



# **TIBCO® Data Virtualization**

## **Facebook Adapter Guide**

Version 8.7.0 | October 2023

# Contents

---

<b>Contents</b>	<b>2</b>
<b>Facebook Adapter</b>	<b>4</b>
Getting Started	4
Basic Tab	5
Logging	9
Creating a Custom OAuth App	11
Connecting to Facebook	12
Insight Mapping	14
Changelog	32
Advanced Features	38
User Defined Views	39
SSL Configuration	41
Firewall and Proxy	42
Query Processing	42
Logging	43
SQL Compliance	46
SELECT Statements	47
SELECT INTO Statements	49
INSERT Statements	50
UPDATE Statements	51
DELETE Statements	51
EXECUTE Statements	52
PIVOT and UNPIVOT	53
Data Model	54
Tables	54
Views	62
Stored Procedures	125

Connection String Options .....	132
Authentication .....	136
OAuth .....	137
SSL .....	145
Firewall .....	146
Proxy .....	150
Logging .....	156
Schema .....	157
Miscellaneous .....	158
<b>TIBCO Product Documentation and Support Services .....</b>	<b>165</b>
How to Access TIBCO Documentation .....	165
How to Contact TIBCO Support .....	166
Release Version Support .....	166
How to Join TIBCO Community .....	167
<b>Legal and Third-Party Notices .....</b>	<b>168</b>

# Facebook Adapter

---

## Facebook Version Support

By default, the adapter models the current version of the Facebook Graph API, as bidirectional tables. Older versions can be specified in the [Version](#) property.

