



# **TIBCO® Data Virtualization**

## **Google Sheets Adapter Guide**

Version 8.7.0 | October 2023

# Contents

---

|                                     |          |
|-------------------------------------|----------|
| <b>Contents</b>                     | <b>2</b> |
| <b>Google Sheets Adapter</b>        | <b>5</b> |
| Google Sheets Version Support       | 5        |
| SQL Compliance                      | 5        |
| Getting Started                     | 5        |
| Connecting to Google Sheets         | 5        |
| Deploying the Google Sheets Adapter | 5        |
| Basic Tab                           | 6        |
| Logging                             | 8        |
| Using OAuth Authentication          | 9        |
| Changelog                           | 18       |
| Advanced Features                   | 21       |
| User Defined Views                  | 21       |
| SSL Configuration                   | 21       |
| Firewall and Proxy                  | 22       |
| Logging                             | 22       |
| User Defined Views                  | 22       |
| SSL Configuration                   | 25       |
| Firewall and Proxy                  | 25       |
| Logging                             | 26       |
| SQL Compliance                      | 28       |
| SELECT Statements                   | 29       |
| INSERT Statements                   | 29       |
| UPDATE Statements                   | 29       |
| DELETE Statements                   | 29       |
| EXECUTE Statements                  | 29       |
| Names and Quoting                   | 29       |

|                                    |    |
|------------------------------------|----|
| Transactions and Batching .....    | 30 |
| SELECT Statements .....            | 30 |
| SELECT INTO Statements .....       | 32 |
| INSERT Statements .....            | 32 |
| UPDATE Statements .....            | 33 |
| DELETE Statements .....            | 34 |
| EXECUTE Statements .....           | 35 |
| Using Spreadsheets as Tables ..... | 36 |
| Spreadsheets as Tables .....       | 36 |
| Tables .....                       | 36 |
| Stored Procedures .....            | 36 |
| Tables .....                       | 36 |
| Columns .....                      | 37 |
| Views .....                        | 37 |
| Stored Procedures .....            | 43 |
| Connection String Options .....    | 59 |
| Authentication .....               | 60 |
| Connection .....                   | 60 |
| OAuth .....                        | 61 |
| JWT OAuth .....                    | 62 |
| SSL .....                          | 62 |
| Firewall .....                     | 62 |
| Proxy .....                        | 63 |
| Logging .....                      | 63 |
| Schema .....                       | 64 |
| Miscellaneous .....                | 64 |
| Authentication .....               | 65 |
| Connection .....                   | 67 |
| OAuth .....                        | 72 |
| JWT OAuth .....                    | 79 |
| SSL .....                          | 85 |
| Firewall .....                     | 87 |

|   |            |
|---|------------|
| Proxy .....   | 90         |
| Logging .....   | 97         |
| Schema .....  | 98         |
| Miscellaneous .....   | 102        |
| <b>TIBCO Product Documentation and Support Services .....</b> | <b>114</b> |
| How to Access TIBCO Documentation .....                       | 114        |
| Release Version Support .....                                 | 115        |
| How to Contact TIBCO Support .....                            | 115        |
| How to Join TIBCO Community .....                             | 116        |
| <b>Legal and Third-Party Notices .....</b>                    | <b>117</b> |

# Google Sheets Adapter

---

## Google Sheets Version Support

The adapter leverages the Google Drive API to enable bidirectional access to Google Sheets.

## SQL Compliance

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

## Getting Started

## Connecting to Google Sheets

[Basic Tab](#) shows how to authenticate to Google Sheets 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 Sheets Adapter

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

1. Unzip the `tdv.googleheets.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.googleheets/tdv.googleheets.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 GoogleSheets
```

## Basic Tab

## Authenticating to Google Sheets

All connections to Google Sheets are authenticated using OAuth. The adapter supports using user accounts, service accounts and GCP instance accounts for authentication.

### Authenticate with a User Account

AuthScheme must be set to **OAuth** in all of the user account flows.

See [Using OAuth Authentication](#) for a authentication guide covering all the supported methods in detail.

### Authenticate with a Service Account

To authenticate using a service account, you must create a new service account and have a copy of the accounts certificate.

For a JSON file, you will need to set these properties:

- AuthScheme: Required. Set this to **OAuthJWT**.
- InitiateOAuth: Required. Set this to **GETANDREFRESH**.
- OAuthJWTCertType: Required. Set this to **GOOGLEJSON**.

- OAuthJWTCert: Required. Set this to the path to the .json file provided by Google.
- OAuthJWTSubject: Optional. Only set this value if the service account is part of a GSuite domain and you want to enable delegation. The value of this property should be the email address of the user whose data you want to access.

For a PFX file, you will need to set these properties instead:

- AuthScheme: Required. Set this to **OAuthJWT**.
- InitiateOAuth: Required. Set this to **GETANDREFRESH**.
- OAuthJWTCertType: Required. Set this to **PFXFILE**.
- OAuthJWTCert: Required. Set this to the path to the .pfx file provided by Google.
- OAuthJWTCertPassword: Optional. Set this to the .pfx file password. In most cases this will need to be provided since Google encrypts PFX certificates.
- OAuthJWTCertSubject: Optional. Set this only if you are using a OAuthJWTCertType which stores multiple certificates. Should not be set for PFX certificates generated by Google.
- OAuthJWTIssuer: Required. Set this to the email address of the service account. This address will usually include the domain **iam.gserviceaccount.com**.
- OAuthJWTSubject: Optional. Only set this value if the service account is part of a GSuite domain and you want to enable delegation. The value of this property should be the email address of the user whose data you want to access.

If you do not already have a service account, you can create one by following the procedure in [Creating a Custom OAuth App](#).

## Authenticate with a GCP Instance Account

When running on a GCP virtual machine, the adapter can authenticate using a service account tied to the virtual machine. To use this mode, set AuthScheme to **GCPInstanceAccount**.

## Authenticate using an API Key

To connect using API Keys, set the APIKey property, and set AuthScheme to **Token**.

You can create an API key in the Google Cloud Console by clicking Create credentials > API key. You can restrict the key before using it in production by clicking Restrict key and selecting one of the Restrictions.

## Logging

The adapter uses TDV Server's logging (log4j) to generate log files. The settings within the TDV Server's logging (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 Sheets 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\_server\_dsrc.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 Sheets. 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 Sheets

The user account flow requires the authenticating user to interact with Google Sheets 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 Sheets 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 Sheets OAuth endpoint to grant permissions to the adapter.

## Using a Service Account to Connect to Google Sheets

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 Sheets 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

You can use a custom OAuth app to authenticate with a service account or a user account. See [Using OAuth Authentication](#) for more information.

## 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 Sheets 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 Sheets 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 Sheets 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:

- OAuthClientId: Set this to the consumer key in your app settings.
- OAuthClientSecret: Set this to the consumer secret in your app settings.
- 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:

- OAuthClientId: Set this to the consumer key in your app settings.
- OAuthClientSecret: Set this to the consumer secret in your app settings.
- 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.
- OAuthClientId: Set this to the consumer key in your app settings.
- OAuthClientSecret: Set this to the consumer secret in your app settings.
- 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 Sheets 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.
- 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

You can use a custom OAuth app to authenticate a service account or a user account.

### Create an OAuth App for User Account Authentication

Follow the procedure below to register an app and obtain the OAuthClientId and OAuthClientSecret.

#### Create a Custom OAuth App: Desktop

1. Log into the Google API Console and open a project. Select the API Manager from the main menu.
2. In the user consent flow, click Credentials -> Create Credentials -> OAuth Client Id. Click Other. After creating the app, the OAuthClientId and OAuthClientSecret are displayed.
3. Click Library -> Spreadsheets API -> Enable API.
4. After going one step back: Click Library -> Spreadsheets API -> Enable API.

## Create an OAuth App for Service Account Authentication

Follow the steps below to create an OAuth application and generate a private key. You will then authorize the service account.

1. Log into the Google API Console and open a project. Select the API Manager from the main menu.
2. Click Create Credentials -> Service Account Key.
3. In the Service Account menu, select New Service Account or select an existing service account.
4. If you are creating a new service account, additionally select one or more roles. You can assign primitive roles at the project level in the IAM and Admin section; other roles enable you to further customize access to Google APIs.
5. In the Key Type section, select the P12 key type.
6. Create the app to download the key pair. The private key's password is displayed: Set this in OAuthJWTCertPassword.
7. In the service accounts section, click Manage Service Accounts and set OAuthJWTIssuer to the email address displayed in the service account Id field.
8. Click Library -> Spreadsheets API -> Enable API.
9. After going one step back: Click Library -> Drive API -> Enable API.

## Changelog

### General Changes

**[7915] - 2021-09-02**

#### Added

- Added support for the STRING\_SPLIT table-valued function in the CROSS APPLY clause.

## **[7889] - 2021-08-07**

### **Changed**

- Add the KeySeq column to the sys\_foreignkeys table.

## **[7888] - 2021-08-06**

### **Changed**

- Add the new sys\_primarykeys system table.

## **[7874] - 2021-07-23**

### **Changed**

- Updated the Literal Function Names for relative date/datetime functions. Previously relative date/datetime functions resolved to a different value when used in the projection vs te predicate. Ie: SELECT LAST\_MONTH() AS lm, Col FROM Table WHERE Col > LAST\_MONTH(). Formerly the two LAST\_MONTH() methods would resolve to different datetimes. Now they will match.
- As a replacement for the previous behavior, the relative date/datetime functions in the criteria may have an 'L' appended to them. Ie: WHERE col > L\_LAST\_MONTH(). This will continue to resolve to the same values that previously were calculated in the criteria. Note that the "L\_" prefix will only work in the predicate - it not available for the projection.

## **[7859] - 2021-07-08**

### **Added**

- Added the TCP Logging Module for the logging information happening on the TCP wire protocol. The transport bytes that are incoming and ongoing will be logged at verbosity=5.

## **[7785] - 2021-04-23**

### **Added**

- Added support for handling client side formulas during insert / update. For example: UPDATE Table SET Col1 = Concat(Col1, " - ", Col2) WHERE Col2 LIKE 'A%'

## **[7783] - 2021-04-23**

### **Changed**

- Updated how display sizes are determined for varchar primary key and foreign key columns so they will match the reported length of the column.

## **[7776] - 2021-04-16**

### **Added**

- Non-conditional updates between two columns is now available to all drivers. For example: UPDATE Table SET Col1=Col2

### **Changed**

- Reduced the length to 255 for varchar primary key and foreign key columns.
- Updated implicit and metadata caching to improve performance and support for multiple connections. Old metadata caches are not compatible - you would need to generate new metadata caches if you are currently using CacheMetadata.
- Updated index naming convention to avoid duplicates
- Updated and standardized Getting Started connection help.
- Added the Advanced Features section to the help of all drivers.
- Categorized connection property listings in the help for all editions.

# **Google Sheets Changes**

## **[8047] - 2022-01-12**

### **Changed**

- Changed the support for working with a custom Range in Select, Insert and UPDATE operations. Columns will instead be named appropriately based on the range instead of starting at A,B,C, etc. This only impacts tables when the Header connection property is set to false. For example: SELECT \* FROM Spreadsheet\_Sheet1#C1:F5. The column names will be: C, D, E, F. Previously you would get columns with names starting from A: A,B,C,D.

## [7839] - 2021-06-18

### Added

- Added support for the GOOGLEJSONBLOB JWT certificate type. This works like the existing GOOGLEJSON certificate type except that the certificate is provided as JSON text instead of as a file path.

## [7703] - 2021-02-02

### Added

- Added a new view: Folders. It can be used to query the folders contained in a user's Google Drive.
- Added new column ParentIds on the Spreadsheets view. It displays the comma-separated list of the parent folder Id's, if any.

## Advanced Features

This section details a selection of advanced features of the Google Sheets adapter.

## User Defined Views

The adapter allows you to define virtual tables, called *user defined views*, whose contents are decided by a pre-configured query. These views are useful when you cannot directly control queries being issued to the drivers. See [User Defined Views](#) for an overview of creating and configuring custom views.

## SSL Configuration

Use [SSL Configuration](#) to adjust how adapter handles TLS/SSL certificate negotiations. You can choose from various certificate formats; see the [SSLServerCert](#) property under "Connection String Options" for more information.

## Firewall and Proxy

Configure the adapter for compliance with [Firewall and Proxy](#), including Windows proxies and HTTP proxies. You can also set up tunnel connections.

## Logging

See [Logging](#) for an overview of configuration settings that can be used to refine CData logging. For basic logging, you only need to set two connection properties, but there are numerous features that support more refined logging, where you can select subsets of information to be logged using the [LogModules](#) connection property.

## User Defined Views

The Google Sheets Adapter allows you to define a virtual table whose contents are decided by a pre-configured query. These are called *User Defined Views*, which are useful in situations where you cannot directly control the query being issued to the driver, e.g. when using the driver from a tool. The User Defined Views can be used to define predicates that are always applied. If you specify additional predicates in the query to the view, they are combined with the query already defined as part of the view.

There are two ways to create user defined views:

- Create a JSON-formatted configuration file defining the views you want.
- DDL statements.

## Defining Views Using a Configuration File

User Defined Views are defined in a JSON-formatted configuration file called *UserDefinedViews.json*. The adapter automatically detects the views specified in this file.

You can also have multiple view definitions and control them using the [UserDefinedViews](#) connection property. When you use this property, only the specified views are seen by the adapter.

This User Defined View configuration file is formatted as follows:

- Each root element defines the name of a view.

- Each root element contains a child element, called **query**, which contains the custom SQL query for the view.

For example:

