDBC Driver Libraries

Before you use the Simba JDBC Driver for Google BigQuery, the JDBC application or Java code that you are using to connect to the data store must be able to access the driver JAR files. In the application or code, specify all the JAR files that you extracted from the appropriate ZIP archive.

Using the Driver in a JDBC Application

Most JDBC applications provide a set of configuration options for adding a list of driver library files. Use the provided options to include all the JAR files from the ZIP archive as part of the driver configuration in the application. For more information, see the documentation for your JDBC application.

Using the Driver in Java Code

You must include all the driver library files in the class path. This is the path that the Java Runtime Environment searches for classes and other resource files. For more information, see "Setting the Class Path" in the Java SE Documentation:

- For Windows: http://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html
- For Linux and Solaris: http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html

Registering the Driver Class

Before connecting to the data store, you must register the appropriate class for your application.

The following is a list of the classes used to connect the Simba JDBC Driver for Google BigQuery to BigQuery data stores. The `Driver` classes extend `java.sql.Driver`, and the `DataSource` classes extend `javax.sql.DataSource` and `javax.sql.ConnectionPoolDataSource`.

To support JDBC 4.1, classes with the following fully-qualified class names (FQCNs) are available:

- `com.simba.googlebigquery.jdbc41.Driver`
- `com.simba.googlebigquery.jdbc41.DataSource`

To support JDBC 4.2, classes with the following FQCNs are available:

- `com.simba.googlebigquery.jdbc42.Driver`
- `com.simba.googlebigquery.jdbc42.DataSource`

The following sample code shows how to use the `DriverManager` to establish a connection for JDBC 4:

```
private static Connection connectViaDM() throws Exception
{
   Connection connection = null;
   connection = DriverManager.getConnection(CONNECTION_URL);
   return connection;
}
```

The following sample code shows how to use the `DataSource` class to establish a connection:

```
private static Connection connectViaDS() throws Exception
{
   Connection connection = null;
   DataSource ds = new
   com.simba.googlebigquery.jdbc41.DataSource();
```

```
    ds.setURL(CONNECTION_URL);
    connection = ds.getConnection();
    return connection;
}
```

Building the Connection URL

Use the connection URL to supply connection information to the data store that you are accessing. The following is the format of the connection URL for the Simba JDBC Driver for Google BigQuery:

```
jdbc:bigquery://[Host]:[Port];ProjectId=[Project];OAuthType=
[AuthValue];[Property1]=[Value1];[Property2]=[Value2];...
```

The variables are defined as follows:

- *[Host]* is the DNS or IP address of the server.
- *[Port]* is the number of the TCP port to connect to. Specifying the port number is optional if you are connecting to port 443.
- *[Project]* is the name of your BigQuery project.
- *[AuthValue]* is a number that specifies the type of authentication used by the driver. For more information, see OAuthType on page 30 and Configuring Authentication on page 13.
- *[Property1..N]* and *[Value1..N]* are additional connection properties supported by the driver. For a list of the properties available in the driver, see Driver Configuration Options on page 24.

! Important:

- Properties are case-sensitive.
- Do not duplicate properties in the connection URL.

Configuring Authentication

The Simba JDBC Driver for Google BigQuery uses the OAuth 2.0 protocol for authentication and authorization. It authenticates your connection through Google OAuth APIs. You can configure the driver to provide your credentials and authenticate the connection to the database using one of the following methods:

- Using a Google User Account on page 13
- Using a Google Service Account on page 14
- Using Pre-Generated Access and Refresh Tokens on page 15
- Using Application Default Credentials on page 15

Using a Google User Account

You can configure the driver to authenticate the connection with a Google user account.

You must provide your Google user account credentials to connect to the server. For more information about authenticating through OAuth 2.0 with a Google user account, see "Using OAuth 2.0 to Access Google APIs" in the Google Identity Platform documentation: https://developers.google.com/identity/protocols/OAuth2.

To configure user account authentication:

1. Connect to the server using a connection URL written in the following format:

   ```
   jdbc:bigquery://[Host]:[Port];ProjectId=[Project];
   OAuthType=1;
   ```

   The variables are defined as follows:

   - *[Host]* is the DNS or IP address of the server.
   - *[Port]* is the number of the TCP port to connect to. Specifying the port number is optional if you are connecting to port 443.
   - *[Project]* is the name of your BigQuery project.