## SQL Compliance

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

# Getting Started

## Connecting to Facebook

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

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

1. Unzip the `tdv.facebook.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.facebook/tdv.facebook.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 Facebook
```

## Basic Tab

### Connecting to Facebook

The following are optional connection properties:

- Target: Some Facebook tables can be filtered by a target. For example, to retrieve comments on a video, specify the Id of the video as the target. This property enables you to restrict the results of all queries in the connection to records that match the specified target. You can also specify this restriction per query with the Target column.
- AggregateFormat: The adapter returns some columns as a string aggregate. For example, the available likes data for an entity is returned in aggregate. By default, the adapter returns aggregate columns in JSON. You can also return aggregates in XML.
- Version: Set this property to the Facebook API version if you need to work with a different version than the default.

### Authenticating to Facebook

Facebook uses the OAuth standard to authenticate users.

#### OAuth

In all OAuth flows, you must set AuthScheme to **OAuth**. The sections below assume that you have done so.

### Web Applications

When connecting via a Web application, you need to create and register a custom OAuth application with Facebook. See [Creating a Custom OAuth App](#) for more information about

custom applications. You can then use the adapter to get and manage the OAuth token values. **Get an OAuth Access Token**

Set the following connection properties to obtain the OAuthAccessToken:

- AuthScheme: Set this to **AzureAD**.
- OAuthClientId: Set this to the client Id in your application settings.
- OAuthClientSecret: Set this to the client secret in your application settings
- Scope (optional): Set this if you need to customize the permissions that the driver requests.
- AuthenticateAsPage (optional): Set this to a page name or Id to make requests as a page. The page must be managed by the authenticated user.

Then call stored procedures to complete the OAuth exchange:

1. Call the [GetOAuthAuthorizationURL](#) stored procedure. Set the AuthMode input to **WEB** and set the CallbackURL input to the Redirect URI you specified in your app settings. If necessary, set the "Permissions" parameter to request custom permissions.
2. Open the URL, log in, and authorize the application. You are redirected back to the callback URL.
3. Call the [GetOAuthAccessToken](#) stored procedure. Set the AuthMode input to **WEB**. Set the Verifier input to the "code" parameter in the query string of the callback URL. If necessary, set the "Permissions" parameter to request custom permissions.

To connect to data, set the OAuthAccessToken connection property to the access token returned by the stored procedure. When the access token expires after ExpiresIn seconds, call [GetOAuthAccessToken](#) again to obtain a new access token.

## Headless Machines

To configure the driver to use OAuth with a user account on a headless machine, you need to authenticate on another device that has an internet browser.

Creating a custom OAuth app is optional in the headless OAuth flow; you can skip creating an app by connecting with the driver's embedded OAuth credentials. You might want to create a custom OAuth app to change the information displayed when users log into Facebook to grant permissions to the driver. See [Creating a Custom OAuth App](#) for more information about creating custom OAuth applications.

1. Choose one of these two options:

- Option 1: Obtain the OAuthVerifier value as described in "Obtain and Exchange a Verifier Code" below.
- Option 2: Install the adapter on another machine and transfer the OAuth authentication values after you authenticate through the usual browser-based flow, as described in "Transfer OAuth Settings" below.

2. Then configure the adapter to automatically refresh the access token from the headless machine.

This section describes the procedure to authenticate and connect to data.

### **Option 1: Obtain and Exchange a Verifier Code**

To obtain a verifier code, you must authenticate at the OAuth authorization URL. Follow the steps below to authenticate from the machine with an internet browser and obtain the OAuthVerifier connection property.

1. If you are using the Embedded OAuth Application:

- a. Call the [GetOAuthAuthorizationURL](#) stored procedure.
- b. Click [Facebook OAuth endpoint](#) to open the endpoint in your browser.

2. If you are using a custom OAuth application, create the Authorization URL by setting the following properties:

- InitiateOAuth: Set to **OFF**.
- OAuthClientId: Set to the client Id assigned when you registered your application.
- OAuthClientSecret: Set to the client secret assigned when you registered your application.

3. Call the [GetOAuthAuthorizationURL](#) stored procedure with the CallbackURI input parameter set to the exact Redirect URI you specified in your application settings.

4. Open the URL returned by the stored procedure in a browser.

5. Log in and grant permissions to the adapter. You are then redirected to the callback URL, which contains the verifier code.

6. Save the value of the verifier code. Later you will set this in the OAuthVerifier connection property.

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

- OAuthClientId: Set to the Client ID in your OAuth Integration settings.
- OAuthClientSecret: Set to the Client Secret in your OAuth Integration settings.
- OAuthVerifier: Set to the verifier code.
- OAuthSettingsLocation: Set to persist the encrypted OAuth authentication values to the specified file.
- InitiateOAuth: Set to **REFRESH**.

### Connect to Data

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

- OAuthSettingsLocation: Set 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 to **REFRESH**.

### Option 2: Transfer OAuth Settings

To install the adapter on another machine, authenticate, and then transfer the resulting OAuth values:

1. On a second machine, install the adapter and connect with the following properties set:
  - OAuthSettingsLocation: Set to a writable text file.
  - InitiateOAuth: Set to **GETANDREFRESH**.
  - OAuthClientId: Set to the client ID in your app settings.
  - OAuthClientSecret: Set to the client secret in your app settings.
  - CallbackURL: Set to the callback URL in your app settings.
2. Test the connection to authenticate. The resulting authentication values are written and encrypted to the path specified by OAuthSettingsLocation. After 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 to **REFRESH**.



- OAuthSettingsLocation: Set 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.

## Requesting Additional Permissions

You may find while using the adapter that Facebook returns an error stating your app does not have permissions to do a certain action. To resolve this, you must generate a new OAuth access token with the required permissions. Set the Scope property in the authentication step. You can find a list of available Facebook permissions here:

<http://developers.facebook.com/docs/authentication/permissions/>

## AuthenticateAsPage Property

Use the AuthenticateAsPage connection property if you want to post as a single page. To query collections of pages, leave AuthenticateAsPage blank, in which case CData tools automatically detect which page tokens to use.

The following sections compare the two options.

### Posting as a Page

After authenticating to Facebook with your user account, you can post, etc. as one of the pages you manage: Set the AuthenticateAsPage property to the Id of the page you want. You can find the Ids for all pages your account has access to by querying the [Pages](#) view.

### Automatic Page

Facebook has made a number of recent changes that require page tokens for most resources owned by a page. This can be troublesome if you manage multiple pages and want to execute the same queries across all pages (such as retrieving Insights). In order to make this work seamlessly with our tools, we have added a way to automatically detect the page token to use. For this to work, simply do not specify the AuthenticateAsPage. Note that the correct page token can only be resolved if the page id is specified as part of the target in the request. This means for some requests you will still need to manually specify AuthenticateAsPage.

## 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 an explanation 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 Facebook 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**.

## Creating a Custom OAuth App

### Create a Custom OAuth App

To obtain the OAuth client credentials, follow the steps below:

1. Log into Facebook and navigate to <https://developers.facebook.com/apps>.
2. Create a new app and select **Settings > Basic**. The OAuthClientId is the App Id displayed. The OAuthClientSecret is the App Secret.
3. Add a website platform on the Settings tab. Enter a Site URL. This value is not used in authentication.

Follow the steps below to configure the OAuth redirect URI.

1. Go to your app settings and add the Facebook Login product from the Products

section.

2. In the product settings, define the OAuth redirect URI.

Set the redirect URI:

- For desktop applications, set the redirect URI to <https://localhost:33333/>, or a similar https url.

## Connecting to Facebook

OAuth enables the adapter to obtain limited access to the service on behalf of a user by orchestrating an approval interaction between the user and the service. This section specifies how the adapter can be configured to authenticate using OAuth.

### Configuring the Adapter

The adapter can be configured for OAuth in three ways using the [InitiateOAuth](#) property.

#### Set InitiateOAuth to OFF

In this configuration, the adapter uses the [OAuthAccessToken](#) for authorization without initiating the OAuth flow. When the [OAuthAccessToken](#) expires, the user must obtain a new [OAuthAccessToken](#).

This configuration is suitable for one-time use or when the [OAuthAccessToken](#) has a long life. This is the simplest way of configuring OAuth if the [OAuthAccessToken](#) can be obtained using other means. For example, with the help of the API or developer console.

#### Set InitiateOAuth to REFRESH

The REFRESH configuration does not require user interaction. This configuration is suitable for scenarios where the studio and server are not on the same machine.

The adapter refreshes the [OAuthAccessToken](#) when it expires. The adapter stores the new [OAuthAccessToken](#) in the [OAuthSettingsLocation](#) configured in the data source.

The following connection properties need to be set:

- [OAuthRefreshToken](#)
- [OAuthClientId](#)

- OAuthClientSecret
- OAuthSettingsLocation

## Set InitiateOAuth to GETANDREFRESH

This configuration requires interaction between the adapter, user, and Facebook.

The adapter launches the browser to allow the user to log in and grant permissions. Since this configuration requires browser interaction, there are limitations on when this configuration can be used. For example, this configuration cannot be used when the TDV monitor is used to start the server.

The following connection properties need to be set:

- OAuthClientId
- OAuthClientSecret
- CallbackURL
- OAuthSettingsLocation

## Configuring a Development Machine

To configure the adapter on a development machine, where the server and studio are running on the same machine, start TDV from a console and set InitiateOAuth to GETANDREFRESH. The resulting OAuthAccessToken is saved in OAuthSettingsLocation.

## Configuring a Production Machine

If the imported data source can be used on the target server without additional modifications, copy the settings file to the target server. Note that the OAuthSettingsLocation must not change.

Alternatively, a refresh token can be obtained from Facebook and can be provided along with the OAuthClientId and OAuthClientSecret. Getting a refresh token requires experience in using the developer APIs and console of the OAuth provider and the token has to be obtained using the same OAuthClientId and OAuthClientSecret configured in the data source.

If Facebook issues a long-lived access token, use the Facebook developer API or console to retrieve the OAuthAccessToken.

## Configuring OAuth in a Cluster

In a cluster, the data source configuration is synced across the members of the cluster. Set the OAuthSettingsLocation to a file path that is on a shared file system that is accessible to all the members.

## Importing/Exporting Archives and Using the Deployment Manager

When the archive contains a data source configured for OAuth and when it is imported or migrated to the target server, the OAuthSettingsLocation is not automatically imported or migrated. The OAuthSettingsLocation needs to be externally migrated or the OAuth flow has to be reinitiated in the target server.

## Insight Mapping

Below is a mapping of specific Facebook insights, which periods are available for them, what view they can be used from, and what types of targets are available. Since Facebook frequently makes changes to the available insights, we also include a table indicating which insights have been removed recently from Facebook, and the previous name for an insight if it has been renamed.

### Current Insights

View or Stored Procedure	Insight Name	Available Periods	Target Type	Previous Name
<a href="#">SimpleInsights</a>	PAGE_ACTIONS_ POST_ REACTIONS_ ANGER_TOTAL	day	page	
<a href="#">SimpleInsights</a>	PAGE_ACTIONS_ POST_ REACTIONS_ HAHA_TOTAL	day	page	
<a href="#">SimpleInsights</a>	PAGE_ACTIONS_ POST_ REACTIONS_	day	page	

	LIKE_TOTAL		
SimpleInsights	PAGE_ACTIONS_ POST_ REACTIONS_ LOVE_TOTAL	day	page
SimpleInsights	PAGE_ACTIONS_ POST_ REACTIONS_ SORRY_TOTAL	day	page
SimpleInsights	PAGE_ACTIONS_ POST_ REACTIONS_ WOW_TOTAL	day	page
SimpleInsights	PAGE_ CONSUMPTIONS	day, week, days_28	page
SimpleInsights	PAGE_ CONSUMPTION S_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_ENGAGED_ USERS	day, week, days_28	page
SimpleInsights	PAGE_FAN_ADDS	day	page
SimpleInsights	PAGE_FAN_ ADDS_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_FAN_ REMOVES	day	page
SimpleInsights	PAGE_FAN_ REMOVES_	day	page

	UNIQUE		
SimpleInsights	PAGE_FANS	day	page
SimpleInsights	PAGE_FANS_ONLINE	day	page
SimpleInsights	PAGE_FANS_ONLINE_PER_DAY	day	page
SimpleInsights	PAGE_IMPRESSIONS	day, week, days_28	page
SimpleInsights	PAGE_IMPRESSIONS_FREQUENCY_DISTRIBUTION	day, week, days_28	page
SimpleInsights	PAGE_IMPRESSIONS_ORGANIC	day, week, days_28	page
SimpleInsights	PAGE_IMPRESSIONS_ORGANIC_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_IMPRESSIONS_PAID	day, week, days_28	page
SimpleInsights	PAGE_IMPRESSIONS_PAID_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_IMPRESSIONS_UNIQUE	day, week, days_28	page



SimpleInsights	PAGE_ IMPRESSIONS_ VIRAL	day, week, days_28	page
SimpleInsights	PAGE_ IMPRESSIONS_ VIRAL_ FREQUENCY_ DISTRIBUTION	day, week, days_28	page
SimpleInsights	PAGE_ IMPRESSIONS_ VIRAL_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_ NEGATIVE_ FEEDBACK	day, week, days_28	page
SimpleInsights	PAGE_ NEGATIVE_ FEEDBACK_ UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_PLACES_ CHECKIN_ MOBILE	day, week, days_28	page
SimpleInsights	PAGE_PLACES_ CHECKIN_ MOBILE_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_PLACES_ CHECKIN_TOTAL	day, week, days_28	page
SimpleInsights	PAGE_PLACES_ CHECKIN_ TOTAL_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_POST_	day,	page

	ENGAGEMENTS	week, days_28		
SimpleInsights	PAGE_POSTS_ IMPRESSIONS	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ ORGANIC	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ ORGANIC_ UNIQUE	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ PAID	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ PAID_UNIQUE	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ UNIQUE	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ VIRAL	day, week, days_28	page	
SimpleInsights	PAGE_POSTS_ IMPRESSIONS_ VIRAL_UNIQUE	day, week, days_28	page	
SimpleInsights	PAGE_CONTENT_ ACTIVITY	day, week, days_28	page	PAGE_STORIES

SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S_AUTOPLAYED	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S_CLICK_TO_PLAY	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S_ORGANIC	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S_PAID	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S_REPEAT_VIEWS	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_COMPLETE_VIEWS_30S_UNIQUE	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_REPEAT_VIEWS	day, week, days_28	page
SimpleInsights	PAGE_VIDEO_VIEWS	day, week, days_28	page

SimpleInsights	PAGE_VIDEO_VIEWS_AUTOPLAYED	day, week, days_28	page	
SimpleInsights	PAGE_VIDEO_VIEWS_CLICK_TO_PLAY	day, week, days_28	page	
SimpleInsights	PAGE_VIDEO_VIEWS_ORGANIC	day, week, days_28	page	
SimpleInsights	PAGE_VIDEO_VIEWS_PAID	day, week, days_28	page	
SimpleInsights	PAGE_VIDEO_VIEWS_UNIQUE	day, week, days_28	page	
SimpleInsights	PAGE_VIEWS_TOTAL	day, week, days_28	page	PAGE_VIEWS
SimpleInsights	PAGE_VIEWS_LOGGED_IN_TOTAL	day, week, days_28	page	PAGE_VIEWS_LOGIN
SimpleInsights	PAGE_VIEWS_LOGGED_IN_UNIQUE	day, week, days_28	page	PAGE_VIEWS_LOGIN_UNIQUE
SimpleInsights	PAGE_VIEWS_LOGOUT	day	page	
SimpleInsights	POST_CLICKS	lifetime	post	POST_CONSUMPTIONS
SimpleInsights	POST_CLICKS_UNIQUE	lifetime	post	POST_CONSUMPTION

S_UNIQUE			
SimpleInsights	POST_ENGAGED_USERS	lifetime	post
SimpleInsights	POST_IMPRESSIONS	lifetime	post
SimpleInsights	POST_IMPRESSIONS_FAN	lifetime	post
SimpleInsights	POST_IMPRESSIONS_FAN_PAID	lifetime	post
SimpleInsights	POST_IMPRESSIONS_FAN_PAID_UNIQUE	lifetime	post
SimpleInsights	POST_IMPRESSIONS_FAN_UNIQUE	lifetime	post
SimpleInsights	POST_IMPRESSIONS_ORGANIC	lifetime	post
SimpleInsights	POST_IMPRESSIONS_ORGANIC_UNIQUE	lifetime	post
SimpleInsights	POST_IMPRESSIONS_PAID	lifetime	post
SimpleInsights	POST_IMPRESSIONS_	lifetime	post

	PAID_UNIQUE			
SimpleInsights	POST_ IMPRESSIONS_ UNIQUE	lifetime	post	
SimpleInsights	POST_ IMPRESSIONS_ VIRAL	lifetime	post	
SimpleInsights	POST_ IMPRESSIONS_ VIRAL_UNIQUE	lifetime	post	
SimpleInsights	POST_ NEGATIVE_ FEEDBACK	lifetime	post	
SimpleInsights	POST_ NEGATIVE_ FEEDBACK_ UNIQUE	lifetime	post	
SimpleInsights	POST_ACTIVITY	lifetime	post	POST_STORIES
SimpleInsights	POST_ACTIVITY_ UNIQUE	lifetime	post	POST_ STORYTELLERS
SimpleInsights	POST_VIDEO_ AVG_TIME_ WATCHED	lifetime	post	
SimpleInsights	POST_VIDEO_ COMPLETE_ VIEWS_ORGANIC	lifetime	post	
SimpleInsights	POST_VIDEO_ COMPLETE_ VIEWS_ ORGANIC_	lifetime	post	

	UNIQUE		
SimpleInsights	POST_VIDEO_COMPLETE_VIEWS_PAID	lifetime	post
SimpleInsights	POST_VIDEO_COMPLETE_VIEWS_PAID_UNIQUE	lifetime	post
SimpleInsights	POST_VIDEO_LENGTH	lifetime	post
SimpleInsights	POST_VIDEO_VIEW_TIME	lifetime	post
SimpleInsights	POST_VIDEO_VIEW_TIME_ORGANIC	lifetime	post
SimpleInsights	POST_VIDEO_VIEWS_10S	lifetime	post
SimpleInsights	POST_VIDEO_VIEWS_10S_AUTOPLAYED	lifetime	post
SimpleInsights	POST_VIDEO_VIEWS_10S_CLICKED_TO_PLAY	lifetime	post
SimpleInsights	POST_VIDEO_VIEWS_10S_ORGANIC	lifetime	post
SimpleInsights	POST_VIDEO_VIEWS_10S_PAID	lifetime	post

SimpleInsights	POST_VIDEO_VIEWS_10S_SOUND_ON	lifetime	post	
SimpleInsights	POST_VIDEO_VIEWS_10S_UNIQUE	lifetime	post	
SimpleInsights	POST_VIDEO_VIEWS_ORGANIC	lifetime	post	
SimpleInsights	POST_VIDEO_VIEWS_ORGANIC_UNIQUE	lifetime	post	
SimpleInsights	POST_VIDEO_VIEWS_PAID	lifetime	post	
SimpleInsights	POST_VIDEO_VIEWS_PAID_UNIQUEPOST_VIDEO_VIEWS_SOUND_ON	lifetime	post	
InsightsByConsumptionType	PAGE_CONSUMPTIONS_BY_CONSUMPTION_TYPE	day, week, days_28	page	
InsightsByConsumptionType	PAGE_CONSUMPTIONS_BY_CONSUMPTION_TYPE_UNIQUE	day, week, days_28	page	
InsightsByConsumptionType	POST_CLICKS_BY_TYPE	lifetime	post	POST_CONSUMPTIONS_BY_



				CONSUMPTION_ TYPE
<a href="#">InsightsByConsumptionType</a>	POST_CLICKS_ BY_TYPE_ UNIQUE	lifetime	post	POST_ CONSUMPTION S_BY_ CONSUMPTION_ TYPE_UNIQUE
<a href="#">InsightsByFeedbackType</a>	PAGE_ NEGATIVE_ FEEDBACK_BY_ TYPE	day, week, days_28	page	
<a href="#">InsightsByFeedbackType</a>	PAGE_ NEGATIVE_ FEEDBACK_BY_ TYPE_UNIQUE	day, week, days_28	page	
<a href="#">InsightsByFeedbackType</a>	PAGE_POSITIVE_ FEEDBACK_BY_ TYPE	day, week, days_28	page	
<a href="#">InsightsByFeedbackType</a>	PAGE_POSITIVE_ FEEDBACK_BY_ TYPE_UNIQUE	day, week, days_28	page	
<a href="#">InsightsByFeedbackType</a>	POST_ NEGATIVE_ FEEDBACK_BY_ TYPE	lifetime	post	
<a href="#">InsightsByFeedbackType</a>	POST_ NEGATIVE_ FEEDBACK_BY_ TYPE_UNIQUE	lifetime	post	
<a href="#">InsightsByLikeSourceType</a>	PAGE_FANS_BY_ LIKE_SOURCE	day	page	

InsightsByLikeSourceType	PAGE_FANS_BY_LIKE_SOURCE_UNIQUE	day	page	
InsightsByUnLikeSourceType	PAGE_FANS_BY_UNLIKE_SOURCE	day	page	
InsightsByUnLikeSourceType	PAGE_FANS_BY_UNLIKE_SOURCE_UNIQUE	day	page	
InsightsByReactionTotals	PAGE_ACTIONS_POST_REACTIONS_TOTAL	day	page	
InsightsByReactionTotals	POST_REACTIONS_BY_TYPE_TOTAL	day	page	
InsightsByStoryType	PAGE_CONTENT_ACTIVITY_BY_ACTION_TYPE	day, week, days_28	page	PAGE_STORIES_BY_STORY_TYPE
InsightsByStoryType	PAGE_IMPRESSIONS_BY_STORY_TYPE	day, week, days_28	page	
InsightsByStoryType	PAGE_IMPRESSIONS_BY_STORY_TYPE_UNIQUE	day, week, days_28	page	
InsightsByStoryType	POST_IMPRESSIONS_BY_STORY_TYPE	day, week, days_28	page	
InsightsByStoryType	POST_IMPRESSIONS_BY_STORY_	day, week, days_28	page	

	TYPE_UNIQUE		
<a href="#">InsightsByTabType</a>	PAGE_TAB_VIEWS_LOGIN_TOP_UNIQUE	day, week	page
<a href="#">InsightsByTabType</a>	PAGE_TAB_VIEWS_LOGIN_TOP	day, week	page
<a href="#">InsightsByTabType</a>	PAGE_TAB_VIEWS_LOGOUT_TOP	day	page
<a href="#">SimpleVideoInsights</a>	PAGE_VIDEO_VIEW_TIME	day	page
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_VIEWS	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_VIEWS_UNIQUE	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_VIEWS_AUTOPLAYED	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_VIEWS_CLICKED_TO_PLAY	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_VIEWS_SOUND_ON	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_COMPLETE_VIEWS	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_VIEWS	lifetime	video

	COMPLETE_VIEWS_UNIQUE		
SimpleVideoInsights	TOTAL_VIDEO_COMPLETE_VIEWS_AUTO_PLAYED	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_COMPLETE_VIEWS_CLICKED_TO_PLAY	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_10S_VIEWS	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_10S_VIEWS_UNIQUE	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_10S_VIEWS_auto_played	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_10S_VIEWS_CLICKED_TO_PLAY	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_10S_VIEWS_SOUND_ON	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_AVG_TIME_WATCHED	lifetime	video
SimpleVideoInsights	TOTAL_VIDEO_VIEW_TOTAL_TIME	lifetime	video

<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_IMPRESSIONS	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_IMPRESSIONS_UNIQUE	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_IMPRESSIONS_VIRAL_UNIQUE	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_IMPRESSIONS_VIRAL	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_IMPRESSIONS_FAN_UNIQUE	lifetime	video
<a href="#">SimpleVideoInsights</a>	TOTAL_VIDEO_IMPRESSIONS_FAN	lifetime	video
<a href="#">VideoInsightsByActionType</a>	TOTAL_VIDEO_STORIES_BY_ACTION_TYPE	lifetime	video
<a href="#">VideoInsightsByDistributionType</a>	TOTAL_VIDEO_VIEWS_BY_DISTRIBUTION_TYPE	lifetime	video
<a href="#">VideoInsightsByDistributionType</a>	TOTAL_VIDEO_VIEW_TIME_BY_DISTRIBUTION_TYPE	lifetime	video
<a href="#">VideoInsightsByReactionType</a>	TOTAL_VIDEO_REACTIONS_BY_TYPE_TOTAL	lifetime	video

CreateInsightSchema	PAGE_CONTENT_ ACTIVITY_BY_ AGE_GENDER_ UNIQUE	day, week, days_28	page	PAGE_ STORYTELLERS_ BY_AGE_GENDER
CreateInsightSchema	PAGE_CONTENT_ ACTIVITY_BY_ CITY_UNIQUE	day, week, days_28	page	PAGE_ STORYTELLERS_ BY_CITY
CreateInsightSchema	PAGE_CONTENT_ ACTIVITY_BY_ COUNTRY_ UNIQUE	day, week, days_28	page	PAGE_ STORYTELLERS_ BY_COUNTRY
CreateInsightSchema	PAGE_CONTENT_ ACTIVITY_BY_ LOCALE_UNIQUE	day, week, days_28	page	PAGE_ STORYTELLERS_ BY_LOCALE
CreateInsightSchema	PAGE_ IMPRESSIONS_ BY_CITY_UNIQUE	day, week, days_28	page	
CreateInsightSchema	PAGE_ IMPRESSIONS_ BY_COUNTRY_ UNIQUE	day, week, days_28	page	
CreateInsightSchema	PAGE_ IMPRESSIONS_ BY_LOCALE_ UNIQUE	day, week, days_28	page	
CreateInsightSchema	PAGE_ IMPRESSIONS_ BY_AGE_ GENDER_UNIQUE	day, week, days_28	page	
CreateInsightSchema	PAGE_PLACES_ CHECKINS_BY_ AGE_GENDER	day	page	

CreateInsightSchema	PAGE_PLACES_ CHECKINS_BY_ LOCALE	day	page
CreateInsightSchema	PAGE_PLACES_ CHECKINS_BY_ COUNTRY	day	page
CreateInsightSchema	PAGE_FANS_ LOCALE	day	page
CreateInsightSchema	PAGE_FANS_CITY	day	page
CreateInsightSchema	PAGE_FANS_ COUNTRY	day	page
CreateInsightSchema	PAGE_FANS_ GENDER_AGE	day	page
CreateInsightSchema	PAGE_VIEWS_ EXTERNAL_ REFERRALS	day	page
CreateInsightSchema	POST_STORIES_ BY_ACTION_TYPE	lifetime	post
CreateInsightSchema	POST_ STORYTELLERS_ BY_ACTION_TYPE	lifetime	post
CreateInsightSchema	PAGE_POSTS_ IMPRESSIONS_ FREQUENCY_ DISTRIBUTION	day, week, days_28	post
CreateInsightSchema	POST_VIDEO_ RETENTION_ GRAPH	lifetime	post

## Insights Removed By Facebook

View or Stored Procedure	Insight Name	Available Periods	Target Type
<a href="#">SimpleInsights</a>	PAGE_VIEWS_UNIQUE	day, week	page
InsightsByPaidStatus	PAGE_STORYTELLERS_BY_STORY_TYPE	day, week, days_28	page
InsightsByPaidStatus	POST_IMPRESSIONS_BY_PAID_NON_PAID	lifetime	page
InsightsByPaidStatus	PAGE_IMPRESSIONS_BY_PAID_NON_PAID	day, week, days_28	page
InsightsByPaidStatus	PAGE_IMPRESSIONS_BY_PAID_NON_PAID_UNIQUE	day, week, days_28	page
InsightsByPaidStatus	POST_IMPRESSIONS_BY_PAID_NON_PAID_UNIQUE	lifetime	page
InsightsByPaidStatus	PAGE_POSTS_IMPRESSIONS_BY_PAID_NON_PAID	day, week, days_28	page
InsightsByPaidStatus	PAGE_POSTS_IMPRESSIONS_BY_PAID_NON_PAID_UNIQUE	day, week, days_28	page

## Changelog

### General Changes

Date	Build Number	Change Type	Description



01/10/2022	8411	Facebook	<p><b>Removed</b></p> <ul style="list-style-type: none"> <li>Added the Attachments aggregate field for the Posts view, and removed fields from the Posts view that would have previously been associated with only a single attachment for a post. Use the StoryAttachments view to access individual post attachment data from the Attachments aggregate field.</li> </ul> <p><b>Removed</b></p> <ul style="list-style-type: none"> <li>Added the Attachments aggregate field for the Comments view, and removed fields from the Comments view that would have previously been associated with only a single attachment for a comment. Use the StoryAttachments view to access individual comment attachment data from the Attachments aggregate field.</li> </ul>
12/28/2022	8398	Facebook	<p><b>Added</b></p> <ul style="list-style-type: none"> <li>Added the StoryAttachments view, which provides individual rows for each attachment of a set of attachments associated with a post or comment.</li> </ul> <p><b>Removed</b></p> <ul style="list-style-type: none"> <li>Added the Attachments aggregate field for the Wall view, and removed fields from the Wall view that would have previously been associated with only a single attachment for a Wall post.</li> </ul>
12/14/2022	8383	General	<p><b>Changed</b></p> <ul style="list-style-type: none"> <li>Added the Default column to the sys_procedureparameters table.</li> </ul>

12/9/2022	8379	Facebook	<b>Changed</b> <ul style="list-style-type: none"> <li>Changed the API version default of the Version property from 14.0 to 15.0.</li> </ul>
10/12/2022	8320	Facebook	<b>Added</b> <ul style="list-style-type: none"> <li>Added FileStream input attribute to add output streams to the CreateInsightSchema stored procedure.</li> </ul>
09/30/2022	8308	General	<b>Changed</b> <ul style="list-style-type: none"> <li>Added the IsPath column to the sys_procedureparameters table.</li> </ul>
08/17/2022	8264	General	<b>Changed</b> <ul style="list-style-type: none"> <li>We now support handling the keyword "COLLATE" as standard function name as well.</li> </ul>
11/02/2021	7976	Facebook	<b>Changed</b> <ul style="list-style-type: none"> <li>Updated the maximum pagesize to be unrestricted for most tables. Previously Facebook would throw exceptions if the pagesize was exceeded, but this seems to have been removed.</li> </ul>
10/26/2021	7969	Facebook	<b>Changed</b> <ul style="list-style-type: none"> <li>The default API version is updated to 12.0.</li> </ul> <b>Removed</b> <ul style="list-style-type: none"> <li>Removed the CreateAppInsights stored procedure. It is now deprecated and scheduled for removal by Facebook.</li> </ul>
09/27/2021	7940	Facebook	<b>Added</b> <ul style="list-style-type: none"> <li>Added extra columns for Dimension,</li> </ul>

Hierarchy and OLAPType to sys_tablecolumns for OLAP properties.			
09/02/2021	7915	General	<b>Added</b> <ul style="list-style-type: none"> <li>Added support for the STRING_SPLIT table-valued function in the CROSS APPLY clause.</li> </ul>
08/07/2021	7889	General	<b>Changed</b> <ul style="list-style-type: none"> <li>Added the KeySeq column to the sys_foreignkeys table.</li> </ul>
08/06/2021	7888	General	<b>Changed</b> <ul style="list-style-type: none"> <li>Added the new sys_primarykeys system table.</li> </ul>
07/23/2021	7874	General	<b>Changed</b> <ul style="list-style-type: none"> <li>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 &gt; LAST_MONTH(). Formerly the two LAST_MONTH() methods would resolve to different datetimes. Now they will match.</li> <li>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 &gt; 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.</li> </ul>

07/08/2021	7859	General	<b>Added</b> <ul style="list-style-type: none"> <li>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.</li> </ul>
06/14/2021	7836	Facebook	<b>Changed</b> <ul style="list-style-type: none"> <li>The default API version is updated to 11.0.</li> <li>Facebook has removed access from apps to return comment ids when the user does not also have permissions to moderate the comments. This may mean in some cases comments will stop returning with our driver when they were working previously.</li> <li>The user_likes and user_posts permissions (no longer used by default with the embedded driver) now have much more difficult requirements to access them from Facebook.</li> <li>Nearly all unique page type columns for the Pages table were deprecated by Facebook. For example, Parking columns, GeneralInfo, Attire, Restaurant Services columns, Awards, etc. These will all start returning null as Facebook transitions to the New Page Experience. These columns will be removed in the 2022 release of the driver.</li> </ul>
04/23/2021	7785	General	<b>Added</b> <ul style="list-style-type: none"> <li>Added support for handling client side formulas during insert / update. For example: UPDATE Table SET Col1 = Concat(Col1, " - ", Col2) WHERE Col2 LIKE</li> </ul>

'A%'			
04/23/2021	7783	General	<b>Changed</b> <ul style="list-style-type: none"> <li>Updated how display sizes are determined for varchar primary key and foreign key columns so they will match the reported length of the column.</li> </ul>
04/16/2021	7776	General	<b>Added</b> <ul style="list-style-type: none"> <li>Non-conditional updates between two columns is now available to all drivers. For example: UPDATE Table SET Col1=Col2</li> </ul> <b>Changed</b> <ul style="list-style-type: none"> <li>Reduced the length to 255 for varchar primary key and foreign key columns.</li> <li>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.</li> <li>Updated index naming convention to avoid duplicates</li> <li>Updated and standardized Getting Started connection help.</li> <li>Added the Advanced Features section to the help of all drivers.</li> <li>Categorized connection property listings in the help for all editions.</li> </ul>
04/15 /2021	7775	General	<b>Changed</b> <ul style="list-style-type: none"> <li>Kerberos authentication is updated to use TCP by default, but will fall back to UDP if a TCP connection cannot be established</li> </ul>

04/13/2021

7749

Facebook

**Changed**

- Updated the default Version to 10.0.

**Deprecated**

- The Permissions connection property is deprecated and replaced with Scope to match other drivers.

## Advanced Features

This section details a selection of advanced features of the Facebook 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.

### Query Processing

The adapter offloads as much of the SELECT statement processing as possible to Facebook and then processes the rest of the query in memory (client-side).

See [Query Processing](#) for more information.

## 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 Facebook 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 Posts WHERE MyColumn = 'value'"
  }
}
```

```

    },
    "MyView2": {
        "query": "SELECT * FROM MyTable WHERE Id IN (1,2,3)"
    }
}

