



TIBCO® Data Virtualization

Google Drive Adapter Guide

Version 8.7.0 | October 2023

Contents

Contents	2
Google Drive Adapter	5
Google Drive Version Support	5
SQL Compliance	5
Getting Started	5
Connecting to Google Drive	5
Deploying the Google Drive Adapter	5
Basic Tab	6
Logging	6
Using OAuth Authentication	8
Advanced Settings	17
SQL Compliance	19
SELECT Statements	19
INSERT Statements	19
UPDATE Statements	19
DELETE Statements	20
EXECUTE Statements	20
Names and Quoting	20
Transactions and Batching	20
SELECT Statements	20
SELECT INTO Statements	23
INSERT Statements	23
UPDATE Statements	24
DELETE Statements	25
EXECUTE Statements	25
Data Model	26
Tables	26

Views	27
Stored Procedures	27
Tables	27
Views	36
Stored Procedures	47
Connection String Options	59
Auth Scheme	62
Firewall Password	62
Firewall Port	63
Firewall Server	63
Firewall Type	64
Firewall User	65
Initiate OAuth	66
Location	66
Log Modules	67
Max Rows	68
OAuth Access Token	68
OAuth Client Id	69
OAuth Client Secret	69
OAuth Expires In	70
OAuth JWT Cert	70
OAuth JWT Cert Password	71
OAuth JWT Cert Subject	72
OAuth JWT Cert Type	73
OAuth JWT Issuer	75
OAuth JWT Subject	75
OAuth Refresh Token	76
OAuth Settings Location	76
OAuth Token Timestamp	78
OAuth Verifier	78
Other	79
Proxy Auth Scheme	80

Proxy Auto Detect	81
Proxy Exceptions	82
Proxy Password	82
Proxy Port	83
Proxy Server	83
Proxy SSL Type	84
Proxy User	85
Readonly	86
SSL Server Cert	86
Team Drive Support	87
Timeout	88
TIBCO Product Documentation and Support Services	89
How to Access TIBCO Documentation	89
Release Version Support	90
How to Contact TIBCO Support	90
How to Join TIBCO Community	91
Legal and Third-Party Notices	92

Google Drive Adapter

Google Drive Version Support

The adapter models the Google Drive API as relational tables. By default version 3 of the API is used.

SQL Compliance

The [SQL Compliance](#) section shows the SQL syntax supported by the adapter and points out any limitations.

Getting Started

Connecting to Google Drive

[Basic Tab](#) shows how to authenticate to Google Drive and configure any necessary connection properties. Additional adapter capabilities can be configured using the available [Connection](#) properties on the Advanced tab. The Advanced Settings section shows how to set up more advanced configurations and troubleshoot connection errors.

Deploying the Google Drive Adapter

To deploy the adapter, you can execute the server_util utility via the command line by

1. Unzip the tdv.googledrive.zip file to the location of your choice.
2. Open a command prompt window.
3. Navigate to the <TDV_install_dir>/bin
4. Enter the server_util command with the -deploy option:

```
server_util -server <hostname> [-port <port>] -user <user> -
password <password> -deploy -package <TDV_install_
dir>/adapters/tdv.googledrive/tdv.googledrive.jar
```

Note: When deploying a build of an existing adapter, you will need to undeploy the existing adapter using the `server_util` command with the `-undeploy` option.

```
server_util -server <hostname> [-port <port>] -user <user> -password
<password> -undeploy -version 1 -name GoogleDrive
```

Basic Tab

Google Drive uses the OAuth authentication standard. You can authorize the adapter to connect to Google APIs on behalf of individual users or on behalf of a domain.

Logging

The adapter uses log4j to generate log files. The settings within the log4j configuration file are used by the adapter to determine the type of messages to log. The following categories can be specified:

- Error: Only error messages are logged.
- Info: Both Error and Info messages are logged.
- Debug: Error, Info, and Debug messages are logged.

The Other property of the adapter can be used to set Verbosity to specify the amount of detail to be included in the log file, that is:

```
Verbosity=4;
```

You can use Verbosity to specify the amount of detail to include in the log within a category. The following verbosity levels are mapped to the log4j categories:

- 0 = Error
- 1-2 = Info
- 3-5 = Debug

For example, if the log4j category is set to DEBUG, the Verbosity option can be set to 3 for the minimum amount of debug information or 5 for the maximum amount of debug information.

Note that the log4j settings override the Verbosity level specified. The adapter never logs at a Verbosity level greater than what is configured in the log4j properties. In addition, if Verbosity is set to a level less than the log4j category configured, Verbosity defaults to the minimum value for that particular category. For example, if Verbosity is set to a value less than 3 and the Debug category is specified, the Verbosity defaults to 3.

The following list is a breakdown of the Verbosity levels and the information that they log.

- 1 - Will log the query, the number of rows returned by it, the start of execution and the time taken, and any errors.
- 2 - Will log everything included in Verbosity 1 and HTTP headers.
- 3 - Will additionally log the body of the HTTP requests.
- 4 - Will additionally log transport-level communication with the data source. This includes SSL negotiation.
- 5 - Will additionally log communication with the data source and additional details that may be helpful in troubleshooting problems. This includes interface commands.

Configure Logging for the Google Drive Adapter

By default, logging is turned on without debugging. If debugging information is desired, uncomment the following line in the TDV Server's log4j.properties file (default location of this file is: C:\Program Files\TIBCO\TDV Server <version>\conf\server):

```
log4j.logger.com.cdata=DEBUG
```

The TDV Server must be restarted after changing the log4j.properties file, which can be accomplished by running the composite.bat script located at: C:\Program Files\TIBCO\TDV Server <version>\bin. Note that reauthenticating to the TDV Studio is required after restarting the server.

Here is an example of the calls:

```
.\composite.bat monitor restart
```

All logs for the adapter are written to the "cs_cdata.log" file as specified in the log4j properties.

Note: The "log4j.logger.com.cdata=DEBUG" option is not required if the **Debug Output Enabled** option is set to true within the TDV Studio. To set this option, navigate to **Administrator > Configuration**. Select **Server > Configuration > Debugging** and set the Debug Output Enabled option to **True**.

Using OAuth Authentication

Use the OAuth authentication standard to connect to Google Drive. You can authenticate with a user account or a service account. The adapter facilitates this as described below.

Using a User Account to Authenticate to Google Drive

The user account flow requires the authenticating user to interact with Google Drive via the browser.

Embedded Credentials

See [Embedded Credentials](#) to connect with the adapter's embedded credentials and skip creating a custom OAuth app.

Custom Credentials

Instead of connecting with the adapter's embedded credentials, you can register an app to obtain the [OAuthClientId](#) and [OAuthClientSecret](#).

When to Create a Custom OAuth App

Creating a custom OAuth app is optional as the adapter is already registered with Google Drive and you can connect with its embedded credentials. You might want to create a custom OAuth app to change the information displayed when users log into the Google Drive OAuth endpoint to grant permissions to the adapter.

Using a Service Account to Connect to Google Drive

Service accounts have silent authentication, without user authentication in the browser. You can also use a service account to delegate enterprise-wide access scopes to the adapter.

You need to create an OAuth application in this flow. You can then connect to Google Drive data that the service account has permission to access. See [Custom Credentials](#) for an authentication guide.

Creating a Custom OAuth App

See [Creating a Custom OAuth App](#) for a procedure.

Embedded Credentials

Authenticate using the Embedded OAuth Credentials

Desktop Authentication with the Embedded OAuth App

You can connect without setting any connection properties for your user credentials. After setting the following, you are ready to connect: InitiateOAuth: Set this to GETANDREFRESH. You can use InitiateOAuth to avoid repeating the OAuth exchange and manually setting the OAuthAccessToken. When you connect the adapter opens the OAuth endpoint in your default browser. Log in and grant permissions to the application. The adapter then completes the OAuth process.

1. Extracts the access token from the callback URL and authenticates requests.
2. Obtains a new access token when the old one expires.
3. Saves OAuth values in OAuthSettingsLocation to be persisted across connections.

Custom Credentials

Authenticate with a User Account

Desktop Authentication with a Custom OAuth App

Follow the steps below to authenticate with the credentials for a custom OAuth app. See [Creating a Custom OAuth App](#).

Get and Refresh the OAuth Access Token

After setting the following, you are ready to connect:

- OAuthClientId: Set this to the client Id assigned when you registered your app.
- OAuthClientSecret: Set this to the client secret assigned when you registered your app.
- InitiateOAuth: Set this to GETANDREFRESH. You can use InitiateOAuth to avoid repeating the OAuth exchange and manually setting the OAuthAccessToken.

When you connect the adapter opens the OAuth endpoint in your default browser. Log in and grant permissions to the application. The adapter then completes the OAuth process:

1. Extracts the access token from the callback URL and authenticates requests.
2. Refreshes the access token when it expires.
3. Saves OAuth values in OAuthSettingsLocation to be persisted across connections.

Authenticate with a Service Account

Service accounts have silent authentication, without user authentication in the browser. You can also use a service account to delegate enterprise-wide access scopes to the adapter.

You need to create an OAuth application in this flow. See [Creating a Custom OAuth App](#) to create and authorize an app. You can then connect to Google Drive data that the service account has permission to access.

After setting the following connection properties, you are ready to connect:

- InitiateOAuth: Set this to GETANDREFRESH.
- OAuthJWTCertType: Set this to "PFXFILE".
- OAuthJWTCert: Set this to the path to the .p12 file you generated.

- OAuthJWTCertPassword: Set this to the password of the .p12 file.
- OAuthJWTCertSubject: Set this to "*" to pick the first certificate in the certificate store.
- OAuthJWTSubject: Set this to the email address of the user for whom the application is requesting delegate access. Note that delegate access must be granted by an administrator.

When you connect the adapter completes the OAuth flow for a service account.

1. Creates and signs the JWT with the claim set required by the adapter.
2. Exchanges the JWT for the access token.
3. Saves OAuth values in OAuthSettingsLocation to be persisted across connections.
4. Submits the JWT for a new access token when the token expires.

Headless Machines

Using OAuth on a Headless Machine

The following sections show how to authenticate a headless server or another machine on which the adapter cannot open a browser. You can authenticate with a user account or with a service account.

Authenticate with a User Account

To authenticate with a user account, you need to authenticate from another machine. Authentication is a two-step process.

1. Instead of installing the adapter on another machine, you can follow the steps below to obtain the OAuthVerifier value. Or, you can install the adapter on another machine and transfer the OAuth authentication values, after you authenticate through the usual browser-based flow.
2. You can then configure the adapter to automatically refresh the access token from the headless machine.