   For example:

   ```
   jdbc:bigquery://https://www.googleapis.com/bigquery/v2:4
   43;ProjectId=MyBigQueryProject;OAuthType=1;
   ```

   The driver returns a connection URL, and requests an access token.

2. In a web browser, navigate to the connection URL, and provide your Google account name and password for authentication.

   The browser returns an access token.
3. In the driver, type or paste the access token and press **ENTER**.

For more information about connection URL syntax, see Building the Connection URL on page 12.

Using a Google Service Account

You can configure the driver to authenticate the connection with a Google service account. The service account can handle the authentication process so that no user input is required.

You must provide a Google service account email address and the full path to a private key file for the service account. You can download the private key file from the Google API console web page. For more information about OAuth authentication using a service account, see "Using OAuth 2.0 for Server to Server Applications" in the Google Identity Platform documentation: https://developers.google.com/identity/protocols/OAuth2ServiceAccount.

To configure service account authentication:

1. Set the `OAuthType` property to `0`.
2. Set the `ProjectID` property to the name of your BigQuery project.
3. Set the `OAuthServiceAcctEmail` property to your Google service account email address.
4. Set the `OAuthPvtKeyPath` property to the full path to the key file that is used to authenticate the service account email address. This parameter supports keys in `.pl2` or `.json` format.

For example, the following connection URL authenticates the connection using a service account:

```
jdbc:bigquery://https://www.googleapis.com/bigquery/v2:443;
ProjectId=MyBigQueryProject;OAuthType=0;
OAuthServiceAcctEmail==bqtest1@data-driver-
testing.iam.gserviceaccount.com;
OAuthPvtKeyPath=C:\SecureFiles\ServiceKeyFile.p12;
```

For more information about connection URL syntax, see Building the Connection URL on page 12.

Using Pre-Generated Access and Refresh Tokens

You can configure the driver to authenticate the connection using access or refresh tokens that have already been generated from the Google Authorization Server. When using this method, you can authenticate your connection by providing an access token, or by providing a refresh token along with a client ID and client secret.

For information about obtaining access and refresh tokens, see "Using OAuth 2.0 to Access Google APIs" in the Google Identity Platform documentation: https://developers.google.com/identity/protocols/OAuth2.

! Important:

When generating the tokens to access BigQuery, you must specify the `https://www.googleapis.com/auth/bigquery` scope. If you are working with federated tables, you should also specify the `https://www.googleapis.com/auth/cloud-platform` scope.

To configure authentication using an access or refresh token:

1. Set the `OAuthType` property to `2`.
2. Set the `ProjectID` property to the name of your BigQuery project.
3. Do one of the following:
   - Set `OAuthAccessToken` to your access token.
   - Or, set `OAuthRefreshToken` to your refresh token.
4. If you are using a refresh token, set the `OAuthClientId` property to to your client ID and set the `OAuthClientSecret` property to your client secret.

For example, the following connection URL authenticates the connection using a refresh token:

```
jdbc:bigquery://https://www.googleapis.com/bigquery/v2:443;
OAuthType=2;ProjectId=MyBigQueryProject;
OAuthAccessToken=a25c7cfd36214f94a79d;OAuthRefreshToken=1jt9
Pcyq8pr3lvu143pfl4r86;OAuthClientId=11b5516f132211e6;OAuthCl
ientSecret=bCD+E1f2Gxhi3J4klmN;
```

For more information about connection URL syntax, see Building the Connection URL on page 12.

Using Application Default Credentials

You can configure the driver to authenticate the connection using credentials obtained through Application Default Credentials on the environment, if they are available. For

information about how to configure Application Default Credentials, see "Google Application Default Credentials" in the Google Identity Platform documentation: https://developers.google.com/identity/protocols/application-default-credentials.

For more information about authenticating through OAuth 2.0, see "Using OAuth 2.0 to Access Google APIs" in the Google Identity Platform documentation: https://developers.google.com/identity/protocols/OAuth2.