```

Use the `UserDefinedViews` connection 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.

## Query Processing

### Query Processing

CData has a client-side SQL engine built into the adapter library. This enables support for the full capabilities that SQL-92 offers, including filters, aggregations, functions, etc.

For sources that do not support SQL-92, the adapter offloads as much of SQL statement processing as possible to Facebook and then processes the rest of the query in memory (client-side). This results in optimal performance.

For data sources with limited query capabilities, the adapter handles transformations of the SQL query to make it simpler for the adapter. The goal is to make smart decisions based on the query capabilities of the data source to push down as much of the computation as possible. The Facebook Query Evaluation component examines SQL queries and returns information indicating what parts of the query the adapter is not capable of executing natively.

The Facebook Query Slicer component is used in more specific cases to separate a single query into multiple independent queries. The client-side Query Engine makes decisions about simplifying queries, breaking queries into multiple queries, and pushing down or computing aggregations on the client-side while minimizing the size of the result set.

There's a significant trade-off in evaluating queries, even partially, client-side. There are always queries that are impossible to execute efficiently in this model, and some can be particularly expensive to compute in this manner. CData always pushes down as much of the query as is feasible for the data source to generate the most efficient query possible and provide the most flexible query capabilities.

## More Information

For a full discussion of how CData handles query processing, see [CData Architecture: Query Execution](#).

# 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 an explanation 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 <a href="#">Verbosity</a> 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 <a href="#">Verbosity</a> to 2 will log everything included in <a href="#">Verbosity</a> 1 and additional information about the request.
3	Setting <a href="#">Verbosity</a> to 3 will additionally log HTTP headers, as well as the body of the request and the response.
4	Setting <a href="#">Verbosity</a> to 4 will additionally log transport-level communication with the data source. This includes SSL negotiation.
5	Setting <a href="#">Verbosity</a> 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 verbatimities, which can delay execution times.

To refine the logged content further by showing/hiding specific categories of information, see [LogModules](#).

## Sensitive Data

Verbosity levels 3 and higher may capture information that you do not want shared

outside of your organization. The following lists information of concern for each level:

- Verbosity 3: The full body of the request and the response, which includes all the data returned by the adapter
- Verbosity 4: SSL certificates
- Verbosity 5: Any extra transfer data not included at Verbosity 3, such as non human-readable binary transfer data

### Best Practices for Data Security

Although we mask sensitive values, such as passwords, in the connection string and any request in the log, it is always best practice to review the logs for any sensitive information before sharing outside your organization.

## 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 Facebook Adapter supports several operations on data, including querying, deleting, modifying, and inserting.

### SELECT Statements

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

See [Data Model](#) for information on the capabilities of the Facebook 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').

## 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 Facebook 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> ]
  [
    LIMIT <expression>
  ]
} | SCOPE_IDENTITY()
<expression> ::=
  | <column_reference>
  | @ <parameter>
  | ?
  | COUNT( * | { [ DISTINCT ] <expression> } )
  | { AVG | MAX | MIN | SUM | COUNT } ( <expression> )
  | NULLIF ( <expression> , <expression> )
  | COALESCE ( <expression> , ... )
  | CASE <expression>
    WHEN { <expression> | <search_condition> } THEN { <expression> |
NULL } [ ... ]
  [ ELSE { <expression> | NULL } ]
  END
  | <literal>
  | <sql_function>
<search_condition> ::=
  {
    <expression> { = | != | > | < | >= | <= | AND | LIKE | NOT LIKE |

```



```
IN | NOT IN } [ <expression> ]
    } [ { AND | OR } ... ]
```

## Examples

1. Return all columns:

```
SELECT * FROM Posts
```

2. Rename a column:

```
SELECT "FromName" AS MY_FromName FROM Posts
```

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

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

4. Search data:

```
SELECT * FROM Posts WHERE Target = '11111'
```

5. The Facebook APIs support the following operators in the WHERE clause: =, !=, >, <, >=, <=, AND, LIKE, NOT LIKE, IN, NOT IN.

```
SELECT * FROM Posts WHERE Target = '11111';
```

## 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, FromName INTO
'csv://c:/Posts.txt' FROM 'Posts' WHERE Target = '11111'");
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 'Posts' IN
'csv://filename=c:/Posts.csv;delimiter=tab' FROM 'Posts' WHERE Target =
'11111'");
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 Posts (FromName) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement
(cmd,Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, "User 2");
```

```
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 Posts SET FromName='User 2' WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "11111_22222");
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:facebook:InitiateOAuth=GETANDREFRESH;",);
String cmd = "DELETE FROM Posts WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "11111_22222");
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;
```

## PIVOT and UNPIVOT

**PIVOT** and **UNPIVOT** can be used to change a table-valued expression into another table.

### PIVOT

PIVOT rotates a table-value expression by turning unique values from one column into multiple columns in the output. PIVOT can run aggregations where required on any column value.

### PIVOT Syntax

```
"SELECT 'AverageCost' AS Cost_Sorted_By_Production_Days, [0], [1], [2],  
[3], [4]  
FROM  
(  
  SELECT DaysToManufacture, StandardCost  
  FROM Production.Product  
) AS SourceTable  
PIVOT  
(  
  AVG(StandardCost)  
  FOR DaysToManufacture IN ([0], [1], [2], [3], [4])  
) AS PivotTable;"
```

### UNPIVOT

UNPIVOT carries out nearly the opposite to PIVOT by rotating columns of a table-valued expressions into column values.