```
{
  "MyView": {
    "query": "SELECT * FROM Spreadsheet1_Sheet1 WHERE MyColumn = 'value'"
  },
  "MyView2": {
    "query": "SELECT * FROM MyTable WHERE Id IN (1,2,3)"
  }
}
```

Use the UserDefinedViews connectio property to specify the location of your JSON configuration file. For example:

```
"UserDefinedViews",
"C:\\Users\\yourusername\\Desktop\\tmp\\UserDefinedViews.json"
```

## Defining Views Using DDL Statements

The adapter is also capable of creating and altering the schema via DDL Statements such as CREATE LOCAL VIEW, ALTER LOCAL VIEW, and DROP LOCAL VIEW.

### Create a View

To create a new view using DDL statements, provide the view name and query as follows:

```
CREATE LOCAL VIEW [MyViewName] AS SELECT * FROM Customers LIMIT 20;
```

If no JSON file exists, the above code creates one. The view is then created in the JSON configuration file and is now discoverable. The JSON file location is specified by the UserDefinedViews connection property.

## Alter a View

To alter an existing view, provide the name of an existing view alongside the new query you would like to use instead:

```
ALTER LOCAL VIEW [MyViewName] AS SELECT * FROM Customers WHERE  
TimeModified > '3/1/2020';
```

The view is then updated in the JSON configuration file.

## Drop a View

To drop an existing view, provide the name of an existing schema alongside the new query you would like to use instead.

```
DROP LOCAL VIEW [MyViewName]
```

This removes the view from the JSON configuration file. It can no longer be queried.

## Schema for User Defined Views

User Defined Views are exposed in the **UserViews** schema by default. This is done to avoid the view's name clashing with an actual entity in the data model. You can change the name of the schema used for UserViews by setting the UserViewsSchemaName property.

## Working with User Defined Views

For example, a SQL statement with a User Defined View called *UserViews.RCustomers* only lists customers in Raleigh:

```
SELECT * FROM Customers WHERE City = 'Raleigh';
```

An example of a query to the driver:

```
SELECT * FROM UserViews.RCustomers WHERE Status = 'Active';
```

Resulting in the effective query to the source:



```
SELECT * FROM Customers WHERE City = 'Raleigh' AND Status = 'Active';
```

That is a very simple example of a query to a User Defined View that is effectively a combination of the view query and the view definition. It is possible to compose these queries in much more complex patterns. All SQL operations are allowed in both queries and are combined when appropriate.

## SSL Configuration

### 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.

## Firewall and Proxy

### 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.

## Logging

Capturing adapter logging can be very helpful when diagnosing error messages or other unexpected behavior.

### Basic Logging

You will simply need to set two connection properties to begin capturing adapter logging.

- Logfile: A filepath which designates the name and location of the log file.
- Verbosity: This is a numerical value (1-5) that determines the amount of detail in the log. See the page in the Connection Properties section for a breakdown of the five levels.
- MaxLogFileSize: When the limit is hit, a new log is created in the same folder with the date and time appended to the end. The default limit is 100 MB. Values lower than 100 kB will use 100 kB as the value instead.
- MaxLogFileCount: A string specifying the maximum file count of log files. When the limit is hit, a new log is created in the same folder with the date and time appended to the end and the oldest log file will be deleted. Minimum supported value is 2. A value of 0 or a negative value indicates no limit on the count.

Once this property is set, the adapter will populate the log file as it carries out various tasks, such as when authentication is performed or queries are executed. If the specified file doesn't already exist, it will be created.

### Log Verbosity

The verbosity level determines the amount of detail that the adapter reports to the Logfile. Verbosity levels from 1 to 5 are supported. These are described in the following list:

- 
- |   |   |
|---|---|
| 1 | Setting <u>Verbosity</u> to 1 will log the query, the number of rows returned by it, the start of execution and the time taken, and any errors. |
|---|---|
- 
- |   |  |
|---|--|
| 2 | Setting <u>Verbosity</u> to 2 will log everything included in <u>Verbosity</u> 1 and additional information about the request. |
|---|--|
- 
- |   |  |
|---|--|
| 3 | Setting <u>Verbosity</u> to 3 will additionally log HTTP headers, as well as the body of the request and the response. |
|---|--|
- 
- |   |  |
|---|--|
| 4 | Setting <u>Verbosity</u> to 4 will additionally log transport-level communication with the data source. This includes SSL negotiation. |
|---|--|
- 
- |   |  |
|---|--|
| 5 | Setting <u>Verbosity</u> to 5 will additionally log communication with the data source and additional details that may be helpful in troubleshooting problems. This includes interface commands. |
|---|--|
- 

The Verbosity should not be set to greater than 1 for normal operation. Substantial amounts of data can be logged at higher verbositys, which can delay execution times.

To refine the logged content further by showing/hiding specific categories of information, see LogModules.

## Java Logging

When Java logging is enabled in Logfile, the Verbosity will instead map to the following logging levels.

- 0: Level.WARNING
- 1: Level.INFO
- 2: Level.CONFIG
- 3: Level.FINE
- 4: Level.FINER
- 5: Level.FINEST

## Advanced Logging

You may want to refine the exact information that is recorded to the log file. This can be accomplished using the [LogModules](#) property.

This property allows you to filter the logging using a semicolon-separated list of logging modules.

All modules are four characters long. **Please note that modules containing three letters have a required trailing blank space.** The available modules are:

- **EXEC:** Query Execution. Includes execution messages for original SQL queries, parsed SQL queries, and normalized SQL queries. Query and page success/failure messages appear here as well.
- **INFO:** General Information. Includes the connection string, driver version (build number), and initial connection messages.
- **HTTP:** HTTP Protocol messages. Includes HTTP requests/responses (including POST messages), as well as Kerberos related messages.
- **SSL :** SSL certificate messages.
- **OAUT:** OAuth related failure/success messages.
- **SQL :** Includes SQL transactions, SQL bulk transfer messages, and SQL result set messages.
- **META:** Metadata cache and schema messages.
- **TCP :** Incoming and Ongoing raw bytes on TCP transport layer messages.

An example value for this property would be.