You can follow the headless OAuth authentication flow using the adapter's embedded OAuth credentials or using the OAuth credentials for your custom OAuth app.

Using the Embedded OAuth Credentials

Obtain a Verifier Code

Follow the steps below to authenticate from another machine and obtain the OAuthVerifier connection property:

1. Click the following link to open the [Google Drive OAuth endpoint](#) in your browser.
2. Log in and grant permissions to the adapter. You are then redirected to the callback URL, which contains the verifier code.
3. Save the value of the verifier code. You will set this in the OAuthVerifier connection property.

On the headless machine, set the following connection properties to obtain the OAuth authentication values.

- OAuthVerifier: Set this to the verifier code.
- InitiateOAuth: Set this to REFRESH.
- OAuthSettingsLocation: Set this to persist the encrypted OAuth authentication values to the specified file.

After the OAuth settings file is generated, set the following properties to connect to data:

- OAuthSettingsLocation: Set this to the file containing the encrypted OAuth authentication values. Make sure this file gives read and write permissions to the adapter to enable the automatic refreshing of the access token.
- InitiateOAuth: Set this to REFRESH.

Transfer OAuth Settings

Follow the steps below to install the adapter on another machine, authenticate, and then transfer the resulting OAuth values.

On a second machine, install the adapter and connect with the following properties set:

- OAuthSettingsLocation: Set this to a writable text file.
- InitiateOAuth: Set this to GETANDREFRESH.

Test the connection to authenticate in the browser. The resulting authentication values are written, encrypted, to the path specified by OAuthSettingsLocation. Once you have successfully tested the connection, copy the OAuth settings file to your headless machine.

On the headless machine, set the following connection properties to connect to data:

- OAuthSettingsLocation: Set this to the path to your OAuth settings file. Make sure this file gives read and write permissions to the adapter to enable the automatic refreshing of the access token.

Using the Credentials for a Custom OAuth App

Create a Custom OAuth App

Creating a custom OAuth app is optional in the headless OAuth flow; you can skip creating an app by connecting with the adapter's embedded OAuth credentials. You might want to create a custom OAuth app to change the information displayed when users log into Google Drive to grant permissions to the adapter.

See [Creating a Custom OAuth App](#) for a procedure. You can then follow the procedures below to authenticate and connect to data.

Obtain a Verifier Code

Set the following properties on the headless machine:

- InitiateOAuth: Set this to OFF.
- OAuthClientId: Set this to the Client Id in your app settings.
- OAuthClientSecret: Set this to the Client Secret in your app settings.

You can then follow the steps below to authenticate from another machine and obtain the OAuthVerifier connection property.

1. Call the [GetOAuthAuthorizationURL](#) stored procedure with the CallbackURL input parameter set to the exact Redirect URI you specified in your app settings.
2. Open the returned URL in a browser. Log in and grant permissions to the adapter. You are then redirected to the callback URL, which contains the verifier code.
3. Save the value of the verifier code. You will set this in the OAuthVerifier connection property.

On the headless machine, set the following connection properties to obtain the OAuth authentication values:

- OAuthVerifier: Set this to the verifier code.
- OAuthSettingsLocation: Set this to persist the encrypted OAuth authentication values to the specified file.

- InitiateOAuth: Set this to REFRESH.

After the OAuth settings file is generated, set the following properties to connect to data:

- OAuthSettingsLocation: Set this to the file containing the encrypted OAuth authentication values. Make sure this file gives read and write permissions to the provider to enable the automatic refreshing of the access token.
- InitiateOAuth: Set this to REFRESH.

Transfer OAuth Settings

Follow the steps below to install the adapter on another machine, authenticate, and then transfer the resulting OAuth values.

On a second machine, install the adapter and connect with the following properties set:

- OAuthSettingsLocation: Set this to a writable text file.
- InitiateOAuth: Set this to GETANDREFRESH.
- OAuthClientId: Set this to the client Id assigned when you registered your app.
- OAuthClientSecret: Set this to the client secret assigned when you registered your app.

Test the connection to authenticate. The resulting authentication values are written, encrypted, to the path specified by OAuthSettingsLocation. Once you have successfully tested the connection, copy the OAuth settings file to your headless machine. On the headless machine, set the following connection properties to connect to data:

- InitiateOAuth: Set this to REFRESH.
- OAuthSettingsLocation: Set this to the path to your OAuth settings file. Make sure this file gives read and write permissions to the adapter to enable the automatic refreshing of the access token.

Authenticate with a Service Account

Service accounts have silent authentication, without user authentication in the browser. You can also use a service account to delegate enterprise-wide access scopes to the adapter.

You need to create an OAuth application in this flow. See [Creating a Custom OAuth App](#) to create and authorize an app. You can then connect to Google Drive data that the service account has permission to access.

After setting the following connection properties, you are ready to connect:

- InitiateOAuth: Set this to GETANDREFRESH.
- OAuthClientId: Set this to the Client Id in your app settings.
- OAuthClientSecret: Set this to the Client Secret in your app settings.
- OAuthJWTCertType: Set this to "PFXFILE".
- OAuthJWTCert: Set this to the path to the .p12 file you generated.
- OAuthJWTCertPassword: Set this to the password of the .p12 file.
- OAuthJWTCertSubject: Set this to "*" to pick the first certificate in the certificate store.
- OAuthJWTIssuer: In the service accounts section, click Manage Service Accounts and set this field to the email address displayed in the service account Id field.
- OAuthJWTSubject: Set this to your enterprise Id if your subject type is set to "enterprise" or your app user Id if your subject type is set to "user".

When you connect the adapter completes the OAuth flow for a service account.

1. Creates and signs the JWT with the claim set required by the adapter.
2. Exchanges the JWT for the access token.
3. Saves OAuth values in OAuthSettingsLocation to be persisted across connections.
4. Submits the JWT for a new access token when the token expires.

Creating a Custom OAuth App

The adapter facilitates the following OAuth authentication flows:

- The user consent flow enables individual users to connect to their own data.
- The service account flow enables access to domain-wide data.

Using a User Account to Connect to Google

This OAuth flow requires the authenticating user to interact with Google using the browser. The adapter facilitates this in various ways as described below.

Authenticate to Google

After setting `InitiateOAuth` to `GETANDREFRESH`, you are ready to connect. You can use `InitiateOAuth` to avoid repeating the OAuth exchange and manually setting the `OAuthAccessToken` connection property. When you connect the adapter opens the OAuth endpoint in your default browser. Log in and grant permissions to the application. The adapter then completes the OAuth process:

1. Extracts the access token from the callback URL and authenticates requests.
2. Refreshes the access token when it expires.
3. Saves OAuth values to be persisted across connections. This file can be configured in `OAuthSettingsLocation`.

Using a Service Account to Connect to Domain-Wide Data

You can use a service account in this OAuth flow to access Google APIs on behalf of users in a domain. A domain administrator can delegate domain-wide access to the service account.

To complete the service account flow, generate a private key in the Google APIs Console. In the service account flow, the adapter exchanges a JSON Web token (JWT) for the `OAuthAccessToken`. The private key is required to sign the JWT. The `OAuthAccessToken` authenticates that the adapter has the same permissions granted to the service account.

Generate a Private Key

Follow the steps below to generate a private key and obtain the credentials for your application:

1. Log into the Google API Console.
2. Click Create Project or select an existing project.
3. In the API Manager, click Credentials -> Create Credentials -> Service Account Key. In

the Service Account menu, select New Service Account or select an existing service account. In the Key Type section, select the P12 key type.

4. Click Create to download the key pair. The private key's password is displayed: Set this in OAuthJWTCertPassword.
5. In the Service Account Keys section on the Credentials page, click Manage Service Accounts and set OAuthJWTIssuer to the email address displayed in service account Id.
6. Click Library -> Google Drive API -> Enable API.

Authenticate with a Service Account

After setting the following connection properties, you are ready to connect:

- InitiateOAuth: Set this to GETANDREFRESH. You can use InitiateOAuth to avoid repeating the OAuth exchange and manually setting the OAuthAccessToken connection property.
- OAuthJWTCertType: Set this to "PFXFILE".
- OAuthJWTCertPassword: Set this to the password of the .p12 file.
- OAuthJWTCertSubject: Set this to "*" to pick the first certificate in the certificate store.
- OAuthJWTIssuer: Set this to the email address of the service account.
- OAuthJWTCert: Set this to the path to the .p12 file.
- OAuthJWTSubject: Set this to the email address of the user for whom the application is requesting delegate access.

When you connect the adapter completes the OAuth flow for a service account:

1. Creates and signs the JWT with the claim set required by the adapter.
2. Exchanges the JWT for the access token.
3. Submits the JWT for a new access token when the token expires.

Advanced Settings

Customizing the SSL Configuration

By default, the adapter attempts to negotiate SSL/TLS by checking the server's certificate against the system's trusted certificate store. To specify another certificate, see the SSLServerCert property for the available formats to do so.

Connecting Through a Firewall or Proxy

HTTP Proxies

To connect through the Windows system proxy, you do not need to set any additional connection properties. To connect to other proxies, set ProxyAutoDetect to false.

In addition, to authenticate to an HTTP proxy, set ProxyAuthScheme, ProxyUser, and ProxyPassword, in addition to ProxyServer and ProxyPort.

Other Proxies

Set the following properties:

- To use a proxy-based firewall, set FirewallType, FirewallServer, and FirewallPort.
- To tunnel the connection, set FirewallType to TUNNEL.
- To authenticate, specify FirewallUser and FirewallPassword.
- To authenticate to a SOCKS proxy, additionally set FirewallType to SOCKS5.

Troubleshooting the Connection

To show adapter activity from query execution to network traffic, use Logfile and Verbosity. The examples of common connection errors below show how to use these properties to get more context. Contact the support team for help tracing the source of an error or circumventing a performance issue.

- **Authentication errors:** Typically, recording a Logfile at Verbosity 4 is necessary to get full details on an authentication error.
- **Queries time out:** A server that takes too long to respond will exceed the adapter's

client-side timeout. Often, setting the [Timeout](#) property to a higher value will avoid a connection error. Another option is to disable the timeout by setting the property to 0. Setting [Verbosity](#) to 2 will show where the time is being spent.

- **The certificate presented by the server cannot be validated:** This error indicates that the adapter cannot validate the server's certificate through the chain of trust. If you are using a self-signed certificate, there is only one certificate in the chain.