## UNPIVOT Sytax

```
"SELECT VendorID, Employee, Orders
FROM
(SELECT VendorID, Emp1, Emp2, Emp3, Emp4, Emp5
FROM pvt) p
UNPIVOT
(Orders FOR Employee IN
(Emp1, Emp2, Emp3, Emp4, Emp5)
)AS unpvt;"
```

For further information on PIVOT and UNPIVOT, see [FROM clause plus JOIN, APPLY, PIVOT \(Transact-SQL\)](#)

## Data Model

The Facebook Adapter models Facebook APIs as relational Tables, Views, and Stored Procedures. API limitations and requirements are documented in this section; you can use the [SupportEnhancedSQL](#) feature, set by default, to circumvent most of these limitations.

### Tables

[Tables](#) describes the available tables.

### Views

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

### Stored Procedures

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

## Tables

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

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

## Facebook Adapter Tables

Name	Description
<a href="#">Likes</a>	Create, delete, and query the Likes for a Target. Alternatively, lists Pages that the specified User or Page Likes. Authentication is required to use this table.
<a href="#">Posts</a>	Create, delete, and query the Posts for a Target based on either the Target or Id. Posts can also be inserted based on a Target, or deleted based on Id. This table requires authentication.

## Likes

Create, delete, and query the Likes for a Target. Alternatively, lists Pages that the specified User or Page Likes. Authentication is required to use this table.

## Table Specific Information

Likes in Facebook represent the users that like a particular target. The target may be a post, page, picture, or other type of valid entity that may have likes associated with it.

## Select

When selecting likes, specify a target. The target represents the username or Id of the entity that likes are being retrieved for. For example:

```
SELECT Id, Name, Picture FROM Likes WHERE Target = '15526475270_410830705612736'
```

If a user or page is specified as the target for likes, then the pages that the user or page likes will be returned. For example:

```
SELECT Id, Name, Username, Category FROM Likes WHERE Target = 'facebook'
```

If no target is specified, the currently authenticated user will be used as the target. In this case, the pages that the authenticated user likes will be returned.

## Insert

To insert a like or to like something, simply issue an INSERT statement and specify the target you are liking. The target must be an album, checkin, comment, photo, post, status update, or other object that can be liked. For example:

```
INSERT INTO Likes (Target) VALUES ('123456789_123456789')
```

## Update

Facebook does not allow likes to be updated.

## Delete

Facebook does not allow likes to be deleted.

## Columns

Name	Type	ReadOnly	Description
ID [KEY]	String	True	The Id of a user who likes the target, which may or may not be combined with the target Id. The user Id will be after the final '_ '.
Target [KEY]	String	False	The Id or username of the target being liked. This may be a post, page, picture, or other valid id with likes.



Name	<i>String</i>	True	The name of the user who likes the target. May alternatively be the name of a page.
Username	<i>String</i>	True	The username of a page when retrieving pages a page has liked.
Picture	<i>String</i>	True	Picture of the user who likes the target. May alternatively be the picture of a page.
Category	<i>String</i>	True	The category of the user or page.
CreatedTime	<i>Datetime</i>	True	The time the like was created, if available.

## 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. For more information, see the WHERE clause section.

Name	Type	Description
offset	<i>String</i>	Which result to begin returning results from. Used for manual paging of results.

## Posts

Create, delete, and query the Posts for a Target based on either the Target or Id. Posts can also be inserted based on a Target, or deleted based on Id. This table requires authentication.

## Table Specific Information

Posts in Facebook are posts to a user's profile feed. Posts can be made to a user, page, application, group, or event.

### Select

When selecting posts, a target may be specified. The target represents a page or another valid entity that may have posts for it. If no target is specified, the authenticated user will be used as the target. For example, to retrieve posts made by a page:

```
SELECT * FROM Posts WHERE Target = 'PageId'
```

If you know the post Id, you can specify the Id to obtain information about the specific post. For example:

```
SELECT * FROM Posts WHERE Id = 'PostId'
```

When querying posts, elements may be retrieved by specifying either the CreatedTime or the UpdatedTime. For example:

```
SELECT * FROM Posts WHERE Target='facebook' AND CreatedTime >=
'1/1/2012' AND CreatedTime <= '2/1/2012'
```

### Insert

To insert a post, you will need to specify the Target and Message of the post. For instance:

```
INSERT INTO Posts (Message, Target) VALUES ('My New Post', 'PageId')
```

Create a photo post by using the Link or ObjectId columns. With the Link column, specify the accessible URL of an image to add the post to. You can also create video posts:

```
INSERT INTO Posts (Message, Target, Link) VALUES ('My New Post',
'PageId' , 'http://imagerepo.net/testimage.png')
```

```
INSERT INTO Posts (Message, Target, Link) VALUES ('My New Post',
'PageId' , 'http://samplevideos.net/newmovie.mp4')
```

You can also set local files in the Link column to upload media. Note that you must set UploadLinkedMedia to true for this to work:

```
INSERT INTO Posts (Message, Target, Link) VALUES ('From Local', 'PageId'
, 'file:///D://test/sample.mp4')
```

With the ObjectId column, specify the ID of an unpublished photo in your account. To upload a photo without publishing it, use the UploadPhoto stored procedure. Video Ids are not supported:

```
INSERT INTO Posts (Message, Target, ObjectId) VALUES ('My New Post',
'PageId' , '43572')
```

## Update

Facebook does not allow posts to be updated.

## Delete

Posts can be deleted by issuing a DELETE statement and specifying the Id of the post. Please note that Facebook allows only posts created by your app to be deleted using your app.

## Columns

Name	Type	ReadOnly	Description
ID [KEY]	String	True	The Id of the post.
Target	String	False	The Id or username of the target

			you are retrieving posts for or are posting to. This can be an event, group, page, or user.
FromId	<i>String</i>	True	Id of the user who made the post.
FromName	<i>String</i>	True	Name of the user who made the post.
FromPicture	<i>String</i>	True	Picture of the user who made the post.
FromCategory	<i>String</i>	True	Category of the user who made the post. FromCategory can only be retrieved if the other From* fields are not selected.
ToData	<i>String</i>	True	An aggregate of users the post was made to.
Message	<i>String</i>	False	The message of the post. A message is required when inserting a post.
MessageTags	<i>String</i>	True	An aggregate of objects tagged in the message such as users, pages, etc.
Attachments	<i>String</i>	True	An aggregate for the attachments of the post.
Picture	<i>String</i>	False	A link to the picture included in the post.
FullPicture	<i>String</i>	False	A link to the original picture included in the post. The one listed in Picture may be a resized smaller version.

Icon	<i>String</i>	True	Link to an icon representing the type of post.
Actions	<i>String</i>	True	An aggregate of available actions on the post such as commenting or liking.
CommentsCount	<i>Integer</i>	True	The number of comments for the post.
LikesCount	<i>Integer</i>	True	The number of times the post has been liked.
SharesCount	<i>Integer</i>	True	The number of times the post has been shared.
PlaceId	<i>String</i>	False	The Id of the location associated with the post, if any.
PlaceName	<i>String</i>	True	The name of the location associated with the post, if any.
ApplicationId	<i>String</i>	True	Id of the application this post came from.
ApplicationName	<i>String</i>	True	Name of the application this post came from.
ApplicationCanvasName	<i>String</i>	True	Information about the application used to create the entity.
ApplicationNamespace	<i>String</i>	True	Information about the application used to create the entity.
Story	<i>String</i>	True	Text of stories not intentionally generated by users, such as those generated when two users become friends; you must have the Include Recent Activity Stories migration enabled in your app to

			retrieve these stories. Requires the read_stream permission.
StoryTags	<i>String</i>	True	An aggregate of objects (users, pages, etc.) associated with the story.
LikesData	<i>String</i>	True	An aggregate of like data.
CommentsData	<i>String</i>	True	An aggregate of comments for this post.
CreatedTime	<i>Datetime</i>	True	When the post was created.
UpdatedTime	<i>Datetime</i>	True	When the post was last updated.

## 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. For more information, see the WHERE clause section.

<b>Name</b>	<b>Type</b>	<b>Description</b>
TaggedUser	<i>String</i>	If set to an Id or username, it will retrieve posts where the specified user has been tagged. Requires the read_stream permission.
LocationUser	<i>String</i>	If set to an Id or username, it will retrieve location posts where the user has been tagged. Requires the user_photos and user_status permissions.

## 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.

Dynamic views, such as queries exposed as views, and views for looking up specific combinations of project\_team work items are supported.

## Facebook Adapter Views

Name	Description
<a href="#">Comments</a>	Create, update, delete, and query the Comments for a Target. Comments may also be inserted based on a Target or deleted based on Id.
<a href="#">Events</a>	Query the Events for a Target such as a page id.
<a href="#">Groups</a>	Query the Pages based on the supplied Id.
<a href="#">InsightsByConsumptionType</a>	Allows retrieval of insights by consumption type.
<a href="#">InsightsByFeedbackType</a>	Allows retrieval of insights by feedback type.
<a href="#">InsightsByLikeSourceType</a>	Allows retrieval of insights by like source type.
<a href="#">InsightsByReactionTotals</a>	Allows retrieval of insights by like source type.
<a href="#">InsightsByStoryType</a>	Allows retrieval of insights by like story type.
<a href="#">InsightsByTabType</a>	Allows retrieval of insights by tab type..
<a href="#">InsightsByUnLikeSourceType</a>	Allows retrieval of insights by like source type.
<a href="#">InstagramAccountInsights</a>	Allows you to get insights for an Instagram Business Account. Requires the <code>instagram_basic</code> and <code>instagram_manage_insights</code> scopes.

<a href="#">Pages</a>	Query the Pages based on the supplied Id.
<a href="#">Permissions</a>	Query the Permissions the User has granted the current application.
<a href="#">Photos</a>	Query Photos associated with a Target. Accessing Photo information typically requires the user_photos permission.
<a href="#">Places</a>	Query the Places based on the supplied Id. Places are stored as Pages in Facebook.
<a href="#">Ratings</a>	List of ratings for a Facebook Page. This view requires authentication.
<a href="#">SimpleInsights</a>	Allows the retrieval of simple insights with a single value in the response.
<a href="#">SimpleVideoInsights</a>	Allows the retrieval of simple video insights with a single value in the response.
<a href="#">StoryAttachments</a>	Query attachments from a post or comment.
<a href="#">TaggedBy</a>	Query information about Posts, Statuses, Photos, and other entities that have tagged the User or Page. This view is a derivative of the Wall connection where only entries that have tagged the Target User or Page will be returned. In general it is only available for Pages.
<a href="#">Users</a>	Returns basic information about the authenticated user.
<a href="#">VideoInsightsByActionType</a>	Allows the retrieval of video insights by story action type.
<a href="#">VideoInsightsByDistributionType</a>	Allows the retrieval of video insights by distribution type.
<a href="#">VideoInsightsByReactionType</a>	Allows the retrieval of video insights by reaction type.
<a href="#">Videos</a>	Query Videos from a Target. Normally requires the user_



---

videos permission.