```
LogModules=INFO;EXEC;SSL ;SQL ;META;
```

Note that these modules refine the information as it is pulled after taking the [Verbosity](#) into account.

## SQL Compliance

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

## SELECT Statements

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

See [Using Spreadsheets as Tables](#) for information on the capabilities of the Google Sheets API.

## INSERT Statements

See [INSERT Statements](#) for a syntax reference and examples, as well as retrieving the new records' Ids.

## 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. Or, use [Batch Processing](#).

## 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 Sheets 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> ]
} | 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> { = } [ <expression> ]
    } [ { AND | OR } ... ]

```

## Examples

1. Return all columns:

```
SELECT * FROM Spreadsheet1_Sheet1
```

2. Rename a column:

```
SELECT "Column1" AS MY_Column1 FROM Spreadsheet1_Sheet1
```

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

```
SELECT CAST(AnnualRevenue AS VARCHAR) AS Str_AnnualRevenue FROM
Spreadsheet1_Sheet1
```

4. Search data:

```
SELECT * FROM Spreadsheet1_Sheet1 WHERE Column2 = 'Bob'
```

5. The Google Sheets APIs support the following operators in the WHERE clause: =.

```
SELECT * FROM Spreadsheet1_Sheet1 WHERE Column2 = 'Bob';
```

## 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, Column1 INTO
'csv://c:/Spreadsheet1_Sheet1.txt' FROM 'Spreadsheet1_Sheet1' WHERE
Column2 = 'Bob'");
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 'Spreadsheet1_Sheet1' IN
'csv://filename=c:/Spreadsheet1_Sheet1.csv;delimiter=tab' FROM
'Spreadsheet1_Sheet1' WHERE Column2 = 'Bob'");
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. To retrieve the Id of the last inserted record use getGeneratedKeys. Additionally, set the **RETURN\_GENERATED\_KEYS** flag of the Statement class when you call prepareStatement.

```
String cmd = "INSERT INTO Spreadsheet1_Sheet1 (Column1) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement
(cmd,Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, "John");
int count = pstmt.executeUpdate();
System.out.println(count+" rows were affected");
ResultSet rs = pstmt.getGeneratedKeys();
while(rs.next()){
    System.out.println(rs.getString("Id"));
}
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 Spreadsheet1_Sheet1 SET Column1='John' WHERE Id =
?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "6");
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:googlesheets:InitiateOAuth=GETANDREFRESH;Spreadsheet=NorthwindOrders", );
String cmd = "DELETE FROM Spreadsheet1_Sheet1 WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "6");
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;
```

# Using Spreadsheets as Tables

## Spreadsheets as Tables

The adapter models spreadsheets and ranges as relational tables.

## Tables

[Tables](#) shows various configuration options to reflect your spreadsheets' organization in the tables; for example, you will find guides for working with headers and querying ranges as tables.

[Columns](#) provides more information on column discovery.

## Stored Procedures

In addition to working with the data in the spreadsheet, you can use the available stored procedures to access functionality in the Google Sheets API that is not modeled as SELECT, INSERT, UPDATE, or DELETE statements.

## Tables

The adapter enables you to represent a top-left-oriented spreadsheet or a user-specified range as a database table. You can control how tables are listed by setting the Header property.

## Top-Left Oriented Tables

You can use the adapter to start working right away with top-left-oriented tables. The default configuration settings are explained below:

- Top-left-oriented tables are represented with the name of the worksheet.
- The default format requires that the table is top-left-oriented and that the first row of data in the worksheet contains the column names. This means that the default value of true for the Header connection string property is required.

- Headers should not contain special characters.
- By default the adapter will return all rows until the first empty row. *Note:* an empty row between data will prevent further data from being returned.

Due to a limitation of Google's Spreadsheet API, all column headers must be non empty.

## User-Specified Range

You can execute SQL commands against a given range as a table by using this format in your query: *WORKSHEET#RANGE*

*Note:* Range notation is only available in a SELECT or UPDATE statement. Ranges are not supported for DELETE and INSERT commands.

## Columns

You can specify column names or generate column names automatically by setting the Header property. This property affects how you use columns in commands.

### Header=True (Default)

- Columns are determined by the first row of the Google spreadsheet. If no values are provided for the first row of the spreadsheet, the adapter will create unique, alphabetized column names that are available only within the scope of that request.
- The adapter also adds an Id column for each row that corresponds to the unique URI of the row on the Google servers. This is used during update and delete operations.

### Header=False

- Columns will be dynamically assigned based on either the specified range or the size of the worksheet. The autogenerated column names are alphabetical.
- The Id column for each row will represent the row number from the top of the sheet. For example, if you specify a range A3:E6, rows 3, 4, 5, and 6 will be 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 Sheets Adapter Views

| Name                         | Description   |
|------------------------------|---|
| <a href="#">Folders</a>      | Query the folders contained in a user's Google Drive.                   |
| <a href="#">Sheets</a>       | Returns a list of a user's sheets and their relevant information.       |
| <a href="#">Spreadsheets</a> | Returns a list of a user's spreadsheets and their relevant information. |

## Folders

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

## Select

The adapter will use the Google Sheets API to process WHERE clause conditions built with the server side supported columns and operators. The rest of the filter is executed client side within the adapter.

The columns and operators that support server side filtering are:

- **Name** supports the 'CONTAINS,=,!=' operators.
- **Description** supports the 'CONTAINS' operator.
- **ModifiedTime** supports the '<=,<,>,>=' operators.
- **OwnerEmail** supports the 'IN' operator.

- **Starred** supports the '=',!= ' operators.
- **Trashed** supports the '=',!= ' operators.
- **ParentIds** supports the 'IN' operator.
- **TeamDriveld** supports the '=' operator. It is used to get all the folders from the specified Team Drive.

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

All the columns that support server side filtering can be paired with the AND and OR logical operators. For example, the following queries are processed server side:

```
SELECT * FROM Folders WHERE Name='example folder'
```

```
SELECT * FROM Folders WHERE OwnerEmail IN
('owner1@email.com','owner2@email.com') AND ModifiedTime >= '2020-04-
01T05:30:00'
```

## Columns

| Name        | Type          | Description   |
|-------------|---------------|---|
| Id<br>[KEY] | <i>String</i> | The ID of the folder.   |
| Name        | <i>String</i> | The name of the folder. 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. |
| TeamDriveld | <i>String</i> | The Id of the teamDrive.  |
| Description | <i>String</i> | A short description of the folder or folder.  |

|              |                 |  |
|--------------|-----------------|--|
| CreatedTime  | <i>Datetime</i> | The creation date of the folder or folder.                                       |
| ModifiedTime | <i>Datetime</i> | The last modified date of the folder or folder.                                  |
| Size         | <i>Long</i>     | The size of the folder 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

Returns a list of a user's sheets and their relevant information.

## Columns

| Name            | Type           | Description                                    |
|-----------------|----------------|--|
| SpreadsheetId   | <i>String</i>  | The assigned Id of the spreadsheet.            |
| SpreadsheetName | <i>String</i>  | The name of the spreadsheet.                   |
| SheetId         | <i>String</i>  | A short description of the spreadsheet.        |
| SheetName       | <i>String</i>  | The name of the sheet.                         |
| SheetIndex      | <i>Integer</i> | The index of the sheet within the spreadsheet. |
| SheetType       | <i>String</i>  | The type of sheet. Defaults to GRID.           |
| RowCount        | <i>Integer</i> | The number of rows in the grid.                |

---

|                   |                |  |
|-------------------|----------------|--|
| ColumnCount       | <i>Integer</i> | The number of columns in the grid.                 |
| FrozenRowCount    | <i>Integer</i> | The number of rows that are frozen in the grid.    |
| FrozenColumnCount | <i>Integer</i> | The number of columns that are frozen in the grid. |

---

## Spreadsheets

Returns a list of a user's spreadsheets and their relevant information.