To resolve this error, you must verify yourself that the certificate can be trusted and specify to the adapter that you trust the certificate. One way you can specify that you trust a certificate is to add the certificate to the trusted system store; another is to set [SSLServerCert](#).

SQL Compliance

The Google Drive Adapter supports several operations on data, including querying, deleting, modifying, and inserting.

SELECT Statements

See [SELECT Statements](#) for a syntax reference and examples.

See [Data Model](#) for information on the capabilities of the Google Drive API.

INSERT Statements

See [INSERT Statements](#) for a syntax reference and examples.

UPDATE Statements

The primary key Id is required to update a record. See [UPDATE Statements](#) for a syntax reference and examples.

DELETE Statements

The primary key Id is required to delete a record. See [DELETE Statements](#) for a syntax reference and examples.

EXECUTE Statements

Use EXECUTE or EXEC statements to execute stored procedures. See [EXECUTE Statements](#) for a syntax reference and examples.

Names and Quoting

- Table and column names are considered identifier names; as such, they are restricted to the following characters: [A-Z, a-z, 0-9, _:@].
- To use a table or column name with characters not listed above, the name must be quoted using double quotes ("name") in any SQL statement.
- Strings must be quoted using single quotes (e.g., 'John Doe').

Transactions and Batching

Transactions are not currently supported.

Additionally, the adapter does not support batching of SQL statements. To execute multiple commands, you can create multiple instances and execute each separately.

SELECT Statements

A SELECT statement can consist of the following basic clauses.

- SELECT
- INTO
- FROM
- JOIN

- WHERE
- GROUP BY
- HAVING
- UNION
- ORDER BY
- LIMIT

SELECT Syntax

The following syntax diagram outlines the syntax supported by the Google Drive adapter:

```

SELECT {
  [ TOP <numeric_literal> ]
  {
    *
    | {
        <expression> [ [ AS ] <column_reference> ]
        | { <table_name> | <correlation_name> } .*
      } [ , ... ]
    }
  [ INTO csv:// [ filename= ] <file_path> [ ;delimiter=tab ] ]
  {
    FROM <table_reference> [ [ AS ] <identifier> ]
  }
  [ WHERE <search_condition> ]
  [
    ORDER BY
    <column_reference> [ ASC | DESC ] [ NULLS FIRST | NULLS LAST ]
  ]
  [
    LIMIT <expression>
  ]
} | SCOPE_IDENTITY()
<expression> ::=
  | <column_reference>
  | @ <parameter>
  | ?
  | COUNT( * | { [ DISTINCT ] <expression> } )
  | { AVG | MAX | MIN | SUM | COUNT } ( <expression> )
  | NULLIF ( <expression> , <expression> )
  | COALESCE ( <expression> , ... )
  | CASE <expression>

```

```

        WHEN { <expression> | <search_condition> } THEN { <expression> |
NULL } [ ... ]
    [ ELSE { <expression> | NULL } ]
    END
    | <literal>
    | <sql_function>
<search_condition> ::=
{
    <expression> { = | AND | IN | >,<,<=> } [ <expression> ]
} [ { AND | OR } ... ]

```

Examples

1. Return all columns:

```
SELECT * FROM Files
```

2. Rename a column:

```
SELECT "Name" AS MY_Name FROM Files
```

3. Cast a column's data as a different data type:

```
SELECT CAST(AnnualRevenue AS VARCHAR) AS Str_AnnualRevenue FROM
Files
```

4. Search data:

```
SELECT * FROM Files WHERE Extension = 'png';
```

5. The Google Drive APIs support the following operators in the WHERE clause: =, AND, IN, >,<,<=>.

```
SELECT * FROM Files WHERE Extension = 'png';
```

6. Sort a result set in ascending order:

```
SELECT Id, Name FROM Files ORDER BY Name ASC
```

SELECT INTO Statements

You can use the SELECT INTO statement to export formatted data to a file.

Data Export with an SQL Query

The following query exports data into a file formatted in comma-separated values (CSV):

```
boolean ret = stat.execute("SELECT Id, Name INTO 'csv://c:/Files.txt'
FROM 'Files' WHERE Extension = 'png'");
System.out.println(stat.getUpdateCount()+" rows affected");
```

You can specify other file formats in the URI. The following example exports tab-separated values:

```
Statement stat = conn.createStatement();
boolean ret = stat.execute("SELECT * INTO 'Files' IN
'csv://filename=c:/Files.csv;delimiter=tab' FROM 'Files' WHERE Extension
= 'png'");
System.out.println(stat.getUpdateCount()+" rows affected");
```

INSERT Statements

To create new records, use INSERT statements.

INSERT Syntax

The INSERT statement specifies the columns to be inserted and the new column values. You can specify the column values in a comma-separated list in the VALUES clause, as shown in the following example:

```
INSERT INTO <table_name>
( <column_reference> [ , ... ] )
```

```
VALUES
( { <expression> | NULL } [ , ... ] )

<expression> ::=
| @ <parameter>
| ?
| <literal>
```

You can use the `executeUpdate` method of the `Statement` and `PreparedStatement` classes to execute data manipulation commands and retrieve the rows affected.

```
String cmd = "INSERT INTO Files (Name) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "My File 2");
int count = pstmt.executeUpdate();
System.out.println(count+" rows were affected");
connection.close();
```

UPDATE Statements

To modify existing records, use UPDATE statements.

Update Syntax

The UPDATE statement takes as input a comma-separated list of columns and new column values as name-value pairs in the SET clause, as shown in the following example:

```
UPDATE <table_name> SET { <column_reference> = <expression> } [ , ... ]
WHERE { Id = <expression> } [ { AND | OR } ... ]
<expression> ::=
| @ <parameter>
| ?
| <literal>
```

You can use the `executeUpdate` method of the `Statement` or `PreparedStatement` classes to execute data manipulation commands and retrieve the rows affected, as shown in the following example:

```
String cmd = "UPDATE Files SET Name='My File 2' WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "S");
```



```
int count = pstmt.executeUpdate();
System.out.println(count + " rows were affected");
connection.close();
```

DELETE Statements

To delete information from a table, use DELETE statements.

DELETE Syntax

The DELETE statement requires the table name in the FROM clause and the row's primary key in the WHERE clause, as shown in the following example:

```
<delete_statement> ::= DELETE FROM <table_name> WHERE { Id =
<expression> } [ { AND | OR } ... ]
<expression> ::=
    | @ <parameter>
    | ?
    | <literal>
```

You can use the executeUpdate method of the Statement or PreparedStatement classes to execute data manipulation commands and retrieve the number of affected rows, as shown in the following example:

```
Connection connection = DriverManager.getConnection
("jdbc:googledrive:InitiateOAuth=GETANDREFRESH;",);
String cmd = "DELETE FROM Files WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "S");
int count=pstmt.executeUpdate();
connection.close();
```

EXECUTE Statements

To execute stored procedures, you can use EXECUTE or EXEC statements.

EXEC and EXECUTE assign stored procedure inputs, referenced by name, to values or parameter names.

Stored Procedure Syntax

To execute a stored procedure as an SQL statement, use the following syntax:

```
{ EXECUTE | EXEC } <stored_proc_name>
{
  [ @ ] <input_name> = <expression>
} [ , ... ]
<expression> ::=
  | @ <parameter>
  | ?
  | <literal>
```

Example Statements

Reference stored procedure inputs by name:

```
EXECUTE my_proc @second = 2, @first = 1, @third = 3;
```

Execute a parameterized stored procedure statement:

```
EXECUTE my_proc second = @p1, first = @p2, third = @p3;
```

Data Model

The Google Drive Adapter models Google Drive APIs as relational Tables, Views, and Stored Procedures. These are defined in schema files, which are simple, text-based configuration files. API limitations and requirements are documented in this section; you can use the [SupportEnhancedSQL](#) feature, set by default, to circumvent most of these limitations.

Tables

[Tables](#) describes the available tables.

Views

[Views](#) are tables that cannot be modified. Typically, data that are read-only and cannot be updated are shown as views.

Stored Procedures

[Stored Procedures](#) are function-like interfaces to the data source. They can be used to search, update, and modify information in the data source.

Tables

The adapter models the data in Google Drive into a list of tables that can be queried using standard SQL statements.

Generally, querying Google Drive tables is the same as querying a table in a relational database. Sometimes there are special cases, for example, including a certain column in the WHERE clause might be required to get data for certain columns in the table. This is typically needed for situations where a separate request must be made for each row to get certain columns. These types of situations are clearly documented at the top of the table page linked below.

Google Drive Adapter Tables

Name	Description
Files	Create, update, delete, and query the files and folders contained in a user's Google Drive.
Permissions	Create, update, delete, and query permissions for resources in a user's Google Drive.
TeamDrives	Create, delete, and query the available TeamDrives for a specific user.

Files

Create, update, delete, and query the files and folders contained in a user's Google Drive.

Select

The Files table supports only a subset of columns for filtering. Below is a table containing those columns with their supported operations. All filters can be connected with 'OR' or 'AND' operators.

Column	Supported Operators
Name	contains, =, !=
MIMETYPE	contains, =, !=
ModifiedTime	<=, <, =, !=, >, >=
Trashed	=, !=
Starred	=, !=
ParentIds	in
OwnerEmail	in

The **contains** operator only performs prefix matching for a **name**. For example, the name "HelloWorld" would match for **name contains 'Hello'** but not **name contains 'World'**.

```
SELECT * FROM [Files] WHERE ModifiedTime>'2017-01-01' OR Contains
(Name, 'CData')
```

```
SELECT * FROM [Files] WHERE OwnerEmail in ('support@cdata.com') AND
Starred = true
```

```
SELECT * FROM [Files] WHERE Starred = true
```

```
SELECT * FROM [Files] WHERE TeamDriveId='0ACkq0ZiV0yJCuk9PVA'
```

Note: You must set the connection property TeamDriveSupport to 'true', in order to query from a specific Team Drive.