---

Wall

---

Query Posts from the Wall of a Target.

---

## Comments

Create, update, delete, and query the Comments for a Target. Comments may also be inserted based on a Target or deleted based on Id.

### Table Specific Information

Comments in Facebook are comments about a specific thing. They are always associated with a target, which is the item the comment is directed toward. For example, this could be a post, a picture, or a video. Using this table, you can list the comments for a specific target and also insert new comments about a target.

### Select

When querying comments, either the Target or the Id of the comment must be specified. For example, to retrieve all the comments about a specific post, your SELECT statement could look something like this:

```
SELECT * FROM Comments WHERE Target = '15526475270_410830705612736'
```

Alternatively, you can specify the Id to retrieve a specific comment. For example:

```
SELECT * FROM Comments WHERE Id = '15526475270_410830705612736_5193593'
```

### Columns

Name	Type	Description
ID [KEY]	String	The Id of the comment.

---

Target	<i>String</i>	The Id of the target you are retrieving comments for. This may be an album, checkin, link, note, photo, post, status update, or video.
FromId	<i>String</i>	Id of the user who made the comment.
FromName	<i>String</i>	Name of the user who made the comment.
FromPicture	<i>String</i>	Picture of the user who made the comment.
Message	<i>String</i>	The text of the comment.
MessageTags	<i>String</i>	Aggregate of tags contained in the message.
Likes	<i>Integer</i>	The number of likes the comment has.
CommentsCount	<i>Integer</i>	The number of comments in reply to this comment.
LikesData	<i>String</i>	Aggregate of likes information on the comment.
UserLikes	<i>Boolean</i>	Boolean indicating if the authenticated user likes the comment.
CanRemove	<i>Boolean</i>	Boolean indicating if the comment can be removed.
Attachments	<i>String</i>	An aggregate for the attachments of the post.
CreatedTime	<i>Datetime</i>	The time the comment was created.

## 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. For more information, see the WHERE clause section.

Name	Type	Description
offset	<i>String</i>	Which result to begin returning results from. Used for manual paging of results.

## Events

Query the Events for a Target such as a page id.

### Table Specific Information

Events in Facebook are events created by a user or page that mark a specific date when something will take place. Events may be queried by Target or Id.

### Select

When selecting events, a target may be specified. The target represents a page that has created events. By default, this will be the currently authenticated user or page. For example, to retrieve events associated with a page:

```
SELECT * FROM Events WHERE Target = 'PageId'
```

If you know the event Id, you may also just specify the Id to obtain information about the specific event. For example:

```
SELECT * FROM Events WHERE Id = 'EventId'
```

## Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the event.
Target	<i>String</i>	The Id or username of the target you are retrieving events for or are posting events to. This may be a page or a user.
Name	<i>String</i>	The name of the event.
StartTime	<i>Datetime</i>	The start time of the event.
EndTime	<i>Datetime</i>	The end time of the event.
Timezone	<i>String</i>	The time zone the event will take place in.
Description	<i>String</i>	The description of the event.
Picture	<i>String</i>	A URL to the picture of the event.
OwnerId	<i>String</i>	The Id of the user that created the event.
OwnerName	<i>String</i>	The name of the user that created the event.
OwnerPicture	<i>String</i>	Picture of the user who created the event.

OwnerCategory	<i>String</i>	Category of the owner, if available.
Location	<i>String</i>	The location of the event, if specified.
LocationId	<i>String</i>	The Id of the location for the event.
LocationStreet	<i>String</i>	The street address of the event.
LocationCity	<i>String</i>	The the city for the event.
LocationState	<i>String</i>	The the state for the event.
LocationCountry	<i>String</i>	The the country for the event.
LocationZip	<i>String</i>	The the ZIP code of the event.
LocationLatitude	<i>String</i>	The latitude of the event.
LocationLongitude	<i>String</i>	The longitude of the event.
TicketUri	<i>String</i>	A URL to a location to buy tickets for this event.
UpdatedTime	<i>Datetime</i>	When the event was last updated.
AttendingCount	<i>Integer</i>	Number of people attending the event.
DeclinedCount	<i>Integer</i>	Number of people who declined the event.

InterestedCount	<i>Integer</i>	Number of people interested in the event.
MaybeCount	<i>Integer</i>	Number of people who maybe going to the event.
NoreplyCount	<i>Integer</i>	Number of people who did not reply to the event.

## Groups

Query the Pages based on the supplied Id.

### Table Specific Information

Groups are a place to communicate about shared interests with certain people. Groups may be queried by Id.

### Select

If you know the groups Id, you may also just specify the Id to obtain information about the specific group. For example:

```
SELECT * FROM Groups WHERE Id = 'GroupId'
```

### Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Group ID.

Name	<i>String</i>	The name of the Group.
Description	<i>String</i>	A brief description of the Group.
Email	<i>String</i>	The email address to upload content to the Group. Only current members of the Group can use this.
Icon	<i>String</i>	The URL for the Group's icon.
MemberCount	<i>Integer</i>	The number of members in the Group.
MemberRequestCount	<i>Integer</i>	The number of members in the Group.
CoverId	<i>String</i>	Id of the cover image for the group.
CoverSource	<i>String</i>	The URL to the cover image for the group.
CoverOffsetY	<i>String</i>	The y-axis offset of the cover image for the group.
CoverOffsetX	<i>String</i>	The x-axis offset of the cover image for the group.
Permissions	<i>String</i>	The permissions a User has granted for an app installed in the Group.
Privacy	<i>String</i>	The privacy setting of the Group. Possible values are CLOSED, OPEN, and SECRET. Requires an access token of an Admin of the Group.
UpdateTime	<i>Datetime</i>	The last time the Group was updated (includes changes Group properties, Posts, and Comments). Requires an access token of an Admin of the Group.

# InsightsByConsumptionType

Allows retrieval of insights by consumption type.

## Table Specific Information

Insights by consumption type refers to any insights that can be retrieved from a page or post that measure how users consumed your content.

## Select

When selecting insights, a Target must be specified. In addition, an InsightName and Period should always be specified. For instance:

```
SELECT * FROM InsightsByConsumptionType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_CONSUMPTIONS_BY_CONSUMPTION_TYPE' AND Period = 'day'
```

Additionally, StartTime and EndTime can be used to specify a range where insight data should be drawn from:

```
SELECT * FROM InsightsByConsumptionType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_CONSUMPTIONS_BY_CONSUMPTION_TYPE' AND Period = 'day'
AND EndTime >= '12/1/2015' AND EndTime <= '12/31/2015'
```

## Columns

Name	Type	Description
RowNumber [KEY]	Int	The row number of the result.
EndTime	Datetime	The most recent date this insight data is relevant for.
VideoPlay	Long	Insight column indication the total for the stated



		consumption type.
OtherClicks	<i>Long</i>	Insight column indication the total for the stated consumption type.
PhotoView	<i>Long</i>	Insight column indication the total for the stated consumption type.
LinkClicks	<i>Long</i>	Insight column indication the total for the stated consumption type.
ButtonClicks	<i>Long</i>	Insight column indication the total for the stated consumption type.
Target	<i>String</i>	The target of the insight. This must always be specified.
InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: PAGE_CONSUMPTIONS_BY_CONSUMPTION_TYPE,PAGE_CONSUMPTIONS_BY_CONSUMPTION_TYPE_UNIQUE,POST_CLICKS_BY_TYPE,POST_CLICKS_BY_TYPE_UNIQUE
Period	<i>String</i>	The period for the insight. This must always be specified. The allowed values are <i>day</i> , <i>week</i> , <i>days_28</i> , <i>lifetime</i> .

## 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. For more information, see the WHERE clause section.

<b>Name</b>	<b>Type</b>	<b>Description</b>
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.

---

ToDateTime	<i>String</i>	The latest time for insight data to have been collected.
------------	---------------	--

---

## InsightsByFeedbackType

Allows retrieval of insights by feedback type.

### Table Specific Information

Insights by feedback type refers to any insights that can be retrieved from a page or post that measure what kind of feedback was left.

### Select

When selecting insights, a Target must be specified. In addition, an InsightName and Period should always be specified. For instance:

```
SELECT * FROM InsightsByFeedbackType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_POSITIVE_FEEDBACK_BY_TYPE' AND Period = 'day'
```

Additionally, StartTime and EndTime can be used to specify a range where insight data should be drawn from:

```
SELECT * FROM InsightsByFeedbackType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_POSITIVE_FEEDBACK_BY_TYPE' AND Period = 'day' AND
EndTime >= '12/1/2015' AND EndTime <= '12/31/2015'
```

### Columns

---

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.

---

EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
Like	<i>Long</i>	Insight column indication the total for the positive feedback type.
Comment	<i>Long</i>	Insight column indication the total for the positive feedback type.
Link	<i>Long</i>	Insight column indication the total for the positive feedback type.
Answer	<i>Long</i>	Insight column indication the total for the positive feedback type.
Claim	<i>Long</i>	Insight column indication the total for the positive feedback type.
Rsvp	<i>Long</i>	Insight column indication the total for the positive feedback type.
HideAllClicks	<i>Long</i>	Insight column indication the total for the negative feedback type.
HideClicks	<i>Long</i>	Insight column indication the total for the negative feedback type.
UnlikePageClicks	<i>Long</i>	Insight column indication the total for the negative feedback type.
ReportSpamClicks	<i>Long</i>	Insight column indication the total for the negative feedback type.
XButtonClicks	<i>Long</i>	Insight column indication the total for the negative feedback type.
Target	<i>String</i>	The target of the insight. This must always be specified.
InsightName	<i>String</i>	The name of the insight. This must always be specified. The

		available values are: PAGE_NEGATIVE_FEEDBACK_BY_TYPE,PAGE_NEGATIVE_FEEDBACK_BY_TYPE_UNIQUE,PAGE_POSITIVE_FEEDBACK_BY_TYPE,PAGE_POSITIVE_FEEDBACK_BY_TYPE_UNIQUE,POST_NEGATIVE_FEEDBACK_BY_TYPE,POST_NEGATIVE_FEEDBACK_BY_TYPE_UNIQUE
Period	<i>String</i>	The period for the insight. This must always be specified.  The allowed values are <i>day</i> , <i>week</i> , <i>days_28</i> , <i>lifetime</i> .