## Columns

| Name         | Type            | Description   |
|--------------|-----------------|---|
| Id<br>[KEY]  | <i>String</i>   | The assigned Id of the spreadsheet.                 |
| Name         | <i>String</i>   | The name of the spreadsheet.                        |
| Description  | <i>String</i>   | A short description of the spreadsheet.             |
| OwnerName    | <i>String</i>   | The name of the resource's owner.                   |
| OwnerEmail   | <i>String</i>   | The email of the resource's owner.                  |
| ModifiedTime | <i>Datetime</i> | The last updated date and time of this spreadsheet. |

---

|             |                 |  |
|-------------|-----------------|--|
| CreatedTime | <i>Datetime</i> | The created date and time of this spreadsheet. |
| Trashed     | <i>Boolean</i>  | Whether the spreadsheet has been trashed.      |
| Starred     | <i>Boolean</i>  | Whether the user has starred the spreadsheet.  |
| Hidden      | <i>Boolean</i>  | Whether the spreadsheet is hidden.             |
| Viewed      | <i>Boolean</i>  | Whether the user has viewed the spreadsheet.   |
| ParentIds   | <i>String</i>   | A comma-separated list of parent folder Ids.   |

## Stored Procedures

Stored procedures are available to complement the data available from the [Using Spreadsheets as Tables](#). 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 Sheets Adapter Stored Procedures

| Name                     | Description  |
|--------------------------|--|
| <a href="#">AddSheet</a> | Add a worksheet to an existing Google spreadsheet. |

|  |  |
|--|--|
| <a href="#">CopySheet</a>                | Copies a sheet from a spreadsheet to another spreadsheet.                                  |
| <a href="#">CreateSchema</a>             | Creates a schema file for the specified table or view.                                     |
| <a href="#">CreateSpreadsheet</a>        | Creates a spreadsheet in the user's Google Drive.  |
| <a href="#">DeleteSheet</a>              | Deletes a worksheet in an existing Google spreadsheet.                                     |
| <a href="#">DeleteSpreadsheet</a>        | Deletes a spreadsheet.   |
| <a href="#">DownloadDocument</a>         | Downloads a file from the user's Google Drive.   |
| <a href="#">FormatRange</a>              | Format cells in a specific range   |
| <a href="#">GetOAuthAccessToken</a>      | Obtains the OAuth access token to be used for authentication with various Google services. |
| <a href="#">GetOAuthAuthorizationURL</a> | Obtains the OAuth authorization URL for authentication with various Google services.       |
| <a href="#">RefreshOAuthAccessToken</a>  | Obtains the OAuth access token to be used for authentication with various Google services. |
| <a href="#">UpdateSheet</a>              | Updates properties of the sheet for the specified SpreadsheetId and SheetId.               |
| <a href="#">UploadDocument</a>           | Uploads a file to the user's Google Drive.   |

## AddSheet

Add a worksheet to an existing Google spreadsheet.

### Input

| Name          | Type   | Description                |
|---------------|--------|----------------------------|
| SpreadsheetId | String | The ID of the spreadsheet. |

|                   |               |   |
|-------------------|---------------|---|
| SheetId           | <i>String</i> | The ID of the sheet. Must be non-negative. This field cannot be changed once set.   |
| Title             | <i>String</i> | The name of the sheet.  |
| Index             | <i>String</i> | The index of the sheet within the spreadsheet.  |
| SheetType         | <i>String</i> | <p>The type of sheet. Defaults to GRID. This field cannot be changed once set.</p> <p>The allowed values are <i>GRID</i>, <i>OBJECT</i>.</p> <p>The default value is <i>GRID</i>.</p> |
| RowCount          | <i>String</i> | The number of rows in the grid.   |
| ColumnCount       | <i>String</i> | The number of columns in the grid.  |
| FrozenRowCount    | <i>String</i> | The number of rows that are frozen in the grid.   |
| FrozenColumnCount | <i>String</i> | The number of columns that are frozen in the grid.  |
| HideGridlines     | <i>String</i> | <p>True if the grid is not showing gridlines in the UI.</p> <p>The allowed values are <i>true</i>, <i>false</i>.</p>  |
| Hidden            | <i>String</i> | <p>True if the sheet is hidden in the UI, false if it is visible.</p> <p>The allowed values are <i>true</i>, <i>false</i>.</p>  |
| RightToLeft       | <i>String</i> | <p>True if the sheet is an RTL sheet instead of an LTR sheet.</p> <p>The allowed values are <i>true</i>, <i>false</i>.</p>  |

## Result Set Columns

| Name    | Type          | Description   |
|---------|---------------|---|
| Success | <i>String</i> | This value shows whether the operation was successful or not. |

## CopySheet

Copies a sheet from a spreadsheet to another spreadsheet.

### Input

| Name                     | Type          | Description  |
|--------------------------|---------------|--|
| SpreadsheetId            | <i>String</i> | The ID of the spreadsheet containing the sheet to copy.                      |
| SheetId                  | <i>String</i> | The ID of the sheet to copy  |
| DestinationSpreadsheetId | <i>String</i> | The ID of the spreadsheet where it will be copied to.                        |
| SheetName                | <i>String</i> | Optional parameter. The new name you want to set to the newly created sheet. |

## Result Set Columns

| Name    | Type          | Description   |
|---------|---------------|---|
| Success | <i>String</i> | This value shows whether the operation was successful or not. |
| SheetId | <i>String</i> | The Id of the newly created sheet                             |

## CreateSchema

Creates a schema file for the specified table or view.

### Input

| Name      | Type          | Description  |
|-----------|---------------|--|
| TableName | <i>String</i> | The name of the table or view.   |
| FileName  | <i>String</i> | The full file path and name of the schema to generate. Ex :<br>'C:\\Users\\User\\Desktop\\SmartSheet\\sheet.rsd' |

### Result Set Columns

| Name   | Type          | Description                 |
|--------|---------------|-----------------------------|
| Result | <i>String</i> | Returns Success or Failure. |

## CreateSpreadsheet

Creates a spreadsheet in the user's Google Drive.

## Input

| Name        | Type          | Description  |
|-------------|---------------|--|
| Title       | <i>String</i> | The title for the spreadsheet.   |
| Description | <i>String</i> | The description for the spreadsheet.   |
| Hidden      | <i>String</i> | This parameter sets whether or not the resource is hidden.<br>The allowed values are <i>TRUE</i> , <i>FALSE</i> .<br>The default value is <i>FALSE</i> .     |
| Restricted  | <i>String</i> | This parameter sets whether or not the resource is restricted.<br>The allowed values are <i>TRUE</i> , <i>FALSE</i> .<br>The default value is <i>FALSE</i> . |
| Starred     | <i>String</i> | This parameter sets whether or not the resource is starred.<br>The allowed values are <i>TRUE</i> , <i>FALSE</i> .<br>The default value is <i>FALSE</i> .    |
| Parents     | <i>String</i> | The Ids of the parent folders for the created spreadsheet.   |

## Result Set Columns

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



|         |               |   |
|---------|---------------|---|
| Success | <i>String</i> | This parameter indicates whether the operation was successful or not. |
| Id      | <i>String</i> | The Id of the new spreadsheet.  |

## DeleteSheet

Deletes a worksheet in an existing Google spreadsheet.

### Input

| Name          | Type          | Description                |
|---------------|---------------|----------------------------|
| SpreadsheetId | <i>String</i> | The ID of the spreadsheet. |
| SheetId       | <i>String</i> | The ID of the sheet.       |

### Result Set Columns

| Name    | Type          | Description   |
|---------|---------------|---|
| Success | <i>String</i> | This value shows whether the operation was successful or not. |

## DeleteSpreadsheet

Deletes a spreadsheet.

## Input

| Name          | Type          | Description                |
|---------------|---------------|----------------------------|
| SpreadsheetId | <i>String</i> | The ID of the spreadsheet. |

## Result Set Columns

| Name    | Type          | Description   |
|---------|---------------|---|
| Success | <i>String</i> | This value shows whether the operation was successful or not. |

## DownloadDocument

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.   |
| FileFormat | <i>String</i> | <p>The file format to be applied when saving the file, such as text/plain.</p> <p>The default value is <i>application/vnd.openxmlformats-officedocument.spreadsheetml.sheet</i>.</p> |
| 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.                    |

## FormatRange

Format cells in a specific range

### Input

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

|                 |               |   |
|-----------------|---------------|---|
| SpreadsheetId   | <i>String</i> | The ID of the spreadsheet.  |
| SheetId         | <i>String</i> | The ID of the sheet. Must be non-negative.  |
| Range           | <i>String</i> | The range of the cells to format. The format should be, for example A1:C3.                  |
| BackgroundColor | <i>String</i> | Specify background color by providing RGB values for Red, Green, Blue. For example 255,0,0. |
| Bold            | <i>String</i> | Bold the text of the cells. Accepts true or false.  |
| FontSize        | <i>String</i> | Set font size by providing an int value.  |
| FontFamily      | <i>String</i> | Set the font used by providing its name.  |

## Result Set Columns

| <b>Name</b> | <b>Type</b>   | <b>Description</b>  |
|-------------|---------------|---|
| Success     | <i>String</i> | This value shows 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 | <p>The type of authentication mode to use.</p> <p>The allowed values are <i>APP</i>, <i>WEB</i>.</p> <p>The default value is <i>WEB</i>.</p>  |
| Verifier    | String | The verifier code returned by Google after permissions have been granted for the app to connect. <i>WEB</i> AuthMode only.  |
| 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 <i>https://www.googleapis.com/auth/drive</i><br/><i>https://www.googleapis.com/auth/spreadsheets</i>.</p>   |
| CallbackURL | String | 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 '/').   |
| Prompt      | String | <p>This field indicates the prompt to present the user. It accepts one of the following values: <i>NONE</i>, <i>CONSENT</i>, <i>SELECT ACCOUNT</i>. The default is <i>SELECT_ACCOUNT</i>, so a given user will be prompted to select the account to connect to. If it is set to <i>CONSENT</i>, the user will see a consent page every time, even if they have previously given consent to the application for a given set of scopes. Lastly, if it is set to <i>NONE</i>, no authentication or consent screens will be displayed to the user.</p> <p>The default value is <i>SELECT_ACCOUNT</i>.</p> |
| AccessType  | String | Indicates if your application needs to access a Google API when the user is not present at the browser. This parameter defaults to <i>OFFLINE</i> . If your application needs to refresh access tokens when the user is not present at the browser, then use <i>OFFLINE</i> . This will result in your application obtaining a refresh token the first time your application exchanges an authorization code for a user.  |

The allowed values are *ONLINE*, *OFFLINE*.