To configure authentication using Application Default Credentials:

1. Set the `OAuthType` property to `3`.
2. Set the `ProjectID` property to the name of your BigQuery project.

For example:

```
jdbc:bigquery://https://www.googleapis.com/bigquery/v2:443;
OAuthType=3;ProjectId=MyBigQueryProject;
```

For more information about connection URL syntax, see Building the Connection URL on page 12.

Configuring Logging

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

! Important:

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

In the connection URL, set the `LogLevel` key to enable logging at the desired level of detail. The following table lists the logging levels provided by the Simba JDBC Driver for Google BigQuery, in order from least verbose to most verbose.

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

To enable logging:

1. Set the `LogLevel` property to the desired level of information to include in log files.
2. Set the `LogPath` property to the full path to the folder where you want to save log files. To make sure that the connection URL is compatible with all JDBC applications, escape the backslashes (`\`) in your file paths by typing another backslash.

For example, the following connection URL enables logging level 3 and saves the log files in the `C:\temp` folder:

```
jdbc:bigquery://localhost;LogLevel=3;LogPath=C:\\temp
```

3. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

   The Simba JDBC Driver for Google BigQuery produces the following log files in the location specified in the `LogPath` property:

   - A `BigQuery_connection_`*`[Number]`*`.log` file for each connection made to the database, where *[Number]* is a number that identifies each log file. This file logs driver activity that is specific to the connection.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

To disable logging:

1. Remove the `LogLevel` and `LogPath` properties from the connection URL.
2. To make sure that the new settings take effect, restart your JDBC application and reconnect to the server.

Features

More information is provided on the following features of the Simba JDBC Driver for Google BigQuery:

- SQL Connector on page 19
- Data Types on page 19
- Nested and Repeated Records on page 20
- Arrays on page 21
- Security and Authentication on page 21
- Catalog and Schema Support on page 22
- Large Result Set Support on page 22
- Write-Back on page 23
- Positional Parameters on page 23

SQL Connector

The SQL Connector feature of the driver allows applications to execute standard SQL queries or legacy BigQuery SQL queries against the database.

The driver does not support query prefixes, and instead determines which dialect to use based on the `QueryDialect` connection setting. By default, `QueryDialect` is set to `SQL` so that the driver executes queries using standard SQL syntax. For more information, see QueryDialect on page 30.

Data Types

The Simba JDBC Driver for Google BigQuery supports many common data formats, converting between BigQuery, SQL, and Java data types.

The following table lists the supported data type mappings.

BigQuery Type	SQL Type	Java Type
BOOL	SQL_BOOLEAN	BOOLEAN
BYTES	SQL_VARBINARY	BYTE[ ]
DATE	SQL_DATE	DATE

BigQuery Type	SQL Type	Java Type
DATETIME	SQL_VARCHAR	STRING
FLOAT64	SQL_DOUBLE	DOUBLE
INT64	SQL_BIGINT	BIGINTEGER
STRING	SQL_VARCHAR	STRING
TIME	SQL_TIME	TIME
TIMESTAMP	SQL_TIMESTAMP	TIMESTAMP

Nested and Repeated Records

The Simba JDBC Driver for Google BigQuery fully supports nested and repeated records.

The Standard SQL syntax represents the sub-components of record data as nested sub-types. In the example below, `city` and `years` belong to the base record type of `address`.

If the record column is specified in a query projection list, the driver returns the base record as a text representation of the JSON record object, and no flattening occurs. The dot operator (`.`) is used to select sub-components. For example, to select from `city`, the column name `address.city` should be used.

```
{
   "address":[
      {
         "city":"Vancouver",
         "years":5
      }
   ],
   "name":"Google"
}
```

In Legacy SQL, sub-components of record types are implicitly flattened and are represented as individual attributes. In the example below, the sub-components of `city` and `years` of the record `address` are represented as individual columns of `address_city` and `address_years`.

```
{
    "address_city":"Vancouver",
    "address_years":"5",
    "name":"Google"
}
```

Arrays

The Simba JDBC Driver for Google BigQuery fully supports array data types. The driver returns the base array type as a text representation of the JSON array object.