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## InsightsByLikeSourceType

Allows retrieval of insights by like source type.

## Columns

Name	Type	Description
RowNumber	<i>Int</i>	The row number of the result.

[KEY]		
EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
Ads	<i>Long</i>	Insight column indication the total for the like source type.
NewsFeed	<i>Long</i>	Insight column indication the total for the like source type.
Other	<i>Long</i>	Insight column indication the total for the like source type.
PageSuggestions	<i>Long</i>	Insight column indication the total for the like source type.
RestoredLikesFromReactivatedAccounts	<i>Long</i>	Insight column indication the total for the like source type.
Search	<i>Long</i>	Insight column indication the total for the like source type.
YourPage	<i>Long</i>	Insight column indication the total for the like source type.
Target	<i>String</i>	The target of the insight. This must always be specified.
InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: PAGE_FANS_BY_LIKE_SOURCE,PAGE_FANS_BY_LIKE_SOURCE_UNIQUE

## 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. For more

information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## InsightsByReactionTotals

Allows retrieval of insights by like source type.

### Table Specific Information

Insights by reaction total type refers to the reaction totals for your page or post.

### Select

When selecting insights, a Target must be specified. This should be set to a page or post depending on the InsightName specified. You can set InsightName to 'PAGE\_ACTIONS\_POST\_REACTIONS\_TOTAL' or 'POST\_REACTIONS\_BY\_TYPE\_TOTAL', by default it will be 'PAGE\_ACTIONS\_POST\_REACTIONS\_TOTAL'. For instance:

```
SELECT * FROM InsightsByReactionTotals WHERE Target = 'mypageid' AND
InsightName='PAGE_ACTIONS_POST_REACTIONS_TOTAL'
```

Additionally, StartTime and EndTime can be used to specify a range where insight data should be drawn from:

```
SELECT * FROM InsightsByReactionTotals WHERE Target = 'mypageid' AND
InsightName='PAGE_ACTIONS_POST_REACTIONS_TOTAL' AND EndTime >=
'12/1/2015' AND EndTime <= '12/31/2015'
```

## Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
Like	<i>Long</i>	Total like reactions.
Love	<i>Long</i>	Total love reactions.
Wow	<i>Long</i>	Total wow reactions.
Haha	<i>Long</i>	Total haha reactions.
Sorry	<i>Long</i>	Total sorry reactions.
Anger	<i>Long</i>	Total anger reactions.
Target	<i>String</i>	The target of the insight. This must always be specified.

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	String	The earliest time for insight data to have been collected.
ToDateTime	String	The latest time for insight data to have been collected.
InsightName	String	Insight Name. The available values are: PAGE_ACTIONS_POST_REACTIONS_TOTAL, POST_REACTIONS_BY_TYPE_TOTAL  The default value is PAGE_ACTIONS_POST_REACTIONS_TOTAL.

## InsightsByStoryType

Allows retrieval of insights by like story type.

### Table Specific Information

Insights by story type refers to any insights that can be retrieved from a page or post that measure the types of stories that have occurred.

### Select

When selecting insights, a Target must be specified. In addition, an InsightName and Period should always be specified. For instance:

```
SELECT * FROM InsightsByStoryType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_STORIES_BY_STORY_TYPE' AND Period = 'day'
```

Additionally, StartTime and EndTime can be used to specify a range where insight data should be drawn from:

```
SELECT * FROM InsightsByStoryType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_STORIES_BY_STORY_TYPE' AND Period = 'day' AND
EndTime >= '12/1/2015' AND EndTime <= '12/31/2015'
```



## Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
UserPost	<i>Long</i>	Insight column indication the total for the stated story type.
PagePost	<i>Long</i>	Insight column indication the total for the stated story type.
Checkin	<i>Long</i>	Insight column indication the total for the stated story type.
Fan	<i>Long</i>	Insight column indication the total for the stated story type.
Question	<i>Long</i>	Insight column indication the total for the stated story type.
Coupon	<i>Long</i>	Insight column indication the total for the stated story type.
Event	<i>Long</i>	Insight column indication the total for the stated story type.
Mention	<i>Long</i>	Insight column indication the total for the stated story type.
Other	<i>Long</i>	Insight column indication the total for the stated story type.
Target	<i>String</i>	The target of the insight. This must always be specified.

InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: PAGE_CONTENT_ACTIVITY_BY_ACTION_TYPE,PAGE_IMPRESSIONS_BY_STORY_TYPE,PAGE_IMPRESSIONS_BY_STORY_TYPE_UNIQUE,POST_IMPRESSIONS_BY_STORY_TYPE,POST_IMPRESSIONS_BY_STORY_TYPE_UNIQUE
Period	<i>String</i>	The period for the insight. This must always be specified. The allowed values are <i>day</i> , <i>week</i> , <i>days_28</i> , <i>lifetime</i> .

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## InsightsByTabType

Allows retrieval of insights by tab type..

### Table Specific Information

Insights by tab type refers to any insights that can be retrieved from a page or post that measure what tab was used to view your content.

## Select

When selecting insights, a Target must be specified. In addition, an InsightName and Period should always be specified. For instance:

```
SELECT * FROM InsightsByTabType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_TAB_VIEWS_LOGIN_TOP_UNIQUE' AND Period = 'day'
```

Additionally, StartTime and EndTime can be used to specify a range where insight data should be drawn from:

```
SELECT * FROM InsightsByTabType WHERE Target = 'mypostid' AND
InsightName = 'PAGE_TAB_VIEWS_LOGIN_TOP_UNIQUE' AND Period = 'day' AND
EndTime >= '12/1/2015' AND EndTime <= '12/31/2015'
```

## Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
About	<i>Long</i>	Insight column indication the total for the tab type.
Ads	<i>Long</i>	Insight column indication the total for the tab type.
Album	<i>Long</i>	Insight column indication the total for the tab type.
AllActivity	<i>Long</i>	Insight column indication the total for the tab type.

App	<i>Long</i>	Insight column indication the total for the tab type.
Community	<i>Long</i>	Insight column indication the total for the tab type.
Events	<i>Long</i>	Insight column indication the total for the tab type.
Home	<i>Long</i>	Insight column indication the total for the tab type.
Info	<i>Long</i>	Insight column indication the total for the tab type.
Insights	<i>Long</i>	Insight column indication the total for the tab type.
Jobs	<i>Long</i>	Insight column indication the total for the tab type.
Likes	<i>Long</i>	Insight column indication the total for the tab type.
Locations	<i>Long</i>	Insight column indication the total for the tab type.
Notes	<i>Long</i>	Insight column indication the total for the tab type.
Photos	<i>Long</i>	Insight column indication the total for the tab type.
PhotosAlbums	<i>Long</i>	Insight column indication the total for the tab type.
PhotosStream	<i>Long</i>	Insight column indication the total for the tab type.
Posts	<i>Long</i>	Insight column indication the total for the tab type.

Profile	<i>Long</i>	Insight column indication the total for the tab type.
ProfileAbout	<i>Long</i>	Insight column indication the total for the tab type.
ProfileCommunity	<i>Long</i>	Insight column indication the total for the tab type.
ProfileHome	<i>Long</i>	Insight column indication the total for the tab type.
ProfileInfo	<i>Long</i>	Insight column indication the total for the tab type.
ProfileLikes	<i>Long</i>	Insight column indication the total for the tab type.
ProfilePhotos	<i>Long</i>	Insight column indication the total for the tab type.
ProfilePosts	<i>Long</i>	Insight column indication the total for the tab type.
Reviews	<i>Long</i>	Insight column indication the total for the tab type.
Timeline	<i>Long</i>	Insight column indication the total for the tab type.
Videos	<i>Long</i>	Insight column indication the total for the tab type.
Wall	<i>Long</i>	Insight column indication the total for the tab type.
Target	<i>String</i>	The target of the insight. This must always be specified.

InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: PAGE_TAB_VIEWS_LOGIN_TOP_UNIQUE,PAGE_TAB_VIEWS_LOGIN_TOP,PAGE_TAB_VIEWS_LOGOUT_TOP
Period	<i>String</i>	The period for the insight. This must always be specified. The allowed values are <i>day</i> , <i>week</i> .

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## InsightsByUnLikeSourceType

Allows retrieval of insights by like source type.

## Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.

EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
DeactivatedOrMemorializedAccountRemovals	<i>Long</i>	Insight column indication the total for the like source type.
SuspiciousAccountRemovals	<i>Long</i>	Insight column indication the total for the like source type.
UnlikesFromPagePostsOrNewsFeed	<i>Long</i>	Insight column indication the total for the like source type.
UnlikesFromSearch	<i>Long</i>	Insight column indication the total for the like source type.
UnderageAccountRemovals	<i>Long</i>	Insight column indication the total for the like source type.
UnlikesFromPageSuggestions	<i>Long</i>	Insight column indication the total for the like source type.
Other	<i>Long</i>	Insight column indication the total for the like source type.
Target	<i>String</i>	The target of the insight. This must always be specified.
InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: PAGE_FANS_BY_UNLIKE_SOURCE, PAGE_FANS_BY_UNLIKE_SOURCE_UNIQUE

## 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. For more

information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## InstagramAccountInsights

Allows you to get insights for an Instagram Business Account. Requires the `instagram_basic` and `instagram_manage_insights` scopes.

### Columns

Name	Type	Description
RowNumber [KEY]	<i>String</i>	The row number of the result.
EndTime	<i>Date</i>	The most recent date this insight data is relevant for.
Value	<i>Long</i>	Insight column for a singular value response.
Description	<i>String</i>	<p>The period for the insight. This controls how rows will be broken up (by day, week, 28 days, or lifetime). This must always be specified.</p> <p>The allowed values are <i>day</i>, <i>week</i>, <i>days_28</i>, <i>lifetime</i>.</p>
InsightName	<i>String</i>	The name of the insight. This must always be specified. The



available values are: IMPRESSIONS,REACH,FOLLOWER_COUNT,EMAIL_CONTACTS,PHONE_CALL_CLICKS,TEXT_MESSAGE_CLICKS,GET_DIRECTIONS_CLICKS,WEBSITE_CLICKS,PROFILE_VIEWS,AUDIENCE_GENDER_AGE,AUDIENCE_LOCALE,AUDIENCE_COUNTRY,AUDIENCE_CITY,ONLINE_FOLLOWERS		
Period	<i>String</i>	<p>The period for the insight. This controls how rows will be broken up (by day, week, 28 days, or lifetime). This must always be specified.</p> <p>The allowed values are <i>day</i>, <i>week</i>, <i>days_28</i>, <i>lifetime</i>.</p>
Target	<i>String</i>	<p>The target of the insight. This value correspond to the InstagramBusinessAccountId of the Pages view. You can also set it to 'Target IN (SELECT InstagramBusinessAccountId FROM Pages)'. This must always be specified.</p>

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## Pages

Query the Pages based on the supplied Id.