The default value is *OFFLINE*.

|       |               |  |
|-------|---------------|--|
| State | <i>String</i> | Indicates any state which 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 the 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. |
|-------|---------------|--|

## 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> | A token that may be used to obtain a new access token.   |
| ExpiresIn         | <i>String</i> | The remaining lifetime on the access token.  |

## GetOAuthAuthorizationURL

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

## Input

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

|             |               |   |
|-------------|---------------|---|
| Scope       | <i>String</i> | <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 <i>https://www.googleapis.com/auth/drive</i><br/><i>https://www.googleapis.com/auth/spreadsheets</i>.</p>   |
| CallbackURL | <i>String</i> | <p>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 '/').</p>  |
| Prompt      | <i>String</i> | <p>This field indicates the prompt to present the user. It accepts one of the following values: NONE, CONSENT, SELECT ACCOUNT. The default is SELECT_ACCOUNT, so a given user will be prompted to select the account to connect to. If it is set to CONSENT, the user will see a consent page every time, even if they have previously given consent to the application for a given set of scopes. Lastly, if it is set to NONE, no authentication or consent screens will be displayed to the user.</p> <p>The default value is <i>SELECT_ACCOUNT</i>.</p> |
| AccessType  | <i>String</i> | <p>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       | <i>String</i> | <p>Indicates any state which 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 the Google authorization server and back. Uses include redirecting the user to the correct resource in your site, nonces, and cross-site-request-forgery mitigations.</p>  |

## 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. |
| ExpiresIn        | <i>String</i> | The remaining lifetime on the access token.  |



## UpdateSheet

Updates properties of the sheet for the specified SpreadsheetId and SheetId.

### Input

| Name              | Type   | Description   |
|-------------------|--------|---|
| SpreadsheetId     | String | The ID of the spreadsheet.  |
| SheetId           | String | The ID of the sheet. Must be non-negative.  |
| Title             | String | The name of the sheet.  |
| Index             | String | The index of the sheet within the spreadsheet.  |
| RowCount          | String | The number of rows in the grid.   |
| ColumnCount       | String | The number of columns in the grid.  |
| FrozenRowCount    | String | The number of rows that are frozen in the grid.   |
| FrozenColumnCount | String | The number of columns that are frozen in the grid.  |
| HideGridlines     | String | True if the grid is not showing gridlines in the UI.<br>The allowed values are <i>true</i> , <i>false</i> . |
| Hidden            | String | True if the sheet is hidden in the UI, false if it is visible.  |

|   |               |  |
|---|---------------|--|
| The allowed values are <i>true</i> , <i>false</i> . |               |  |
| RightToLeft   | <i>String</i> | True if the sheet is an RTL sheet instead of an LTR sheet. |
| The allowed values are <i>true</i> , <i>false</i> . |               |  |

## Result Set Columns

| Name    | Type          | Description   |
|---------|---------------|---|
| Success | <i>String</i> | This value shows whether the operation was successful or not. |

## UploadDocument

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 name 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.<br>The allowed values are <i>TRUE</i> , <i>FALSE</i> . |

|                                     |               |  |
|-------------------------------------|---------------|--|
| The default value is <i>FALSE</i> . |               |  |
| ParentIds                           | <i>String</i> | The Ids of the parent folders for the uploaded document.   |
| MIMETYPE                            | <i>String</i> | The MIME type of the file.<br><br>The default value is <i>application/vnd.google-apps.spreadsheet</i> .                                    |
| 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.<br><br>The allowed values are <i>NONE</i> , <i>BASE64</i> .<br><br>The default value is <i>BASE64</i> .  |

## Result Set Columns

| Name    | Type          | Description  |
|---------|---------------|--|
| Id      | <i>String</i> | The id of the file.  |
| Success | <i>String</i> | 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.

For more information on establishing a connection, see [Basic Tab](#) in the Google Sheets Adapter Guide.

## Authentication

| Property                   | Description  |
|----------------------------|--|
| <a href="#">AuthScheme</a> | The type of authentication to use when connecting to Google Sheets.  |
| <a href="#">APIKey</a>     | If your client application does not use OAuth 2.0, then it must include an API key when it calls an API that's enabled within a Google Cloud Platform project. |

## Connection

| Property                              | Description   |
|---------------------------------------|---|
| <a href="#">Spreadsheet</a>           | A comma-separated list of the names or Ids of the spreadsheets to be viewed.  |
| <a href="#">FolderName</a>            | A comma separated list of the folders' names from which to retrieve spreadsheets in the format FolderName='name1,name2'.  |
| <a href="#">FolderId</a>              | A comma separated list of the folders' ids from which to retrieve spreadsheets in the format FolderId='id1,id2,id3'.  |
| <a href="#">ShowTrashedFiles</a>      | Indicates whether or not the trashed files will be listed.  |
| <a href="#">TeamDriveSupport</a>      | Determines whether or not to retrieve Team Drive items.   |
| <a href="#">UseIdAsTableName</a>      | Indicates whether or not to use Ids as Spreadsheet and Sheet name.  |
| <a href="#">TeamDrive</a>             | A drive's names or ids from which to retrieve spreadsheets in the format TeamDrive='Shared drive 2, Shared drive 3', or TeamDrive='0BKwyFj1j9FOsUk9EVO, 0ANMIP9RIe1LQUk9PVA'. |
| <a href="#">DomainSharedFilesOnly</a> | Boolean determining if the exposed sheets are limited to only   |

|                                |   |
|--------------------------------|---|
|                                | the sheets shared to the user's domain or not.  |
| <a href="#">RecurseFolders</a> | Used in case FolderId/FolderName properties are defined. If set to True this makes the driver return all the Spreadsheets inside nested folders, else the driver will return only the files directly to that folder. By default this is set to false. |

## OAuth

| Property                              | Description  |
|---------------------------------------|--|
| <a href="#">InitiateOAuth</a>         | Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.  |
| <a href="#">OAuthClientId</a>         | The client Id assigned when you register your application with an OAuth authorization server.  |
| <a href="#">OAuthClientSecret</a>     | The client secret assigned when you register your application with an OAuth authorization server.  |
| <a href="#">OAuthAccessToken</a>      | The access token for connecting using OAuth.   |
| <a href="#">OAuthSettingsLocation</a> | 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://. |
| <a href="#">OAuthVerifier</a>         | The verifier code returned from the OAuth authorization URL.   |
| <a href="#">OAuthRefreshToken</a>     | The OAuth refresh token for the corresponding OAuth access token.  |
| <a href="#">OAuthExpiresIn</a>        | The lifetime in seconds of the OAuth AccessToken.  |
| <a href="#">OAuthTokenTimestamp</a>   | The Unix epoch timestamp in milliseconds when the current Access Token was created.  |

## JWT OAuth

| Property                             | Description  |
|--------------------------------------|--|
| <a href="#">OAuthJWTCert</a>         | The JWT Certificate store.   |
| <a href="#">OAuthJWTCertType</a>     | The type of key store containing the JWT Certificate.                      |
| <a href="#">OAuthJWTCertPassword</a> | The password for the OAuth JWT certificate.                                |
| <a href="#">OAuthJWTCertSubject</a>  | The subject of the OAuth JWT certificate.                                  |
| <a href="#">OAuthJWTIssuer</a>       | The issuer of the Java Web Token.  |
| <a href="#">OAuthJWTSubject</a>      | The user subject for which the application is requesting delegated access. |

## SSL

| Property                      | Description   |
|-------------------------------|---|
| <a href="#">SSLServerCert</a> | The certificate to be accepted from the server when connecting using TLS/SSL. |

## Firewall

| Property                       | Description                                       |
|--------------------------------|---|
| <a href="#">FirewallType</a>   | The protocol used by a proxy-based firewall.      |
| <a href="#">FirewallServer</a> | The name or IP address of a proxy-based firewall. |

|                                  |   |
|----------------------------------|---|
| <a href="#">FirewallPort</a>     | The TCP port for a proxy-based firewall.                          |
| <a href="#">FirewallUser</a>     | The user name to use to authenticate with a proxy-based firewall. |
| <a href="#">FirewallPassword</a> | A password used to authenticate to a proxy-based firewall.        |