Insert

You must specify values at least for Name and one of LocalFile or FileData.

```
Insert into Files (Name,LocalFile) VALUES('MyFile','C:\\\\file.txt')
```

Update

Id is required for updating a File.

```
Update Files SET Name='UpdatedName' WHERE Id =  
'19YFv8wmvKixCYaJJAeE8jN3R0t7x1ZicvXwflswV0rw'
```

Also the content of the file can be updated. Note that this will replace the actual content.

```
Update Files SET LocalFile='C:\\\\file.txt' WHERE Id =  
'19YFv8wmvKixCYaJJAeE8jN3R0t7x1ZicvXwflswV0rw'
```

Delete

To delete a File, the Id is required.

```
DELETE FROM [Files] WHERE Id =  
'1Dx6GTyhgTmTjtoy8GuG0n0qa0sKyhwrOG6MG8A2QQYA'
```

Columns

Name	Type	ReadOnly	Description
Id [KEY]	<i>String</i>	True	The ID of the file.
Name	<i>String</i>	False	The name of the file. This is not necessarily unique within a folder. Note that for immutable items such as the top level folders of Team Drives, My Drive root folder, and Application Data folder the name is constant.
TeamDriveId	<i>String</i>	True	The Id of the teamDrive.
Description	<i>String</i>	False	A short description of the file or folder.
Extension	<i>String</i>	True	The extension of the file.
MIMETYPE	<i>String</i>	False	The MIME type of the file.
CreatedTime	<i>Datetime</i>	True	The creation date of the file or folder.
ModifiedTime	<i>Datetime</i>	True	The last modified date of the file or folder.
Size	<i>Long</i>	True	The size of the file in bytes.
OwnerName	<i>String</i>	True	The name of the resource's owner.
OwnerEmail	<i>String</i>	True	The email of the resource's owner.

Folder	<i>Boolean</i>	True	This field shows whether or not the resource is a folder.
Starred	<i>Boolean</i>	False	This field sets whether or not the resource is starred.
Trashed	<i>Boolean</i>	True	This field sets whether or not the resource has been moved to the trash.
Viewed	<i>Boolean</i>	True	This field sets whether or not the resource has been viewed by the current user.
ParentIds	<i>String</i>	True	A comma-separated list of parent folder ids.
ChildIds	<i>String</i>	True	A semicolon-separated list of child resource ids.
ChildLinks	<i>String</i>	True	A semicolon-separated list of child resource links.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Query	<i>String</i>	This field accepts a valid Google Drive SDK query, which overrides conditionals in the WHERE clause.
LocalFile	<i>String</i>	The local file path, including file name, of the file to be uploaded. Required when FileData is not specified. Used only for inserting and updating a file.
FileData	<i>String</i>	If the LocalFile input is empty, file data will be output in the format

		specified by the Encoding input. Used only for inserting and updating a file.
Encoding	<i>String</i>	<p>The FileData input encoding type. Used only for inserting and updating a file.</p> <p>The allowed values are <i>NONE</i>, <i>BASE64</i>.</p> <p>The default value is <i>BASE64</i>.</p>

Permissions

Create, update, delete, and query permissions for resources in a user's Google Drive.

Select

The ResourceId field must be specified to get data from this table. This is the only supported filter.

```
SELECT * FROM Permissions WHERE ResourceId =
'0B5AH3NIqjXDKX3pJS3NncTZJa01'
```

Insert

To insert into Permissions, you must specify values at least for the ResourceId, Role, Type, and EmailAddress fields.

```
INSERT INTO Permissions (ResourceId, Role, Type, EmailAddress) VALUES
('0B5AH3NIqjXDKX3pJS3NncTZJa01', 'WRITER', 'USER', 'support@cdata.com')
```

Update

The PermissionId and ResourceId fields are required for updating a Permission.

```
UPDATE Permissions SET Role= 'organizer' where PermissionId ='3NIqjXDK'
AND ResourceId= '0B5AH3NIqjXDKX3pJS3NncTZJa01'
```


Delete

To delete a Permission, the PermissionId and ResourceId fields are required.

```
Delete From Permissions WHERE PermissionId ='3NIqjXDK' AND ResourceId=
'0B5AH3NIqjXDKX3pJS3NncTZJa01'
```

Columns

Name	Type	ReadOnly	Description
PermissionId [KEY]	String	True	The Id of the permission for the resource.
ResourceId [KEY]	String	True	The Id of the resource (a file or folder).
Role	String	False	The role specified for the permission.
Type	String	False	The type entity to which the permission applies. It can take only these values: 'USER', 'GROUP', 'DOMAIN', 'ANYONE'.
EmailAddress	String	False	The email address of the user or group to which this permission refers.
Domain	String	False	The domain to which this permission refers.
AllowFileDiscovery	Boolean	False	Whether the permission allows the file or folder to be discovered through search. This is only applicable for permissions of type 'DOMAIN' or 'ANYONE'. The default value for this field is 'false'

TeamDrives

Create, delete, and query the available TeamDrives for a specific user.

Select

The TeamDrives table supports only a subset of columns for filtering. Below is a table containing those columns with their supported operations. All filters can be connected with 'OR' or 'AND' operators.

Column	Supported Operators
Name	contains, =, !=
CreatedTime	<=, <, =, !=, >, >=

```
SELECT * FROM TeamDrives ORDER BY NAME DESC LIMIT 2
```

```
SELECT * FROM TeamDrives WHERE Name='First Team Drive' AND  
DomainAdminAccess=true
```

```
SELECT * FROM TeamDrives WHERE CreatedTime>='2018-01-01' AND  
DomainAdminAccess=true
```

Insert

You must specify a value for the field 'Name', in order to insert a new Team Drive.

```
INSERT INTO TeamDRIVES(Name) VALUES('TestTeamDriveCDATA')
```

Delete

To delete a Team Drive, the Id is required.

```
DELETE FROM TeamDrives WHERE Id='0ALw8avjM8FrtUk9PVA'
```

Columns

Name	Type	ReadOnly	Description
Id [KEY]	<i>String</i>	True	The Id of the team drive.
Name	<i>String</i>	False	The name of the team drive.
Capabilities	<i>String</i>	True	This field describes the effective capabilities that the current user has for the team drive.
CreatedTime	<i>Datetime</i>	True	The creation date of the team drive.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
DomainAdminAccess	<i>Boolean</i>	If this field is set to true, then all Team Drives of the domain in which you are an administrator are returned.

Views

Views are composed of columns and pseudo columns. Views are similar to tables in the way that data is represented; however, views do not support updates. Entities that are represented as views are typically read-only entities. Often, a stored procedure is available to update the data if such functionality is applicable to the data source.

Queries can be executed against a view as if it were a normal table, and the data that comes back is similar in that regard. To find out more about tables and stored procedures, please navigate to their corresponding entries in this help document.

Google Drive Adapter Views

Name	Description
Docs	Query the Google Docs contained in a user's Google Drive.
Folders	Query the folders contained in a user's Google Drive.
Photos	Query the photos contained in a user's Google Drive.
Sheets	Query the Google Sheets contained in a user's Google Drive.
Videos	Query the Google Videos contained in a user's Google Drive.

Docs

Query the Google Docs contained in a user's Google Drive.

Select

It is also possible to get all the docs from a Team Drive.

Note: You must set the connection property [TeamDriveSupport](#) to 'true', in order to query

from a specific Team Drive.

```
SELECT * FROM Docs WHERE TeamDriveId='0ACkq0ZiV0yJCuk9PVA'
```

Columns

Name	Type	Description
Id [KEY]	<i>String</i>	The ID of the file.
Name	<i>String</i>	The name of the file. This is not necessarily unique within a folder. Note that for immutable items such as the top level folders of Team Drives, My Drive root folder, and Application Data folder the name is constant.
TeamDriveId	<i>String</i>	The Id of the teamDrive.
Description	<i>String</i>	A short description of the file or folder.
Extension	<i>String</i>	The extension of the file.
CreatedTime	<i>Datetime</i>	The creation date of the file or folder.
ModifiedTime	<i>Datetime</i>	The last modified date of the file or folder.
Size	<i>Long</i>	The size of the file in bytes.
OwnerName	<i>String</i>	The name of the resource's owner.

OwnerEmail	<i>String</i>	The email of the resource's owner.
Starred	<i>Boolean</i>	This field sets whether or not the resource is starred.
Trashed	<i>Boolean</i>	This field sets whether or not the resource has been moved to the trash.
Viewed	<i>Boolean</i>	This field sets whether or not the resource has been viewed by the current user.
ParentIds	<i>String</i>	A comma-separated list of parent folder ids.
ChildIds	<i>String</i>	A semicolon-separated list of child resource ids.
ChildLinks	<i>String</i>	A semicolon-separated list of child resource links.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Query	<i>String</i>	This field accepts a valid Google Drive SDK query, which overrides conditionals in the WHERE clause.

Folders

Query the folders contained in a user's Google Drive.

Select

It is also possible to get all the folders from a Team Drive.

Note: You must set the connection property TeamDriveSupport to 'true', in order to query from a specific Team Drive.

```
SELECT * FROM Folders WHERE TeamDriveId='0ACkq0ZiV0yJCuk9PVA'
```

Columns

Name	Type	Description
Id [KEY]	<i>String</i>	The ID of the file.
Name	<i>String</i>	The name of the file. This is not necessarily unique within a folder. Note that for immutable items such as the top level folders of Team Drives, My Drive root folder, and Application Data folder the name is constant.
TeamDriveId	<i>String</i>	The Id of the teamDrive.
Description	<i>String</i>	A short description of the file or folder.
CreatedTime	<i>Datetime</i>	The creation date of the file or folder.
ModifiedTime	<i>Datetime</i>	The last modified date of the file or folder.
Size	<i>Long</i>	The size of the file in bytes.
OwnerName	<i>String</i>	The name of the resource's owner.

OwnerEmail	<i>String</i>	The email of the resource's owner.
Starred	<i>Boolean</i>	This field sets whether or not the resource is starred.
Trashed	<i>Boolean</i>	This field sets whether or not the resource has been moved to the trash.
Viewed	<i>Boolean</i>	This field sets whether or not the resource has been viewed by the current user.
ParentIds	<i>String</i>	A comma-separated list of parent folder ids.
ChildIds	<i>String</i>	A semicolon-separated list of child resource ids.
ChildLinks	<i>String</i>	A semicolon-separated list of child resource links.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Query	<i>String</i>	This field accepts a valid Google Drive SDK query, which overrides conditionals in the WHERE clause.

Photos

Query the photos contained in a user's Google Drive.

Select

It is also possible to get all the photos from a Team Drive.

Note: You must set the connection property TeamDriveSupport to 'true', in order to query from a specific Team Drive.

```
SELECT * FROM Photos WHERE TeamDriveId='0ACkq0ZiV0yJCUk9PVA'
```

Columns

Name	Type	Description
Id [KEY]	<i>String</i>	The ID of the file.
Name	<i>String</i>	The name of the file. This is not necessarily unique within a folder. Note that for immutable items such as the top level folders of Team Drives, My Drive root folder, and Application Data folder the name is constant.
TeamDriveId	<i>String</i>	The Id of the teamDrive.
Description	<i>String</i>	A short description of the file or folder.
Extension	<i>String</i>	The extension of the file.
CreatedTime	<i>Datetime</i>	The creation date of the file or folder.
ModifiedTime	<i>Datetime</i>	The last modified date of the file or folder.
Size	<i>Long</i>	The size of the file in bytes.