For example, the SQL statement SELECT [1,2,3] returns the following JSON:

```
{
    "v":[
        {
            "v":"1",
        },
        {
            "v":"2",
        },
        {
            "v":"3"
        }
    ]
}
```

Security and Authentication

To protect data from unauthorized access, BigQuery data stores require all connections to be authenticated using the OAuth 2.0 protocol. The Simba JDBC Driver for Google BigQuery provides mechanisms that allow you to complete an OAuth 2.0 authentication flow using a personal Google account, a Google service account, or Application Default Credentials. You can also specify an access token or refresh token that you have already generated from the Google Authorization Server, and use those credentials to connect to Google BigQuery.

When you connect to BigQuery using a personal Google account, a Google service account, or Application Default Credentials, the driver automatically initiates an OAuth 2.0 authentication flow. The driver retrieves an access token based on the credentials specified in the connection URL, and then uses the token to authenticate the connection to the database. When you connect using an access token or refresh token,

the driver authenticates the connection to BigQuery without going through an OAuth 2.0 authentication flow.

For detailed driver configuration instructions, see Configuring Authentication on page 13.

For more information about OAuth 2.0, see "Using OAuth 2.0 to Access Google APIs" in the Google Identity Platform documentation: https://developers.google.com/identity/protocols/OAuth2.

Catalog and Schema Support

The Simba JDBC Driver for Google BigQuery supports both catalogs and schemas to make it easy for the driver to work with various JDBC applications. Projects are mapped to catalogs, and table datasets are mapped to schemas. For more information, see Catalog (Project) on page 25.

Large Result Set Support

Normally, when you execute queries using Legacy SQL, result sets that are larger than 128MB (compressed) are not supported. To enable large result sets in Legacy SQL, the Simba JDBC Driver for Google BigQuery supports the `AllowLargeResults` option in BigQuery job configurations. Large result sets are always enabled in Standard SQL, so the `AllowLargeResults` option is not relevant when you execute queries using Standard SQL.

! Important:

Working with large data sets may cause you to reach the query limits defined in Google BigQuery. For information about query limits, see the following sections in the Google BigQuery documentation:

- "Queries" in *Quota Policy*: https://cloud.google.com/bigquery/quotas.
- "Limits" in *Running Interactive and Batch Queries*: https://cloud.google.com/bigquery/docs/running-queries.

In Legacy SQL

By default, when working in Legacy SQL, the driver stores large result sets using a temporary hidden dataset named "_simba_jdbc" and a temporary table with a name consisting of the prefix "temp_table_" followed by the time of table creation. These default temporary datasets and tables are deleted after 24 hours.

If you want to store large result sets for longer than 24 hours, you can configure the driver to create and use persistent tables and datasets. To do this, set the `LargeResultTable` and `LargeResultDataset` properties to the names of the table and dataset, respectively. Query results are stored in these tables until you overwrite them by running another query with the same table and dataset specified.

! Important:

When `AllowLargeResults` is enabled, all query results are written to and read from the temporary tables described above, regardless of the query and its result size. Because of this, the result cache is not available for subsequent queries, and you are billed for every query that you make.

In Standard SQL

When you execute queries using Standard SQL, large results are always supported, and the `AllowLargeResults` option is not used. All query results, regardless of size, are stored in temporary tables that serve as a snapshot in time. Note that the result cache is always available for subsequent queries when you use Standard SQL.

Write-Back

The Simba JDBC Driver for Google BigQuery supports Data Manipulation Language (DML) statements such as INSERT, UPDATE, and DELETE.

For example, the following INSERT statement is supported:

INSERT INTO MyTable (Col1, Col2) VALUES ("Key", "Value");”

Positional Parameters

A parameterized query contains placeholders that are used for parameters. The values of those parameters are supplied at execution time.

The Simba JDBC Driver for Google BigQuery supports SQL positional parameters. Parameters are specified in queries with a question mark (`?`).

For example, the following parameterized query is supported:

SELECT * FROM MyTable WHERE Col1=?”

Driver Configuration Options

Driver Configuration Options lists and describes the properties that you can use to configure the behavior of the Simba JDBC Driver for Google BigQuery.

You can set configuration properties using the connection URL. For more information, see Building the Connection URL on page 12.

Note:

Property names and values are case-sensitive.