## Proxy

| Property                        | Description   |
|---------------------------------|---|
| <a href="#">ProxyAutoDetect</a> | 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. |
| <a href="#">ProxyServer</a>     | The hostname or IP address of a proxy to route HTTP traffic through.  |
| <a href="#">ProxyPort</a>       | The TCP port the ProxyServer proxy is running on.   |
| <a href="#">ProxyAuthScheme</a> | The authentication type to use to authenticate to the ProxyServer proxy.  |
| <a href="#">ProxyUser</a>       | A user name to be used to authenticate to the ProxyServer proxy.  |
| <a href="#">ProxyPassword</a>   | A password to be used to authenticate to the ProxyServer proxy.   |
| <a href="#">ProxySSLType</a>    | The SSL type to use when connecting to the ProxyServer proxy.   |
| <a href="#">ProxyExceptions</a> | A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .  |

## Logging

| Property                   | Description                                  |
|----------------------------|--|
| <a href="#">LogModules</a> | Core modules to be included in the log file. |

## Schema

| Property                             | Description   |
|--------------------------------------|---|
| <a href="#">Location</a>             | A path to the directory that contains the schema files defining tables, views, and stored procedures. |
| <a href="#">TypeDetectionScheme</a>  | Determines how to determine the data types of columns.  |
| <a href="#">Header</a>               | Indicates whether or not the first row should be used as a column header.                             |
| <a href="#">PrimaryKeyIdentifier</a> | Set this property to control the name of the primary key.   |
| <a href="#">DefineTables</a>         | Define the tables within the Google Spreadsheet.  |
| <a href="#">Orientation</a>          | Indicates whether the data in the sheet is laid out horizontally or vertically.                       |

## Miscellaneous

| Property                             | Description   |
|--------------------------------------|---|
| <a href="#">AutoAdjustRange</a>      | If set to true, the driver will automatically expand the dimensions in case the updated/insert/deleted value is outside the range of the sheet.   |
| <a href="#">DateTimeRenderOption</a> | Determines how dates, times, and durations should be represented in the output. This is ignored if the ValueRenderOption is FormattedValue. The default datetime render option is SerialNumber. |
| <a href="#">InsertDataOption</a>     | Determines how existing data is changed when new data is input.   |
| <a href="#">MaxRows</a>              | 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.        |
| <a href="#">NullValueMode</a>       | Indicates whether to read empty cells as null or as empty.                                |
| <a href="#">Other</a>               | These hidden properties are used only in specific use cases.                              |
| <a href="#">Pagesize</a>            | The maximum number of results to return per page from Google Sheets.                      |
| <a href="#">PercentageToDecimal</a> | Boolean determining if percentage columns should be considered as decimal.                |
| <a href="#">Readonly</a>            | You can use this property to enforce read-only access to Google Sheets from the provider. |
| <a href="#">RowScanDepth</a>        | The maximum number of rows to scan to look for the columns available in a table.          |
| <a href="#">ShowEmptyRows</a>       | Indicates whether or not the empty rows should be pushed.                                 |
| <a href="#">Timeout</a>             | The value in seconds until the timeout error is thrown, canceling the operation.          |
| <a href="#">UseSimpleNames</a>      | Boolean determining if simple names should be used for tables and columns.                |
| <a href="#">ValueInputOption</a>    | Determines how inserted values should be treated.   |
| <a href="#">ValueRenderOption</a>   | Determines how values should be rendered in the output.                                   |

## Authentication

This section provides a complete list of the Authentication properties you can configure in the connection string for this provider.

| Property | Description |
|----------|-------------|
|          |             |

---

|                            |  |
|----------------------------|--|
| <a href="#">AuthScheme</a> | The type of authentication to use when connecting to Google Sheets.  |
| <a href="#">APIKey</a>     | If your client application does not use OAuth 2.0, then it must include an API key when it calls an API that's enabled within a Google Cloud Platform project. |

---

## AuthScheme

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

### Possible Values

Auto, Token, OAuth, OAuthJWT, GCPIInstanceAccount

### Data Type

string

### Default Value

"Auto"

### Remarks

- Auto: Lets the driver decide automatically based on the other connection properties you have set.
- Token: Set this to perform Token Based Authentication via the [APIKey](#) property.
- OAuth: Set this to perform OAuth authentication using a standard user account.
- OAuthJWT: Set this to perform OAuth authentication using an OAuth service account.
- GCPIInstanceAccount: Set this to get Access Token from Google Cloud Platform instance.

## APIKey

If your client application does not use OAuth 2.0, then it must include an API key when it calls an API that's enabled within a Google Cloud Platform project.

## Data Type

string

## Default Value

""

## Remarks

If your client application does not use OAuth 2.0, then it must include an API key when it calls an API that's enabled within a Google Cloud Platform project.

## Connection

This section provides a complete list of the Connection properties you can configure in the connection string for this provider.

| Property                         | Description  |
|----------------------------------|--|
| <a href="#">Spreadsheet</a>      | A comma-separated list of the names or Ids of the spreadsheets to be viewed.   |
| <a href="#">FolderName</a>       | A comma separated list of the folders' names from which to retrieve spreadsheets in the format FolderName='name1,name2'. |
| <a href="#">FolderId</a>         | A comma separated list of the folders' ids from which to retrieve spreadsheets in the format FolderId='id1,id2,id3'.     |
| <a href="#">ShowTrashedFiles</a> | Indicates whether or not the trashed files will be listed.   |
| <a href="#">TeamDriveSupport</a> | Determines whether or not to retrieve Team Drive items.  |
| <a href="#">UseIdAsTableName</a> | Indicates whether or not to use Ids as Spreadsheet and Sheet   |

|                              |   |
|------------------------------|---|
|                              | name.   |
| <b>TeamDrive</b>             | A drive's names or ids from which to retrieve spreadsheets in the format TeamDrive='Shared drive 2, Shared drive 3', or TeamDrive='0BKwyFj1j9FOsUk9EVO, 0ANMIP9RIe1LQUk9PVA'.   |
| <b>DomainSharedFilesOnly</b> | Boolean determining if the exposed sheets are limited to only the sheets shared to the user's domain or not.  |
| <b>RecurseFolders</b>        | Used in case FolderId/FolderName properties are defined. If set to True this makes the driver return all the Spreadsheets inside nested folders, else the driver will return only the files directly to that folder. By default this is set to false. |

## Spreadsheet

A comma-separated list of the names or Ids of the spreadsheets to be viewed.

### Data Type

string

### Default Value

""

### Remarks

A comma-separated list of the names or Ids of the spreadsheets to be viewed. Query the Spreadsheets view to retrieve this data.

**Note:** In case you are providing the names of the spreadsheets, make sure to provide the exact spreadsheet name, including the leading and/or trailing spaces. Also, you should not add extra spaces before and after the comma separator.

## FolderName

A comma separated list of the folders' names from which to retrieve spreadsheets in the format FolderName='name1,name2'.

## Data Type

string

## Default Value

""

## Remarks

A comma separated list of the folders' names from which to retrieve spreadsheets in the format FolderName='name1,name2'.

## FolderId

A comma separated list of the folders' ids from which to retrieve spreadsheets in the format FolderId='id1,id2,id3'.

## Data Type

string

## Default Value

""

## Remarks

A comma separated list of the folders' ids from which to retrieve spreadsheets in the format FolderId='id1,id2,id3'.

## ShowTrashedFiles

Indicates whether or not the trashed files will be listed.

## Data Type

bool

## Default Value

false

## Remarks

If true, the driver will list the files/spreadsheets that have been trashed.

## TeamDriveSupport

Determines whether or not to retrieve Team Drive items.

## Data Type

bool

## Default Value

false

## Remarks

If you set this property to 'true', you can query from any Team Drive spreadsheets.

## UseIdAsTableName

Indicates whether or not to use Ids as Spreadsheet and Sheet name.

## Data Type

bool

## Default Value

false

## Remarks

Indicates whether or not to use Ids as Spreadsheet and Sheet name. To select in a sheet instead of SpreadsheetName\_SheetName use: SpreadsheetId\_SheetId. Ex: SELECT \* FROM 11696gdF5QUL1EnYikYiUeMTHRqA1111KbdYDoINql\_1151117664

## TeamDrive

A drive's names or ids from which to retrieve spreadsheets in the format TeamDrive='Shared drive 2, Shared drive 3', or TeamDrive='0BKwyFj1j9FOsUk9EVO, 0ANMIP9Rle1LQUk9PVA'.

## Data Type

string

## Default Value

""

## Remarks

A drive's names or ids from which to retrieve spreadsheets in the format TeamDrive='Shared drive 2, Shared drive 3', or TeamDrive='0BKwyFj1j9FOsUk9EVO, 0ANMIP9Rle1LQUk9PVA'.

## DomainSharedFilesOnly

Boolean determining if the exposed sheets are limited to only the sheets shared to the user's domain or not.

## Data Type

bool

## Default Value

false

## Remarks

If true, the driver will retrieve and expose only the sheets shared to the user's domain, excluding the sheets owned by the user. If false, the driver will retrieve both, the files owned by and shared to the user.

## RecurseFolders

Used in case FolderId/FolderName properties are defined. If set to True this makes the driver return all the Spreadsheets inside nested folders, else the driver will return only the files directly to that folder. By default this is set to false.

## Data Type

string

## Default Value

"false"

## Remarks

Used in case FolderId/FolderName properties are defined. If set to True this makes the driver return all the Spreadsheets inside nested folders, else the driver will return only the files directly to that folder. By default this is set to false.

## OAuth

This section provides a complete list of the OAuth properties you can configure in the connection string for this provider.