OwnerName	<i>String</i>	The name of the resource's owner.
OwnerEmail	<i>String</i>	The email of the resource's owner.
Starred	<i>Boolean</i>	This field sets whether or not the resource is starred.
Trashed	<i>Boolean</i>	This field sets whether or not the resource has been moved to the trash.
Viewed	<i>Boolean</i>	This field sets whether or not the resource has been viewed by the current user.
ParentIds	<i>String</i>	A comma-separated list of parent folder Ids.
ChildIds	<i>String</i>	A semicolon-separated list of child resource Ids.
ChildLinks	<i>String</i>	A semicolon-separated list of child resource links.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Query	<i>String</i>	This field accepts a valid Google Drive SDK query, which overrides conditionals in the WHERE clause.

Sheets

Query the Google Sheets contained in a user's Google Drive.

Select

It is also possible to get all the sheets from a Team Drive.

Note: You must set the connection property TeamDriveSupport to 'true', in order to query from a specific Team Drive.

```
SELECT * FROM Sheets WHERE TeamDriveId='0ACkq0ZiV0yJCUk9PVA'
```

Columns

Name	Type	Description
Id [KEY]	String	The ID of the file.
Name	String	The name of the file. This is not necessarily unique within a folder. Note that for immutable items such as the top level folders of Team Drives, My Drive root folder, and Application Data folder the name is constant.
TeamDriveId	String	The Id of the teamDrive.
Description	String	A short description of the file or folder.
Extension	String	The extension of the file.
CreatedTime	Datetime	The creation date of the file or folder.
ModifiedTime	Datetime	The last modified date of the file or folder.

Size	<i>Long</i>	The size of the file in bytes.
OwnerName	<i>String</i>	The name of the resource's owner.
OwnerEmail	<i>String</i>	The email of the resource's owner.
Starred	<i>Boolean</i>	This field sets whether or not the resource is starred.
Trashed	<i>Boolean</i>	This field sets whether or not the resource has been moved to the trash.
Viewed	<i>Boolean</i>	This field sets whether or not the resource has been viewed by the current user.
ParentIds	<i>String</i>	A comma-separated list of parent folder Ids.
ChildIds	<i>String</i>	A semicolon-separated list of child resource Ids.
ChildLinks	<i>String</i>	A semicolon-separated list of child resource links.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description

Query	<i>String</i>	This field accepts a valid Google Drive SDK query, which overrides conditionals in the WHERE clause.
-------	---------------	--

Videos

Query the Google Videos contained in a user's Google Drive.

Select

It is also possible to get all the videos from a Team Drive.

Note: You must set the connection property TeamDriveSupport to 'true', in order to query from a specific Team Drive.

```
SELECT * FROM Videos WHERE TeamDriveId='0ACkq0ZiV0yJCuk9PVA'
```

Columns

Name	Type	Description
Id [KEY]	<i>String</i>	The ID of the file.
Name	<i>String</i>	The name of the file. This is not necessarily unique within a folder. Note that for immutable items such as the top level folders of Team Drives, My Drive root folder, and Application Data folder the name is constant.
TeamDriveId	<i>String</i>	The Id of the teamDrive.
Description	<i>String</i>	A short description of the file or folder.

Extension	<i>String</i>	The extension of the file.
CreatedTime	<i>Datetime</i>	The creation date of the file or folder.
ModifiedTime	<i>Datetime</i>	The last modified date of the file or folder.
Size	<i>Long</i>	The size of the file in bytes.
OwnerName	<i>String</i>	The name of the resource's owner.
OwnerEmail	<i>String</i>	The email of the resource's owner.
Starred	<i>Boolean</i>	This field sets whether or not the resource is starred.
Trashed	<i>Boolean</i>	This field sets whether or not the resource has been moved to the trash.
Viewed	<i>Boolean</i>	This field sets whether or not the resource has been viewed by the current user.
ParentIds	<i>String</i>	A comma-separated list of parent folder Ids.
ChildIds	<i>String</i>	A semicolon-separated list of child resource Ids.
ChildLinks	<i>String</i>	A semicolon-separated list of child resource links.

Pseudo-Columns

Pseudo column fields are used in the WHERE clause of SELECT statements and offer a more granular control over the tuples that are returned from the data source.

Name	Type	Description
Query	<i>String</i>	This field accepts a valid Google Drive SDK query, which overrides conditionals in the WHERE clause.

Stored Procedures

Stored procedures are available to complement the data available from the [Data Model](#). It may be necessary to update data available from a view using a stored procedure because the data does not provide for direct, table-like, two-way updates. In these situations, the retrieval of the data is done using the appropriate view or table, while the update is done by calling a stored procedure. Stored procedures take a list of parameters and return back a dataset that contains the collection of tuples that constitute the response.

Google Drive Adapter Stored Procedures

Name	Description
CreateFolder	Creates a folder in the user's Google Drive.
DownloadFile	Downloads a file from the user's Google Drive.
DownloadFileStorage	Downloads a file from the Google Storage.
EmptyTrash	Empties the user's trash.
GetOAuthAccessToken	Obtains the OAuth access token to be used for authentication with various Google services.
GetOAuthAuthorizationURL	Obtains the OAuth authorization URL used for authentication with various Google services.

RefreshOAuthAccessToken	Obtains the OAuth access token to be used for authentication with various Google services.
UploadFile	Uploads a file to the user's Google Drive.
UploadFileStorage	Uploads a file to the Google Storage.

CreateFolder

Creates a folder in the user's Google Drive.

Input

Name	Type	Description
Name	<i>String</i>	The title for the folder.
Description	<i>String</i>	The description for the folder.
Starred	<i>String</i>	This parameter sets whether or not the resource is starred. The allowed values are <i>TRUE</i> , <i>FALSE</i> . The default value is <i>FALSE</i> .
ParentIds	<i>String</i>	The comma-separated Ids of the parent folders for the new folder.

Result Set Columns

Name	Type	Description
------	------	-------------

Success	<i>String</i>	This parameter sets whether the operation was successful or not.
Id	<i>String</i>	The ID of the new folder.

DownloadFile

Downloads a file from the user's Google Drive.

Input

Name	Type	Description
Id	<i>String</i>	The Id of the resource to be downloaded.
LocalFile	<i>String</i>	The local file path including the file name for the location where the file will be saved on disk. Leave empty to keep the file in memory.
Encoding	<i>String</i>	<p>If the LocalFile input is left empty, the data will be output to FileData in the specified encoding.</p> <p>The allowed values are <i>NONE</i>, <i>BASE64</i>.</p> <p>The default value is <i>BASE64</i>.</p>
Overwrite	<i>String</i>	<p>What to do when downloaded file exists. Set true to overwrite.</p> <p>The allowed values are <i>true</i>, <i>false</i>.</p> <p>The default value is <i>false</i>.</p>
FileFormat	<i>String</i>	Used for converting the downloaded file to a different file type. Files may only be converted up to 10MB. Leave this blank to download the file in the same format it is stored in Google. The possible values are application/vnd.openxmlformats-officedocument.wordprocessingml.document,

application/vnd.oasis.opendocument.text, application/rtf, text/html, text/plain, and application/pdf.

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	If the LocalFile input is empty, file data will be output in the format specified by the Encoding input.
Success	<i>String</i>	This value shows a boolean indication of whether the operation was successful or not.

DownloadFileStorage

Downloads a file from the Google Storage.

Input

Name	Type	Description
Bucket	<i>String</i>	Name of the bucket from which to get the object.
ProjectId	<i>String</i>	The ID of the project you want to use.
Name	<i>String</i>	The name of the resource to be downloaded.
LocalFile	<i>String</i>	The local file path including the file name for the location where the file will be saved on disk. Leave empty to keep the file in memory.

Encoding	<i>String</i>	<p>If the LocalFile input is left empty, the data will be output to FileData in the specified encoding.</p> <p>The allowed values are <i>NONE</i>, <i>BASE64</i>.</p> <p>The default value is <i>BASE64</i>.</p>
Overwrite	<i>String</i>	<p>What to do when downloaded file exists. Set true to overwrite.</p> <p>The allowed values are <i>true</i>, <i>false</i>.</p> <p>The default value is <i>false</i>.</p>

Result Set Columns

Name	Type	Description
FileData	<i>String</i>	If the LocalFile input is empty, file data will be output in the format specified by the Encoding input.
Success	<i>String</i>	This value shows a boolean indication of whether the operation was successful or not.

EmptyTrash

Empties the user's trash.

Result Set Columns

Name	Type	Description
Success	<i>String</i>	This parameter sets whether the operation was successful or not.

GetOAuthAccessToken

Obtains the OAuth access token to be used for authentication with various Google services.

Input

Name	Type	Description
AuthMode	String	The type of authentication mode to use. The allowed values are <i>APP</i> , <i>WEB</i> . The default value is <i>WEB</i> .
Verifier	String	The verifier code returned by Google after permission for the app to connect has been granted. <i>WEB</i> AuthMode only.
Scope	String	The scope of access to Google APIs. By default, access to all APIs used by this data provider will be specified. The default value is <i>https://www.googleapis.com/auth/drive</i> <i>https://www.googleapis.com/auth/devstorage.read_write</i> .
CallbackURL	String	This field determines where the response is sent. The value of this parameter must exactly match one of the values registered in the APIs Console, including the HTTP or HTTPS schemes, capitalization, and trailing forward slash ('/').
Prompt	String	This field indicates the prompt to present the user. The default is <i>CONSENT</i> , so a given user will see a consent page every time, even if they have previously given consent to the application for a given set of scopes. If it is set to <i>SELECT_ACCOUNT</i> , the user will be prompted to select the account to connect to. Lastly, if it is set to <i>NONE</i> , no authentication or consent screens will be displayed to the user. The allowed values are <i>NONE</i> , <i>CONSENT</i> , <i>SELECT_ACCOUNT</i> . The default value is <i>CONSENT</i> .