AllowLargeResults

Default Value	Data Type	Required
0	Integer	No

Description

This option specifies whether the driver supports query results larger than 128MB when working in Legacy SQL.

- `1`: The driver allows query results that are larger than 128MB in size.
- `0`: The driver returns an error when query results are larger than 128MB in size.

When working in Standard SQL (the `QueryDialect` property is set to `SQL`), this option is always considered to be enabled. For more information about the supported SQL dialects, see QueryDialect on page 30.

Important:

When this option is enabled for Legacy SQL, all query results are written to and read from the temporary tables described above, regardless of the query and its result size. Because of this, the result cache is not available to subsequent queries, and you are billed for every query that you make.

For detailed information about how the driver stores large result sets, see Large Result Set Support on page 22.

Catalog (Project)

Key Name	Default Value	Required
Catalog	None	Yes

Description

The name of your BigQuery project. This project is the default project that the Simba JDBC Driver for Google BigQuery queries against, and is also the project that is billed for queries that are run using the DSN.

Simba JDBC Driver for Google BigQuery supports multiple catalogs, which are equivalent to projects in Google BigQuery terms.

For queries, tables in the projection list must be fully qualified, in the format of `catalog.schema.table'. If the catalog is not specified, the driver will assume the projectId specified by the "projectId" connection option.

For catalog functions, in order to retrieve information from the desired catalog, the JDBC **Connection.setCatalog(myCatalog)** method must be called with **myCatalog** set to the desired catalog.

LargeResultDataset

Default Value	Data Type	Required
_simba_jdbc	String	No

Description

The dataset for storing query results when using Legacy SQL with the AllowLargeResults property enabled. For more information, see AllowLargeResults on page 24 and Large Result Set Support on page 22.

LargeResultTable

Default Value	Data Type	Required
A value consisting of the prefix `temp_table_` followed by the time of table creation.	String	No

Description

The table for storing query results when using Legacy SQL with the `AllowLargeResults` property enabled. For more information, see AllowLargeResults on page 24 and Large Result Set Support on page 22.

LogLevel

Default Value	Data Type	Required
`0`	Integer	No

Description

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

! Important:

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

Set the property to one of the following numbers:

- `0`: Disable all logging.
- `1`: Enable logging on the FATAL level, which logs very severe error events that will lead the driver to abort.
- `2`: Enable logging on the ERROR level, which logs error events that might still allow the driver to continue running.
- `3`: Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.

- `4`: Enable logging on the INFO level, which logs general information that describes the progress of the driver.
- `5`: Enable logging on the DEBUG level, which logs detailed information that is useful for debugging the driver.
- `6`: Enable logging on the TRACE level, which logs all driver activity.

When logging is enabled, the driver produces the following log files in the location specified in the `LogPath` property:

- A `BigQuery_connection_[Number].log` file for each connection made to the database, where *[Number]* is a number that distinguishes each log file from the others. This file logs driver activity that is specific to the connection.

If the `LogPath` value is invalid, then the driver sends the logged information to the standard output stream (`System.out`).

LogPath

Default Value	Data Type	Required
The current working directory	String	No

Description

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

MaxResults

Default Value	Data Type	Required
`0`	Integer	No

Description

The maximum number of results that are displayed per page.

The default value of `0` allows an unlimited number of results to be displayed per page.

OAuthAccessToken

Default Value	Data Type	Required
None	String	No

Description

The pre-generated access token you are using to authenticate into BigQuery. For details, see Using Pre-Generated Access and Refresh Tokens on page 15.

OAuthClientId

Default Value	Data Type	Required
None	String	Yes, if using a refresh token.

Description

The Client ID you are using to authenticate into BigQuery with a pre-generated refresh token. For details, see Using Pre-Generated Access and Refresh Tokens on page 15.

OAuthClientSecret

Default Value	Data Type	Required
None	String	Yes, if using a refresh token.

Description

The client secret you are using to authenticate into BigQuery with a pre-generated refresh token. For details, see Using Pre-Generated Access and Refresh Tokens on page 15.