---



| Property                              | Description  |
|---------------------------------------|--|
| <a href="#">InitiateOAuth</a>         | Set this property to initiate the process to obtain or refresh the OAuth access token when you connect.  |
| <a href="#">OAuthClientId</a>         | The client Id assigned when you register your application with an OAuth authorization server.  |
| <a href="#">OAuthClientSecret</a>     | The client secret assigned when you register your application with an OAuth authorization server.  |
| <a href="#">OAuthAccessToken</a>      | The access token for connecting using OAuth.   |
| <a href="#">OAuthSettingsLocation</a> | 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://. |
| <a href="#">OAuthVerifier</a>         | The verifier code returned from the OAuth authorization URL.   |
| <a href="#">OAuthRefreshToken</a>     | The OAuth refresh token for the corresponding OAuth access token.  |
| <a href="#">OAuthExpiresIn</a>        | The lifetime in seconds of the OAuth AccessToken.  |
| <a href="#">OAuthTokenTimestamp</a>   | The Unix epoch timestamp in milliseconds when the current Access Token was created.  |

## InitiateOAuth

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

### Possible Values

OFF, GETANDREFRESH, REFRESH

### 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.

## OAuthClientId

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](#).

## OAuthClientSecret

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.

## OAuthAccessToken

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.

## OAuthSettingsLocation

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%\CDData\GoogleSheets 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%\CDData\GoogleSheets 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                     |

## OAuthVerifier

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 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.

## OAuthRefreshToken

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.

## OAuthExpiresIn

The lifetime in seconds of the OAuth AccessToken.

### Data Type

string

### Default Value

""

### Remarks

Pair with OAuthTokenTimestamp to determine when the AccessToken will expire.

## OAuthTokenTimestamp

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.

## JWT OAuth

This section provides a complete list of the JWT OAuth properties you can configure in the connection string for this provider.

| Property                             | Description  |
|--------------------------------------|--|
| <a href="#">OAuthJWTCert</a>         | The JWT Certificate store.   |
| <a href="#">OAuthJWTCertType</a>     | The type of key store containing the JWT Certificate.                      |
| <a href="#">OAuthJWTCertPassword</a> | The password for the OAuth JWT certificate.                                |
| <a href="#">OAuthJWTCertSubject</a>  | The subject of the OAuth JWT certificate.                                  |
| <a href="#">OAuthJWTIssuer</a>       | The issuer of the Java Web Token.  |
| <a href="#">OAuthJWTSubject</a>      | The user subject for which the application is requesting delegated access. |

## OAuthJWTCert

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).



## OAuthJWTCertType

The type of key store containing the JWT Certificate.

### Possible Values

USER, MACHINE, PFXFILE, PFXBLOB, JKSFILe, JKSBLOB, PEMKEY\_FILE, PEMKEY\_BLOB, PUBLIC\_KEY\_FILE, PUBLIC\_KEY\_BLOB, SSHPUBLIC\_KEY\_FILE, SSHPUBLIC\_KEY\_BLOB, P7BFILE, PPKFILE, XMLFILE, XMLBLOB, GOOGLEJSON, GOOGLEJSONBLOB

### Data Type

string

### Default Value

"USER"

### Remarks

This property can take one of the following values:

|         |   |
|---------|---|
| USER    | 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.                                 |
| GOOGLEJSONBLOB     | The certificate store is a string that contains the service account JSON. Only valid when connecting to a Google service.  |

## OAuthJWTCertPassword

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.

## OAuthJWTCertSubject

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.

## OAuthJWTIssuer

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.

## OAuthJWTSubject

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.

## SSL

This section provides a complete list of the SSL properties you can configure in the connection string for this provider.

| Property                      | Description   |
|-------------------------------|---|
| <a href="#">SSLServerCert</a> | The certificate to be accepted from the server when connecting using TLS/SSL. |

## SSLServerCert

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-----<br>MIICHTCCAe4CAQAwDQYJKoZIhvc...<br>...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-----<br>MIGfMA0GCSq.....AQAB -----END<br>RSA PUBLIC KEY-----             |
| The MD5 Thumbprint (hex values can also be either space or colon separated)  | 34a629226a1529c58a1e9e09828d<br>70e4   |
| The SHA1 Thumbprint (hex values can also be either space or colon separated) | 34a629226a1529c58a1e9e09828d<br>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.

## Firewall

This section provides a complete list of the Firewall properties you can configure in the connection string for this provider.

---

| Property                         | Description   |
|----------------------------------|---|
| <a href="#">FirewallType</a>     | The protocol used by a proxy-based firewall.                      |
| <a href="#">FirewallServer</a>   | The name or IP address of a proxy-based firewall.                 |
| <a href="#">FirewallPort</a>     | The TCP port for a proxy-based firewall.                          |
| <a href="#">FirewallUser</a>     | The user name to use to authenticate with a proxy-based firewall. |
| <a href="#">FirewallPassword</a> | A password used to authenticate to a proxy-based firewall.        |

## FirewallType

The protocol used by a proxy-based firewall.

### Possible Values

NONE, TUNNEL, SOCKS4, SOCKS5

## 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 Sheets 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 <a href="#">FirewallServer</a> and <a href="#">FirewallPort</a> and passes the <a href="#">FirewallUser</a> 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 <a href="#">FirewallServer</a> and <a href="#">FirewallPort</a> . If your proxy requires authentication, set <a href="#">FirewallUser</a> and <a href="#">FirewallPassword</a> to credentials the proxy recognizes. |

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

## FirewallServer

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.

## FirewallPort

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](#).

## FirewallUser

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](#).

## FirewallPassword

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](#).

## Proxy

This section provides a complete list of the Proxy properties you can configure in the connection string for this provider.

| Property                        | Description   |
|---------------------------------|---|
| <a href="#">ProxyAutoDetect</a> | 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. |
| <a href="#">ProxyServer</a>     | The hostname or IP address of a proxy to route HTTP traffic through.  |
| <a href="#">ProxyPort</a>       | The TCP port the ProxyServer proxy is running on.   |
| <a href="#">ProxyAuthScheme</a> | The authentication type to use to authenticate to the ProxyServer proxy.  |
| <a href="#">ProxyUser</a>       | A user name to be used to authenticate to the ProxyServer proxy.  |
| <a href="#">ProxyPassword</a>   | A password to be used to authenticate to the ProxyServer proxy.   |
| <a href="#">ProxySSLType</a>    | The SSL type to use when connecting to the ProxyServer proxy.   |
| <a href="#">ProxyExceptions</a> | A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .  |

## ProxyAutoDetect

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](#).

## ProxyServer

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.

## ProxyPort

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](#).