AccessType	String	<p>This field indicates if your application needs to access a Google API when the user is not present at the browser. This parameter defaults to OFFLINE. If your application needs to refresh access tokens when the user is not present at the browser, then use OFFLINE. This will result in your application obtaining a refresh token the first time your application exchanges an authorization code for a user.</p> <p>The allowed values are <i>ONLINE</i>, <i>OFFLINE</i>.</p> <p>The default value is <i>OFFLINE</i>.</p>
State	String	<p>This field indicates any state that may be useful to your application upon receipt of the response. Your application receives the same value it sent, as this parameter makes a round-trip to Google authorization server and back. Uses include redirecting the user to the correct resource in your site, using nonces, and mitigating cross-site request forgery.</p>

Result Set Columns

Name	Type	Description
OAuthAccessToken	String	The authentication token returned from Google. This can be used in subsequent calls to other operations for this particular service.
OAuthRefreshToken	String	A token that may be used to obtain a new access token.
ExpiresIn	String	The remaining lifetime on the access token.

GetOAuthAuthorizationURL

Obtains the OAuth authorization URL used for authentication with various Google services.

Input

Name	Type	Description
Scope	String	<p>The scope of access to Google APIs. By default, access to all APIs used by this data provider will be specified.</p> <p>The default value is <code>https://www.googleapis.com/auth/drive</code> <code>https://www.googleapis.com/auth/devstorage.read_write</code>.</p>
CallbackURL	String	<p>This field determines where the response is sent. The value of this parameter must exactly match one of the values registered in the APIs Console, including the HTTP or HTTPS schemes, case, and trailing forward slash ('/').</p>
Prompt	String	<p>This field indicates the prompt to present the user. The default is <code>CONSENT</code>, so a given user will see a consent page every time, even if they have previously given consent to the application for a given set of scopes. If it is set to <code>SELECT_ACCOUNT</code>, the user will be prompted to select the account to connect to. Lastly, if it is set to <code>NONE</code>, no authentication or consent screens will be displayed to the user.</p> <p>The allowed values are <code>NONE</code>, <code>CONSENT</code>, <code>SELECT_ACCOUNT</code>.</p> <p>The default value is <code>CONSENT</code>.</p>
AccessType	String	<p>This field indicates if your application needs to access a Google API when the user is not present at the browser. This parameter defaults to <code>OFFLINE</code>. If your application needs to refresh access tokens when the user is not present at the browser, then use <code>OFFLINE</code>. This will result in your application obtaining a refresh token the first time your application exchanges an authorization code for a user.</p> <p>The allowed values are <code>ONLINE</code>, <code>OFFLINE</code>.</p> <p>The default value is <code>OFFLINE</code>.</p>
State	String	<p>This field indicates any state that may be useful to your application upon receipt of the response. Your application receives the same value it sent, as this parameter makes a</p>

round-trip to the Google authorization server and back. Possible uses include redirecting the user to the correct resource in your site, using nonces, and mitigating cross-site request forgery.

Result Set Columns

Name	Type	Description
URL	<i>String</i>	The URL to complete user authentication.

RefreshOAuthAccessToken

Obtains the OAuth access token to be used for authentication with various Google services.

Input

Name	Type	Description
OAuthRefreshToken	<i>String</i>	The refresh token returned from the original authorization code exchange.

Result Set Columns

Name	Type	Description

OAuthAccessToken	<i>String</i>	The authentication token returned from Google. This can be used in subsequent calls to other operations for this particular service.
OAuthRefreshToken	<i>String</i>	The authentication token returned from Google. This can be used in subsequent calls to other operations for this particular service.
ExpiresIn	<i>String</i>	The remaining lifetime on the access token.

UploadFile

Uploads a file to the user's Google Drive.

Input

Name	Type	Description
Id	<i>String</i>	The id for the file. Only needs to be set when updating an existing document.
Name	<i>String</i>	The title for the file, including the extension.
Description	<i>String</i>	The description for the file.
Starred	<i>String</i>	This parameter sets whether or not the resource is starred. The allowed values are <i>TRUE</i> , <i>FALSE</i> . The default value is <i>FALSE</i> .
Trashed	<i>String</i>	This field sets whether or not the resource has been moved to the trash. The default value is <i>FALSE</i> .

Viewed	<i>String</i>	This field sets whether or not the resource has been viewed by the current user. The default value is <i>FALSE</i> .
MIMETYPE	<i>String</i>	This parameter explicitly sets the MIME type for the document. If left empty, it will be determined automatically.
ParentIds	<i>String</i>	The comma-separated Ids of the parent folders for the uploaded document.
LocalFile	<i>String</i>	The local file path including the file name of the file to be uploaded. A value for this field is required when FileData is not specified.
FileData	<i>String</i>	If the LocalFile input is empty, the file data will be output to a file in the format specified by the Encoding parameter.
Encoding	<i>String</i>	The FileData input encoding type. The allowed values are <i>NONE</i> , <i>BASE64</i> . The default value is <i>BASE64</i> .

Result Set Columns

Name	Type	Description
Id	<i>String</i>	The id for the file which was uploaded or updated.
Success	<i>String</i>	This parameter sets whether the operation was successful or not.

UploadFileStorage

Uploads a file to the Google Storage.

Input

Name	Type	Description
Bucket	<i>String</i>	Name of the bucket in which to store the new object.
ProjectId	<i>String</i>	API project identifier.
Name	<i>String</i>	The object's name that will be saved in Storage. If it is used with LocalFile, it will overwrite the name of the original file, if it is used with FileData then it is mandatory.
Acl	<i>String</i>	<p>Apply a predefined set of access controls to this object.</p> <p>The allowed values are <i>PRIVATE</i>, <i>PUBLIC_READ</i>, <i>PUBLIC_READ_WRITE</i>, <i>AUTHENTICATED_READ</i>, <i>BUCKET_OWNER_READ</i>, <i>BUCKET_OWNER_FULL_CONTROL</i>.</p> <p>The default value is <i>PRIVATE</i>.</p>
MIMETYPE	<i>String</i>	This parameter explicitly sets the MIME type for the document. If left empty, it will be determined automatically.
ContentDisposition	<i>String</i>	A request and response header that specifies presentational information about the data being transmitted.
LocalFile	<i>String</i>	The local file path including the file name of the file to be uploaded. A value for this field is required when FileData is not specified.
FileData	<i>String</i>	If the LocalFile input is empty, the file data will be output to a file in the format specified by the Encoding parameter.
Encoding	<i>String</i>	The FileData input encoding type.

The allowed values are *NONE*, *BASE64*.

The default value is *BASE64*.

Result Set Columns

Name	Type	Description
Success	String	This parameter sets whether the operation was successful or not.

Connection String Options

The connection string properties are the various options that can be used to establish a connection. This section provides a complete list of the options you can configure in the connection string for this provider. Click the links for further details.

Auth Scheme	The type of authentication to use when connecting to Google Drive.
Firewall Password	A password used to authenticate to a proxy-based firewall.
Firewall Port	The TCP port for a proxy-based firewall.
Firewall Server	The name or IP address of a proxy-based firewall.
Firewall Type	The protocol used by a proxy-based firewall.
Firewall User	The user name to use to authenticate with a proxy-based firewall.
Initiate OAuth	Set this property to initiate the process to obtain or refresh the OAuth access

	token when you connect.
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
Log Modules	Core modules to be included in the log file.
Max Rows	Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.
OAuth Access Token	The access token for connecting using OAuth.
OAuth Client Id	The client ID assigned when you register your application with an OAuth authorization server.
OAuth Client Secret	The client secret assigned when you register your application with an OAuth authorization server.
OAuth Expires In	The lifetime in seconds of the OAuth AccessToken.
OAuth JWT Cert	The JWT Certificate store.
OAuth JWT Cert Password	The password for the OAuth JWT certificate.
OAuth JWT Cert Subject	The subject of the OAuth JWT certificate.
OAuth JWT Cert Type	The type of key store containing the JWT Certificate.
OAuth JWT Issuer	The issuer of the Java Web Token.
OAuth JWT Subject	The user subject for which the application is requesting delegated access.

OAuth Refresh Token	The OAuth refresh token for the corresponding OAuth access token.
OAuth Settings Location	The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.
OAuth Token Timestamp	The Unix epoch timestamp in milliseconds when the current Access Token was created.
OAuth Verifier	The verifier code returned from the OAuth authorization URL.
Other	These hidden properties are used only in specific use cases.
Proxy Auth Scheme	The authentication type to use to authenticate to the ProxyServer proxy.
Proxy Auto Detect	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
Proxy Exceptions	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .
Proxy Password	A password to be used to authenticate to the ProxyServer proxy.
Proxy Port	The TCP port the ProxyServer proxy is running on.
Proxy Server	The hostname or IP address of a proxy to route HTTP traffic through.
Proxy SSL Type	The SSL type to use when connecting to the ProxyServer proxy.
Proxy User	A user name to be used to authenticate to the ProxyServer proxy.
Readonly	You can use this property to enforce read-only access to Google Drive from the provider.

SSL Server Cert	The certificate to be accepted from the server when connecting using TLS/SSL.
Team Drive Support	Determines whether or not to enable Team Drive support.
Timeout	The value in seconds until the timeout error is thrown, canceling the operation.

Auth Scheme

The type of authentication to use when connecting to Google Drive.

Data Type

string

Default Value

"Auto"

Remarks

- Auto: Lets the driver decide automatically based on the other connection properties you have set.
- OAuth: Set this to perform OAuth authentication using a standard user account.
- OAuthJWT: Set this to perform OAuth authentication using an OAuth service account.
- GCPIstanceAccount: Set this to get Access Token from Google Cloud Platform instance.

Firewall Password

A password used to authenticate to a proxy-based firewall.

Data Type

string

Default Value

""

Remarks

This property is passed to the proxy specified by [FirewallServer](#) and [FirewallPort](#), following the authentication method specified by [FirewallType](#).

Firewall Port

The TCP port for a proxy-based firewall.

Data Type

int

Default Value

0

Remarks

This specifies the TCP port for a proxy allowing traversal of a firewall. Use [FirewallServer](#) to specify the name or IP address. Specify the protocol with [FirewallType](#).

Firewall Server

The name or IP address of a proxy-based firewall.

Data Type

string

Default Value

""

Remarks

This property specifies the IP address, DNS name, or host name of a proxy allowing traversal of a firewall. The protocol is specified by [FirewallType](#): Use [FirewallServer](#) with this property to connect through SOCKS or do tunneling. Use [ProxyServer](#) to connect to an HTTP proxy.

Note that the adapter uses the system proxy by default. To use a different proxy, set [ProxyAutoDetect](#) to false.

Firewall Type

The protocol used by a proxy-based firewall.