OAuthPvtKeyPath

Default Value	Data Type	Required
None	String	Yes, if `OAuthUserAuth=0`

Description

The full path to the `.p12` or `.json` key file that is used to authenticate the service account email address, if you are authenticating your connection using a service account. For more information, see Using a Google Service Account on page 14.

OAuthRefreshToken

Default Value	Data Type	Required
None	String	No

Description

The pre-generated refresh token you are using to authenticate into BigQuery. For details, see Using Pre-Generated Access and Refresh Tokens on page 15.

OAuthServiceAcctEmail

Default Value	Data Type	Required
None	String	Yes, if `OAuthUserAuth=0`

Description

The service account email ID that is used for authentication if you are using service authentication. For more information, see Using a Google Service Account on page 14.

OAuthType

Default Value	Data Type	Required
0	Integer	No

Description

This option specifies how the driver obtains or provides the credentials for OAuth 2.0 authentication.

- 0: The driver uses service-based OAuth authentication (see Using a Google Service Account on page 14).
- 1: The driver uses user-based OAuth authentication (see Using a Google User Account on page 13).
- 2: The driver uses pre-generated tokens for authentication (see Using Pre-Generated Access and Refresh Tokens on page 15).
- 3: The driver uses Application Default Credentials for authentication (see Using Application Default Credentials on page 15).

ProjectId

Default Value	Data Type	Required
None	String	Yes

Description

The name of your BigQuery project. This project is the default project that the Simba JDBC Driver for Google BigQuery queries against, and also the project that is billed for queries that are run using the DSN.

QueryDialect

Default Value	Data Type	Required
SQL	Enumerated	No

Description

This option specifies whether the driver executes queries using standard SQL syntax or the legacy BigQuery SQL syntax.

- `SQL`: The driver uses standard SQL.
- `BIG_QUERY`: The driver uses legacy SQL.

Timeout

Default Value	Data Type	Required
`10`	Integer	No

Description

The length of time, in seconds, that the driver waits for a query to retrieve the results of an executed job.

TimestampFallback

Default Value	Data Type	Required
`0`	Integer	No

Description

Important:

This connection property is deprecated. Only enable this option if you need to temporarily support connections that used driver version 1.1.0 or earlier, while transitioning your applications to connect using driver version 1.1.1 or later.

This property specifies whether the driver sets the default timezone of the JVM to UTC.

- `1`: The driver sets the default timezone of the JVM to UTC. This behavior is non-optimal, but consistent with the behavior from driver versions 1.1.0 and earlier.

Note:

Before enabling `TimestampFallback`, be aware of the following:

- This property applies to all connections that use the Simba JDBC Driver for Google BigQuery.
- Enabling this property causes the driver to change JVM settings, affecting other processes that are running on the same JVM.
- Once you have made a connection with this property enabled, all subsequent connections must also have this property enabled. Otherwise, the driver returns an error.

- `0`: The driver does not change the timezone settings on the JVM. This is the preferred driver behavior.

useQueryCache

Default Value	Data Type	Required
`true`	Boolean	No

Description

This option specifies whether the driver uses cached query results.

- `true`: The driver uses cached query results.
- `false`: The driver does not use cached query results.

Third-Party Trademarks

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Google BigQuery, Google, and BigQuery are trademarks or registered trademarks of Google, Inc. or its subsidiaries in Canada, the United States and/or other countries.

All other trademarks are trademarks of their respective owners.

Third-Party Licenses

The licenses for the third-party libraries that are included in this product are listed below.

Apache License, Version 2.0

The following notice is included in compliance with the Apache License, Version 2.0 and is applicable to all software licensed under the Apache License, Version 2.0.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

   "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is

included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

    (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

    (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

    (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

    (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

    You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required

for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software that is licensed under the Apache License, Version 2.0 (listed below):

Google API Client
Copyright © 2010-2016 Google, Inc.

Google BigQuery API v2 Client Library for Java
Copyright © 2010 Google, Inc.

Google GSON
Copyright © 2008-2011 Google, Inc.

Google Guava
Copyright © 2006-2015 Google, Inc.

Google HTTP Client
Copyright © 2010-2016 Google, Inc.

Google OAuth Client
Copyright © 2010-2016 Google, Inc.

Jackson
Copyright © 2009-2011 FasterXML, LLC

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.