## ProxyAuthScheme

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

### Possible Values

BASIC, DIGEST, NONE, NEGOTIATE, NTLM, PROPRIETARY

### 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](#).

## ProxyUser

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
```

## ProxyPassword

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.

## ProxySSLType

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

## Possible Values

AUTO, ALWAYS, NEVER, TUNNEL

## 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:

|               |  |
|---------------|--|
| <b>AUTO</b>   | 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. |
| <b>ALWAYS</b> | The connection is always SSL enabled.  |
| <b>NEVER</b>  | The connection is not SSL enabled.   |
| <b>TUNNEL</b> | 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.  |

## ProxyExceptions

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.

## Logging

This section provides a complete list of the Logging properties you can configure in the connection string for this provider.

---

| Property                   | Description                                  |
|----------------------------|--|
| <a href="#">LogModules</a> | Core modules to be included in the log file. |

---

## LogModules

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.

See the [Logging](#) page for an overview.

## Schema

This section provides a complete list of the Schema properties you can configure in the connection string for this provider.

| Property                             | Description   |
|--------------------------------------|---|
| <a href="#">Location</a>             | A path to the directory that contains the schema files defining tables, views, and stored procedures. |
| <a href="#">TypeDetectionScheme</a>  | Determines how to determine the data types of columns.  |
| <a href="#">Header</a>               | Indicates whether or not the first row should be used as a column header.                             |
| <a href="#">PrimaryKeyIdentifier</a> | Set this property to control the name of the primary key.   |
| <a href="#">DefineTables</a>         | Define the tables within the Google Spreadsheet.  |
| <a href="#">Orientation</a>          | Indicates whether the data in the sheet is laid out horizontally or vertically.                       |

## Location

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

## Data Type

string

## Default Value

"%APPDATA%\\CData\\GoogleSheets 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\\GoogleSheets 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                                     |

## TypeDetectionScheme

Determines how to determine the data types of columns.

## Possible Values

None, RowScan

## Data Type

string

## Default Value

"RowScan"

## Remarks

|         |   |
|---------|---|
| None    | Setting <u>TypeDetectionScheme</u> to None will return all columns as the string type. <i>Note:</i> Even when set to None, the column names will still be scanned when Header is set to True. |
| RowScan | Setting <u>TypeDetectionScheme</u> to RowScan will scan rows to heuristically determine the data type. The <a href="#">RowScanDepth</a> determines the number of rows to be scanned.          |

## Header

Indicates whether or not the first row should be used as a column header.

## Data Type

bool

## Default Value

true

## Remarks

If true, the first row will be used as a column header. Otherwise, the pseudo column names (A, B, C, etc.) will be used.

The Header property is used in conjunction with the Orientation property. When Header is set to false and Orientation is set to Columns, column names are reported as R1, R2, R3, etc.

## PrimaryKeyIdentifier

Set this property to control the name of the primary key.

## Data Type

string

## Default Value

""

## Remarks

Determines the name of the primary key column which holds the row number. The default value of the primary key is Id.

Set this property if there is a column named ID in the table you are quering, or if you prefer to change the name of the primary key.

## DefineTables

Define the tables within the Google Spreadsheet.

## Data Type

string

## Default Value

""

## Remarks

This property is used to define the ranges within a sheet that will appear as tables. The value is a comma-separated list of name-value pairs in the form [Table Name]=[Spreadsheet Name]\_[Sheet Name]\_[Range] or [Table Name]=[Spreadsheet Name]\_[Sheet Name]\_[Range]. Table Name is the name of the table you want to use for the data and will be used when issuing queries. Sheet Name is the name of the sheet within the Google Spreadsheet and Range is the range of cells that contain the data for the table.

Here is an example DefineTables value: DefineTables="Table1=Spreadsheet1\_Sheet1!A1:N25,Table2=Spreadsheet1\_Sheet2!C3:M53,Table4=xlsPcLs2-bF3AavQcSLCfzs3kGc\_Sheet4!C20:N60".

## Orientation

Indicates whether the data in the sheet is laid out horizontally or vertically.

### Possible Values

Horizontal, Vertical

### Data Type

string

### Default Value

"Vertical"

### Remarks

|            |  |
|------------|--|
| Horizontal | Specifies that the adapter operates on the rows of a sheet.    |
| Vertical   | Specifies that the adapter operates on the columns of a sheet. |

By default, the adapter models vertically oriented spreadsheet data -- rows arranged vertically below a header row.

Set this to "Horizontal" if the rows are arranged left to right. The first column contains the column names and subsequent columns become rows.

## Miscellaneous

This section provides a complete list of the Miscellaneous properties you can configure in the connection string for this provider.

| Property | Description |
|----------|-------------|
|          |             |

|                                      |   |
|--------------------------------------|---|
| <a href="#">AutoAdjustRange</a>      | If set to true, the driver will automatically expand the dimensions in case the updated/insert/deleted value is outside the range of the sheet.   |
| <a href="#">DateTimeRenderOption</a> | Determines how dates, times, and durations should be represented in the output. This is ignored if the <a href="#">ValueRenderOption</a> is <a href="#">FormattedValue</a> . The default datetime render option is <a href="#">SerialNumber</a> . |
| <a href="#">InsertDataOption</a>     | Determines how existing data is changed when new data is input.   |
| <a href="#">MaxRows</a>              | 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.   |
| <a href="#">NullValueMode</a>        | Indicates whether to read empty cells as null or as empty.  |
| <a href="#">Other</a>                | These hidden properties are used only in specific use cases.  |
| <a href="#">Pagesize</a>             | The maximum number of results to return per page from Google Sheets.  |
| <a href="#">PercentageToDecimal</a>  | Boolean determining if percentage columns should be considered as decimal.  |
| <a href="#">ReadOnly</a>             | You can use this property to enforce read-only access to Google Sheets from the provider.   |
| <a href="#">RowScanDepth</a>         | The maximum number of rows to scan to look for the columns available in a table.  |
| <a href="#">ShowEmptyRows</a>        | Indicates whether or not the empty rows should be pushed.   |
| <a href="#">Timeout</a>              | The value in seconds until the timeout error is thrown, canceling the operation.  |
| <a href="#">UseSimpleNames</a>       | Boolean determining if simple names should be used for tables and columns.  |
| <a href="#">ValueInputOption</a>     | Determines how inserted values should be treated.   |

---

**ValueRenderOption**

Determines how values should be rendered in the output.

---

## AutoAdjustRange

If set to true, the driver will automatically expand the dimensions in case the updated/insert/deleted value is outside the range of the sheet.

### Data Type

bool

### Default Value

false

### Remarks

If set to true, the driver will automatically expand the dimensions in case the updated/insert/deleted value is outside the range of the sheet.

## DateTimeRenderOption

Determines how dates, times, and durations should be represented in the output. This is ignored if the ValueRenderOption is FormattedValue. The default datetime render option is SerialNumber.

### Possible Values

SerialNumber, FormattedString

### Data Type

string

### Default Value

"SerialNumber"

---



## Remarks

|                 |   |
|-----------------|---|
| SerialNumber    | Instructs the adapter to output date, time, datetime, and duration fields as doubles in "serial number" format, as popularized by Lotus 1-2-3. The whole number portion of the value (left of the decimal) counts the days since December 30th 1899. The fractional portion (right of the decimal) counts the time as a fraction of the day. For example, January 1st 1900 at noon would be 2.5, 2 because it's 2 days after December 30st, 1899, and .5 because noon is half a day. February 1st, 1900 at 3pm would be 33.625. This correctly treats the year 1900 as not a leap year. |
| FormattedString | Instructs the adapter to output date, time, datetime, and duration fields as strings in their given number format (which is dependent on the spreadsheet locale).   |

## InsertDataOption

Determines how existing data is changed when new data is input.

### Possible Values

Overwrite, InsertRows

### Data Type

string

### Default Value

"Overwrite"

## Remarks

|           |  |
|-----------|--|
| Overwrite | The new data replaces the contents of the row after the last row in the table. |
|-----------|--|

---

Note that this could potentially overwrite data after the last row in the table, as the adapter stops returning rows if it encounters a blank row.

See [Tables](#) for more information on how the adapter discovers tables from the spreadsheet data.

---

|            |  |
|------------|--|
| InsertRows | The adapter will insert a new row at the line specified (or at the end of the table). This avoids overwriting data below the table by incrementing the Ids of all rows below by one. It also allows you to insert data between existing rows -- the following query inserts a new row 2. The existing row 2 becomes row 3, row 3 becomes row 4, and so on. |
|------------|--|

```
INSERT INTO Spreadsheet1_Sheet1(Id,Name, Amount) VALUES
(2,'Test', 10)
```

---

## MaxRows

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.

## NullValueMode

Indicates whether to read empty cells as null or as empty.

## Possible Values

ReadAsNull, ReadAsEmpty

## Data Type

string

## Default Value

"ReadAsNull"

## Remarks

NullValueMode controls how empty Google Sheets cells are modelled. An empty cell is a cell that has not been set and thus contains a null string. If NullValueMode is set to ReadAsNull, NULL is reported for an empty cell; if NullValueMode is set to ReadAsEmpty, an empty string is reported for an empty cell.

## 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.   |

## Pagesize

The maximum number of results to return per page from Google Sheets.

## Data Type

int

## Default Value

1000

## Remarks

The Pagesize property affects the maximum number of results to return per page from Google Sheets. Setting a higher value may result in better performance at the cost of additional memory allocated per page consumed.

## PercentageToDecimal

Boolean determining if percentage columns should be considered as decimal.

## Data Type

bool

## Default Value

false

## Remarks

Boolean determining if percentage columns should be considered as decimal.

## Readonly

You can use this property to enforce read-only access to Google Sheets 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.

## RowScanDepth

The maximum number of rows to scan to look for the columns available in a table.

## Data Type

int

## Default Value

50

## Remarks

The columns in a table must be determined by scanning table rows. This value determines the maximum number of rows that will be scanned.

Setting a high value may decrease performance. Setting a low value may prevent the data type from being determined properly, especially when there is null data.

## ShowEmptyRows

Indicates whether or not the empty rows should be pushed.

## Data Type

bool

## Default Value

false

## Remarks

If true, the empty rows will be pushed at the output.

## 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.

## UseSimpleNames

Boolean determining if simple names should be used for tables and columns.

## Data Type

bool

## Default Value

false

## Remarks

Google Sheets tables and columns can use special characters in names that are normally not allowed in standard databases. UseSimpleNames makes the adapter easier to use with traditional database tools.

Setting UseSimpleNames to true will simplify the names of tables and columns returned. It will enforce a naming scheme such that only alphanumeric characters and the underscore are valid for the displayed table and column names. Any nonalphanumeric characters will be converted to an underscore.

## ValueInputOption

Determines how inserted values should be treated.

## Possible Values

UserEntered, Raw

## Data Type

string

## Default Value

"UserEntered"

## Remarks

|             |   |
|-------------|---|
| Raw         | The values the user has entered will not be parsed and will be stored as-is.  |
| UserEntered | The values will be parsed as if the user typed them into the UI. Numbers will stay as numbers, but strings may be converted to numbers, dates, etc. -- following the same rules that are applied when entering text into a cell via the Google Sheets UI. |

## ValueRenderOption

Determines how values should be rendered in the output.

## Possible Values

FormattedValue, UnformattedValue, Formula

## Data Type

string

## Default Value

"FormattedValue"



## Remarks

|                  |  |
|------------------|--|
| FormattedValue   | Values will be calculated and formatted in the reply according to the cell's formatting. Formatting is based on the spreadsheet's locale, not the requesting user's locale. For example, if A1 is "1.23" and A2 is "=A1" and formatted as currency, then A2 would return "\$1.23". |
| UnformattedValue | Values will be calculated, but not formatted in the reply. For example, if A1 is "1.23" and A2 is "=A1" and formatted as currency, then A2 would return the number "1.23".   |
| Formula          | Values will not be calculated. The reply will include the formulas. For example, if A1 is "1.23" and A2 is "=A1" and formatted as currency, then A2 would return "=A1".  |

# 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](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.