Data Type

string

Default Value

"NONE"

Remarks

This property specifies the protocol that the adapter will use to tunnel traffic through the [FirewallServer](#) proxy. Note that by default, the adapter connects to the system proxy; to disable this behavior and connect to one of the following proxy types, set [ProxyAutoDetect](#) to false.

Type	Default Port	Description
TUNNEL	80	When this is set, the adapter opens a connection to Google Drive and traffic flows back and forth through the proxy.
SOCKS4	1080	When this is set, the adapter sends data through the SOCKS 4 proxy specified by FirewallServer and FirewallPort and passes the FirewallUser value to the proxy, which determines if the connection request should be granted.
SOCKS5	1080	When this is set, the adapter sends data through the SOCKS 5 proxy specified by FirewallServer and FirewallPort . If your proxy requires authentication, set FirewallUser and FirewallPassword to credentials the proxy recognizes.

To connect to HTTP proxies, use [ProxyServer](#) and [ProxyPort](#). To authenticate to HTTP proxies, use [ProxyAuthScheme](#), [ProxyUser](#), and [ProxyPassword](#).

Firewall User

The user name to use to authenticate with a proxy-based firewall.

Data Type

string

Default Value

""

Remarks

The [FirewallUser](#) and [FirewallPassword](#) properties are used to authenticate against the proxy specified in [FirewallServer](#) and [FirewallPort](#), following the authentication method specified in [FirewallType](#).

Initiate OAuth

Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.

Data Type

string

Default Value

"OFF"

Remarks

The following options are available:

1. **OFF**: Indicates that the OAuth flow will be handled entirely by the user. An OAuthAccessToken will be required to authenticate.
2. **GETANDREFRESH**: Indicates that the entire OAuth Flow will be handled by the adapter. If no token currently exists, it will be obtained by prompting the user via the browser. If a token exists, it will be refreshed when applicable.
3. **REFRESH**: Indicates that the adapter will only handle refreshing the OAuthAccessToken. The user will never be prompted by the adapter to authenticate via the browser. The user must handle obtaining the OAuthAccessToken and OAuthRefreshToken initially.

Location

A path to the directory that contains the schema files defining tables, views, and stored procedures.

Data Type

string

Default Value

"%APPDATA%\\CData\\GoogleDrive Data Provider\\Schema"

Remarks

The path to a directory which contains the schema files for the adapter (.rsd files for tables and views, .rsb files for stored procedures). The folder location can be a relative path from the location of the executable. The Location property is only needed if you want to customize definitions (for example, change a column name, ignore a column, and so on) or extend the data model with new tables, views, or stored procedures.

If left unspecified, the default location is "%APPDATA%\\CData\\GoogleDrive Data Provider\\Schema" with %**APPDATA**% being set to the user's configuration directory:

Platform	%APPDATA%
Windows	The value of the APPDATA environment variable
Mac	~/Library/Application Support
Linux	~/.config

Log Modules

Core modules to be included in the log file.

Data Type

string

Default Value

""

Remarks

Only the modules specified (separated by ';') will be included in the log file. By default all modules are included.

Max Rows

Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

Data Type

int

Default Value

-1

Remarks

Limits the number of rows returned rows when no aggregation or group by is used in the query. This helps avoid performance issues at design time.

OAuth Access Token

The access token for connecting using OAuth.

Data Type

string

Default Value

""

Remarks

The OAuthAccessToken property is used to connect using OAuth. The OAuthAccessToken is retrieved from the OAuth server as part of the authentication process. It has a server-dependent timeout and can be reused between requests.

The access token is used in place of your user name and password. The access token protects your credentials by keeping them on the server.

OAuth Client Id

The client ID assigned when you register your application with an OAuth authorization server.

Data Type

string

Default Value

""

Remarks

As part of registering an OAuth application, you will receive the OAuthClientId value, sometimes also called a consumer key, and a client secret, the [OAuthClientSecret](#).

OAuth Client Secret

The client secret assigned when you register your application with an OAuth authorization server.

Data Type

string

Default Value

""

Remarks

As part of registering an OAuth application, you will receive the [OAuthClientId](#), also called a consumer key. You will also receive a client secret, also called a consumer secret. Set the client secret in the [OAuthClientSecret](#) property.

OAuth Expires In

The lifetime in seconds of the OAuth AccessToken.

Data Type

string

Default Value

""

Remarks

Pair with OAuthTokenTimestamp to determine when the AccessToken will expire.

OAuth JWT Cert

The JWT Certificate store.

Data Type

string

Default Value

""

Remarks

The name of the certificate store for the client certificate.

The [OAuthJWTCertType](#) field specifies the type of the certificate store specified by [OAuthJWTCert](#). If the store is password protected, specify the password in [OAuthJWTCertPassword](#).

[OAuthJWTCert](#) is used in conjunction with the [OAuthJWTCertSubject](#) field in order to specify client certificates. If [OAuthJWTCert](#) has a value, and [OAuthJWTCertSubject](#) is set, a search for a certificate is initiated. Please refer to the [OAuthJWTCertSubject](#) field for details.

Designations of certificate stores are platform-dependent.

The following are designations of the most common User and Machine certificate stores in Windows:

MY	A certificate store holding personal certificates with their associated private keys.
CA	Certifying authority certificates.
ROOT	Root certificates.
SPC	Software publisher certificates.

In Java, the certificate store normally is a file containing certificates and optional private keys.

When the certificate store type is `PFXFile`, this property must be set to the name of the file. When the type is `PFXBlob`, the property must be set to the binary contents of a PFX file (i.e. PKCS12 certificate store).

OAuth JWT Cert Password

The password for the OAuth JWT certificate.

Data Type

string

Default Value

""

Remarks

If the certificate store is of a type that requires a password, this property is used to specify that password in order to open the certificate store.

This is not required when using the GOOGLEJSON [OAuthJWTCertType](#). Google JSON keys are not encrypted.

OAuth JWT Cert Subject

The subject of the OAuth JWT certificate.

Data Type

string

Default Value

"*"

Remarks

When loading a certificate the subject is used to locate the certificate in the store.

If an exact match is not found, the store is searched for subjects containing the value of the property.

If a match is still not found, the property is set to an empty string, and no certificate is selected.

The special value "*" picks the first certificate in the certificate store.

The certificate subject is a comma separated list of distinguished name fields and values. For instance "CN=www.server.com, OU=test, C=US, E=support@cdata.com". Common fields and their meanings are displayed below.

Field	Meaning
CN	Common Name. This is commonly a host name like www.server.com.
O	Organization
OU	Organizational Unit
L	Locality
S	State
C	Country
E	Email Address

If a field value contains a comma it must be quoted.

OAuth JWT Cert Type

The type of key store containing the JWT Certificate.

Data Type

string

Default Value

"USER"

Remarks

This property can take one of the following values:

USER - default	For Windows, this specifies that the certificate store is a certificate store owned by the current user. <i>Note:</i> This store type is not available in Java.
MACHINE	For Windows, this specifies that the certificate store is a machine store. <i>Note:</i> this store type is not available in Java.
PFXFILE	The certificate store is the name of a PFX (PKCS12) file containing certificates.
PFXBLOB	The certificate store is a string (base-64-encoded) representing a certificate store in PFX (PKCS12) format.
JKSFILE	The certificate store is the name of a Java key store (JKS) file containing certificates. <i>Note:</i> this store type is only available in Java.
JKSBLOB	The certificate store is a string (base-64-encoded) representing a certificate store in Java key store (JKS) format. <i>Note:</i> this store type is only available in Java.
PEMKEY_FILE	The certificate store is the name of a PEM-encoded file that contains a private key and an optional certificate.
PEMKEY_BLOB	The certificate store is a string (base64-encoded) that contains a private key and an optional certificate.
PUBLIC_KEY_FILE	The certificate store is the name of a file that contains a PEM- or DER-encoded public key certificate.
PUBLIC_KEY_BLOB	The certificate store is a string (base-64-encoded) that contains a PEM- or DER-encoded public key certificate.
SSHPUBLIC_KEY_FILE	The certificate store is the name of a file that contains an SSH-style public key.
SSHPUBLIC_KEY_BLOB	The certificate store is a string (base-64-encoded) that contains an SSH-style public key.
P7BFILE	The certificate store is the name of a PKCS7 file containing

	certificates.
PPKFILE	The certificate store is the name of a file that contains a PPK (PuTTY Private Key).
XMLFILE	The certificate store is the name of a file that contains a certificate in XML format.
XMLBLOB	The certificate store is a string that contains a certificate in XML format.
GOOGLEJSON	The certificate store is the name of a JSON file containing the service account information. Only valid when connecting to a Google service.

OAuth JWT Issuer

The issuer of the Java Web Token.

Data Type

string

Default Value

""

Remarks

The issuer of the Java Web Token. This is typically either the Client ID or Email Address of the OAuth Application.

This is not required when using the GOOGLEJSON [OAuthJWTCertType](#). Google JSON keys contain a copy of the issuer account.

OAuth JWT Subject

The user subject for which the application is requesting delegated access.

Data Type

string

Default Value

""

Remarks

The user subject for which the application is requesting delegated access. Typically, the user account name or email address.

OAuth Refresh Token

The OAuth refresh token for the corresponding OAuth access token.

Data Type

string

Default Value

""

Remarks

The OAuthRefreshToken property is used to refresh the [OAuthAccessToken](#) when using OAuth authentication.

OAuth Settings Location

The location of the settings file where OAuth values are saved when InitiateOAuth is set to GETANDREFRESH or REFRESH. Alternatively, this can be held in memory by specifying a value starting with memory://.

Data Type

string

Default Value

"%APPDATA%\\CData\\GoogleDrive Data Provider\\OAuthSettings.txt"

Remarks

When [InitiateOAuth](#) is set to GETANDREFRESH or REFRESH, the adapter saves OAuth values to avoid requiring the user to manually enter OAuth connection properties and allowing the credentials to be shared across connections or processes.

Alternatively to specifying a file path, memory storage can be used instead. Memory locations are specified by using a value starting with 'memory://' followed by a unique identifier for that set of credentials (ex: memory://user1). The identifier can be anything you choose but should be unique to the user. Unlike with the file based storage, you must manually store the credentials when closing the connection with memory storage to be able to set them in the connection when the process is started again. The OAuth property values can be retrieved with a query to the sys_connection_props system table. If there are multiple connections using the same credentials, the properties should be read from the last connection to be closed.

If left unspecified, the default location is "%APPDATA%\\CData\\GoogleDrive Data Provider\\OAuthSettings.txt" with %**APPDATA**% being set to the user's configuration directory:

Platform	%APPDATA%
Windows	The value of the APPDATA environment variable
Mac	~/Library/Application Support
Linux	~/.config

OAuth Token Timestamp

The Unix epoch timestamp in milliseconds when the current Access Token was created.

Data Type

string

Default Value

""

Remarks

Pair with OAuthExpiresIn to determine when the AccessToken will expire.

OAuth Verifier

The verifier code returned from the OAuth authorization URL.

Data Type

string

Default Value

""

Remarks

The verifier code returned from the OAuth authorization URL. This can be used on systems where a browser cannot be launched such as headless systems.

Authentication on Headless Machines

See [Getting Started](#) to obtain the [OAuthVerifier](#) value.

Set [OAuthSettingsLocation](#) along with [OAuthVerifier](#). When you connect, the adapter exchanges the [OAuthVerifier](#) for the OAuth authentication tokens and saves them, encrypted, to the specified file. Set [InitiateOAuth](#) to GETANDREFRESH automate the exchange.

Once the OAuth settings file has been generated, you can remove [OAuthVerifier](#) from the connection properties and connect with [OAuthSettingsLocation](#) set.

To automatically refresh the OAuth token values, set [OAuthSettingsLocation](#) and additionally set [InitiateOAuth](#) to REFRESH.

Other

These hidden properties are used only in specific use cases.

Data Type

string

Default Value

""

Remarks

The properties listed below are available for specific use cases. Normal driver use cases and functionality should not require these properties.

Specify multiple properties in a semicolon-separated list.

Integration and Formatting

DefaultColumnSize

Sets the default length of string fields when the data source

	does not provide column length in the metadata. The default value is 2000.
ConvertDateTimeToGMT	Determines whether to convert date-time values to GMT, instead of the local time of the machine.
RecordToFile=filename	Records the underlying socket data transfer to the specified file.

Proxy Auth Scheme

The authentication type to use to authenticate to the ProxyServer proxy.

Data Type

string

Default Value

"BASIC"

Remarks

This value specifies the authentication type to use to authenticate to the HTTP proxy specified by [ProxyServer](#) and [ProxyPort](#).

Note that the adapter will use the system proxy settings by default, without further configuration needed; if you want to connect to another proxy, you will need to set [ProxyAutoDetect](#) to false, in addition to [ProxyServer](#) and [ProxyPort](#). To authenticate, set [ProxyAuthScheme](#) and set [ProxyUser](#) and [ProxyPassword](#), if needed.

The authentication type can be one of the following:

- **BASIC:** The adapter performs HTTP BASIC authentication.
- **DIGEST:** The adapter performs HTTP DIGEST authentication.
- **NEGOTIATE:** The adapter retrieves an NTLM or Kerberos token based on the applicable protocol for authentication.

- **PROPRIETARY:** The adapter does not generate an NTLM or Kerberos token. You must supply this token in the Authorization header of the HTTP request.

If you need to use another authentication type, such as SOCKS 5 authentication, see [FirewallType](#).

Proxy Auto Detect

This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.

Data Type

bool

Default Value

true

Remarks

This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.

NOTE: When this property is set to True, the proxy used is determined as follows:

- A search from the JVM properties (**http.proxy**, **https.proxy**, **socksProxy**, etc.) is performed.
- In the case that the JVM properties don't exist, a search from **java.home/lib/net.properties** is performed.
- In the case that java.net.useSystemProxies is set to True, a search from **the SystemProxy** is performed.
- In Windows only, an attempt is made to retrieve these properties from the **Internet Options** in the **registry**.

To connect to an HTTP proxy, see [ProxyServer](#). For other proxies, such as SOCKS or tunneling, see [FirewallType](#).

Proxy Exceptions

A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

Data Type

string

Default Value

""

Remarks

The [ProxyServer](#) is used for all addresses, except for addresses defined in this property. Use semicolons to separate entries.

Note that the adapter uses the system proxy settings by default, without further configuration needed; if you want to explicitly configure proxy exceptions for this connection, you need to set [ProxyAutoDetect](#) = false, and configure [ProxyServer](#) and [ProxyPort](#). To authenticate, set [ProxyAuthScheme](#) and set [ProxyUser](#) and [ProxyPassword](#), if needed.

Proxy Password

A password to be used to authenticate to the ProxyServer proxy.

Data Type

string

Default Value

""

Remarks

This property is used to authenticate to an HTTP proxy server that supports NTLM (Windows), Kerberos, or HTTP authentication. To specify the HTTP proxy, you can set [ProxyServer](#) and [ProxyPort](#). To specify the authentication type, set [ProxyAuthScheme](#).

If you are using HTTP authentication, additionally set [ProxyUser](#) and [ProxyPassword](#) to HTTP proxy.

If you are using NTLM authentication, set [ProxyUser](#) and [ProxyPassword](#) to your Windows password. You may also need these to complete Kerberos authentication.

For SOCKS 5 authentication or tunneling, see [FirewallType](#).

By default, the adapter uses the system proxy. If you want to connect to another proxy, set [ProxyAutoDetect](#) to false.

Proxy Port

The TCP port the ProxyServer proxy is running on.

Data Type

int

Default Value

80

Remarks

The port the HTTP proxy is running on that you want to redirect HTTP traffic through. Specify the HTTP proxy in [ProxyServer](#). For other proxy types, see [FirewallType](#).

Proxy Server

The hostname or IP address of a proxy to route HTTP traffic through.

Data Type

string

Default Value

""

Remarks

The hostname or IP address of a proxy to route HTTP traffic through. The adapter can use the HTTP, Windows (NTLM), or Kerberos authentication types to authenticate to an HTTP proxy.

If you need to connect through a SOCKS proxy or tunnel the connection, see [FirewallType](#).

By default, the adapter uses the system proxy. If you need to use another proxy, set [ProxyAutoDetect](#) to false.

Proxy SSL Type

The SSL type to use when connecting to the ProxyServer proxy.

Data Type

string

Default Value

"AUTO"

Remarks

This property determines when to use SSL for the connection to an HTTP proxy specified by [ProxyServer](#). This value can be AUTO, ALWAYS, NEVER, or TUNNEL. The applicable values are the following:

AUTO	Default setting. If the URL is an HTTPS URL, the adapter will use the TUNNEL option. If the URL is an HTTP URL, the component will use the NEVER option.
ALWAYS	The connection is always SSL enabled.
NEVER	The connection is not SSL enabled.
TUNNEL	The connection is through a tunneling proxy. The proxy server opens a connection to the remote host and traffic flows back and forth through the proxy.

Proxy User

A user name to be used to authenticate to the ProxyServer proxy.

Data Type

string

Default Value

""

Remarks

The [ProxyUser](#) and [ProxyPassword](#) options are used to connect and authenticate against the HTTP proxy specified in [ProxyServer](#).

You can select one of the available authentication types in [ProxyAuthScheme](#). If you are using HTTP authentication, set this to the user name of a user recognized by the HTTP proxy. If you are using Windows or Kerberos authentication, set this property to a user name in one of the following formats:

```
user@domain  
domain\user
```

Readonly

You can use this property to enforce read-only access to Google Drive from the provider.

Data Type

bool

Default Value

false

Remarks

If this property is set to true, the adapter will allow only SELECT queries. INSERT, UPDATE, DELETE, and stored procedure queries will cause an error to be thrown.

SSL Server Cert

The certificate to be accepted from the server when connecting using TLS/SSL.

Data Type

string

Default Value

""

Remarks

If using a TLS/SSL connection, this property can be used to specify the TLS/SSL certificate to be accepted from the server. Any other certificate that is not trusted by the machine is rejected.

This property can take the following forms:

Description	Example
A full PEM Certificate (example shortened for brevity)	-----BEGIN CERTIFICATE----- MIICHTCCAe4CAQAwDQYJKoZIhvc... ...Qw== -----END CERTIFICATE-----
A path to a local file containing the certificate	C:\cert.cer
The public key (example shortened for brevity)	-----BEGIN RSA PUBLIC KEY----- MIGfMA0GCSq.....AQAB -----END RSA PUBLIC KEY-----
The MD5 Thumbprint (hex values can also be either space or colon separated)	e04b5e1529c58a1e9e09828d 70e4
The SHA1 Thumbprint (hex values can also be either space or colon separated)	34b29926a081b2ec14b4a3d904f 801cbb150d

If not specified, any certificate trusted by the machine is accepted.

Certificates are validated as trusted by the machine based on the System's trust store. The trust store used is the 'javax.net.ssl.trustStore' value specified for the system. If no value is specified for this property, Java's default trust store is used (for example, JAVA_HOME\lib\security\cacerts).

Use '*' to signify to accept all certificates. Note that this is not recommended due to security concerns.

Team Drive Support

Determines whether or not to enable Team Drive support.

Data Type

bool

Default Value

false

Remarks

If you set this property to 'true', you can query from a specific Team Drive using the TeamDriveld as a filter.

Timeout

The value in seconds until the timeout error is thrown, canceling the operation.

Data Type

int

Default Value

60

Remarks

If Timeout = 0, operations do not time out. The operations run until they complete successfully or until they encounter an error condition.

If Timeout expires and the operation is not yet complete, the adapter throws an exception.

TIBCO Product Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Data Virtualization](#) page.

- **Users**
 - TDV Getting Started Guide
 - TDV User Guide
 - TDV Web UI User Guide
 - TDV Client Interfaces Guide
 - TDV Tutorial Guide
 - TDV Northbay Example
- **Administration**
 - TDV Installation and Upgrade Guide
 - TDV Administration Guide
 - TDV Active Cluster Guide
 - TDV Security Features Guide
- **Data Sources**

TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

- **References**

TDV Reference Guide

TDV Application Programming Interface Guide

- **Other**

TDV Business Directory Guide

TDV Discovery Guide

- *TIBCO TDV and Business Directory Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Release Version Support

TDV 8.5 is designated as a Long Term Support (LTS) version. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also

https://docs.tibco.com/pub/tdv/general/LTS/tdv_LTS_releases.htm.

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, visit [TIBCO Community](#).

Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, TIBCO logo, Two-Second Advantage, TIBCO Spotfire, TIBCO ActiveSpaces, TIBCO Spotfire Developer, TIBCO EMS, TIBCO Spotfire Automation Services, TIBCO Enterprise Runtime for R, TIBCO Spotfire Server, TIBCO Spotfire Web Player, TIBCO Spotfire Statistics Services, S-PLUS, and TIBCO Spotfire S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2002-2023 Cloud Software Group, Inc All Rights Reserved.