## Table Specific Information

Pages in Facebook are pages that are created by a user and may be maintained by one or multiple users. Pages, like users, may be referred to by Id or by screen name.

### Select

When querying pages, if nothing is specified then the pages you administrate will be displayed by default. Otherwise, the Id of the page can be specified. For example:

```
SELECT * FROM Pages WHERE Id = 'facebook'
```

### Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the page.
Username	<i>String</i>	The username for the page, if any.
Name	<i>String</i>	The name of the page.
Category	<i>String</i>	The category of the page.
Categories	<i>String</i>	An aggregate of categories associated with the page if multiple categories are available.
Link	<i>String</i>	A link to the page.
Picture	<i>String</i>	A link to the profile picture of the page for the page.

CoverId	<i>String</i>	Id of the cover image for the page.
CoverSource	<i>String</i>	The URL to the cover image for the page.
CoverOffsetY	<i>String</i>	The y-axis offset of the cover image for the page.
CoverOffsetX	<i>String</i>	The x-axis offset of the cover image for the page.
About	<i>String</i>	Basic information about the page.
Description	<i>String</i>	A description of the page, if available.
GeneralInfo	<i>String</i>	General information provided by the page.
LocationStreet	<i>String</i>	The street address of the page.
LocationCity	<i>String</i>	The city for the page.
LocationState	<i>String</i>	The state for the page.
LocationCountry	<i>String</i>	The country for the page.
LocationZip	<i>String</i>	The ZIP code of the page.
LocationLatitude	<i>String</i>	The latitude of the page.
LocationLongitude	<i>String</i>	The longitude of the page.

StoreNumber	<i>Integer</i>	Unique store number for this location page, if applicable.
Phone	<i>String</i>	The phone number of the page, if available.
Website	<i>String</i>	A link to the website for the page.
Likes	<i>Integer</i>	The number of people who like the page.
Checkins	<i>Integer</i>	The total number of users who have checked in to the place associated with the page.
TalkingAboutCount	<i>Integer</i>	The number of users talking about the page.
WereHereCount	<i>Integer</i>	The number of users who were at the location the page is for, if applicable.
CanPost	<i>Boolean</i>	Boolean indicating if the authenticated user can post of the page for the page.
IsPublished	<i>Boolean</i>	Boolean indicating if the page for the page has been published.
IsCommunityPage	<i>Boolean</i>	Boolean indicating if this is a community page.
PublicTransit	<i>String</i>	The public transit available for the page, if any.
ParkingStreet	<i>Boolean</i>	Boolean indicating if street parking is available.
ParkingLot	<i>Boolean</i>	Boolean indicating if a parking lot is available.

ParkingValet	<i>Boolean</i>	Boolean indicating if valet parking is available.
PromotionEligible	<i>Boolean</i>	Boosted posts eligibility status. Requires the manage_pages permission and you must be an administrator of the page.
PromotionIneligibleReason	<i>String</i>	Reason boosted posts are not eligible. Requires the manage_pages permission and you must be an administrator of the page.
Founded	<i>String</i>	When the company is founded. Applicable to companies.
Mission	<i>String</i>	The company mission. Applicable to companies.
Products	<i>String</i>	The products of this company. Applicable to companies.
Hours	<i>String</i>	An aggregate for the hours of operation. Applicable to businesses and places.
Attire	<i>String</i>	Dress code of the business. Applicable to restaurants or nightlife. Valid values are Casual, Dressy or Unspecified.
AcceptsCashOnly	<i>Boolean</i>	Whether the business accepts only cash as a payment option. Applicable to restaurants or nightlife.
AcceptsVisa	<i>Boolean</i>	Whether the business accepts Visa as a payment option. Applicable to restaurants or nightlife.
AcceptsAmericanExpress	<i>Boolean</i>	Whether the business accepts American Express as a payment option. Applicable to restaurants or nightlife.
AcceptsMasterCard	<i>Boolean</i>	Whether the business accepts MasterCard as a

		payment option. Applicable to restaurants or nightlife.
AcceptsDiscover	<i>Boolean</i>	Whether the business accepts Discover as a payment option. Applicable to restaurants or nightlife.
PriceRange	<i>String</i>	Price range of the business. Applicable to restaurants or nightlife. Valid values are \\$(0-10), \\$(10-30), \\$(30-50), \\$(50+), or Unspecified.
TakesReservations	<i>Boolean</i>	Whether the restaurant takes reservations. Only applicable to restaurants.
AllowsWalkins	<i>Boolean</i>	Whether the restaurant allows walk-ins. Only applicable to restaurants.
AllowsGroups	<i>Boolean</i>	Whether the restaurant accommodates groups. Only applicable to restaurants.
AllowsKids	<i>Boolean</i>	Whether the restaurant allows kids. Only applicable to restaurants.
ProvidesTakeout	<i>Boolean</i>	Whether the restaurant provides a takeout service. Only applicable to restaurants.
ProvidesDelivery	<i>Boolean</i>	Whether the restaurant provides a delivery service. Only applicable to restaurants.
ProvidesCatering	<i>Boolean</i>	Whether the restaurant provides a catering service. Only applicable to restaurants.
HasWaiters	<i>Boolean</i>	Whether the restaurant has waiters. Only applicable to restaurants.
HasOutdoorSeating	<i>Boolean</i>	Whether the restaurant has outdoor seating. Only applicable to restaurants.
ServesBreakfast	<i>Boolean</i>	Whether the restaurant serves breakfast. Only

		applicable to restaurants.
ServesLunch	<i>Boolean</i>	Whether the restaurant serves lunch. Only applicable to restaurants.
ServesDinner	<i>Boolean</i>	Whether the restaurant serves dinner. Only applicable to restaurants.
ServesCoffee	<i>Boolean</i>	Whether the restaurant serves coffee. Only applicable to restaurants.
ServesDrinks	<i>Boolean</i>	Whether the restaurant serves drinks. Only applicable to restaurants.
CulinaryTeam	<i>String</i>	Culinary team of the business. Applicable to restaurants or nightlife.
PharmaSafetyInfo	<i>String</i>	Pharmacy safety information. Applicable to pharmaceutical companies.
Affiliation	<i>String</i>	Affiliation of this person. Applicable to pages representing people.
Birthday	<i>String</i>	Birthday of this person. Applicable to pages representing people.
PersonalInfo	<i>String</i>	Personal information. Applicable to pages representing people.
PersonalInterests	<i>String</i>	Personal interests. Applicable to pages representing people.
ArtistsWeLike	<i>String</i>	Artists the band likes. Applicable to bands.
BandInterests	<i>String</i>	Band interests. Applicable to bands.
BandMembers	<i>String</i>	Members of the band. Applicable to bands.

Bio	<i>String</i>	Biography of the band. Applicable to bands.
BookingAgent	<i>String</i>	Booking agent of the band. Applicable to bands.
GeneralManager	<i>String</i>	General manager of the business. Applicable to restaurants or nightlife. Applicable to bands.
Hometown	<i>String</i>	Hometown of the band. Applicable to bands.
PressContact	<i>String</i>	Press contact information of the band. Applicable to bands.
RecordLabel	<i>String</i>	Record label of the band. Applicable to bands.
Awards	<i>String</i>	Awards information for the film or TV show. Applicable to films and TV shows.
DirectedBy	<i>String</i>	The director of the film or TV show. Applicable to films and TV shows.
Genre	<i>String</i>	The genre of the film or TV show. Applicable to films and TV shows.
Influences	<i>String</i>	Influences on the band. Applicable to bands.
PlotOutline	<i>String</i>	The plot outline of the film or TV show. Applicable to films and TV shows.
ProducedBy	<i>String</i>	The producer of the film. Applicable to films.
ReleaseData	<i>String</i>	The film's release data. Applicable to films and TV shows.



ScreenplayBy	<i>String</i>	The screenwriter of the film. Applicable to films and TV shows.
Starring	<i>String</i>	The cast of the film or TV show. Applicable to films and TV shows.
Studio	<i>String</i>	The studio for the film production. Applicable to films.
Network	<i>String</i>	The network the TV show airs on. Applicable to TV shows.
Schedule	<i>String</i>	The air schedule of the TV show. Applicable to TV shows.
Season	<i>String</i>	The current season of the TV show. Applicable to TV shows.
WrittenBy	<i>String</i>	The writer of the TV show. Applicable to TV shows.
Built	<i>String</i>	The information about when the vehicle was built. Applicable to vehicles.
Features	<i>String</i>	Features of the vehicle. Applicable to vehicles.
MPG	<i>String</i>	Miles per gallon for the vehicle. Applicable to vehicles.
Members	<i>String</i>	Members of this org. Applicable to pages representing team orgs.
InstagramBusinessAccountId	<i>String</i>	The business instagram account id associated with this page.

## 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. For more

information, see the WHERE clause section.

Name	Type	Description
offset	<i>String</i>	Which result to begin returning results from. Used for manual paging of results.

## Permissions

Query the Permissions the User has granted the current application.

### Columns

Name	Type	Description
PermissionName [KEY]	<i>String</i>	The name of the permission.
Status	<i>String</i>	The status of the requested permission.

## Photos

Query Photos associated with a Target. Accessing Photo information typically requires the user\_photos permission.

### Table Specific Information

Photos in Facebook are photos that are uploaded by a user (for example, to a page) and are part of a photo album.

## Select

When querying photos, specify a target. The target represents the user, page, album, or event that photos are being retrieved for. For example:

```
SELECT * FROM Photos WHERE Target = 'facebook'
```

If no target is specified, the currently authenticated user will be used as the target.

When querying photos, elements may be retrieved by specifying either the CreatedTime or the UpdatedTime. For example:

```
SELECT * FROM Photos WHERE Target='thesimpsons' AND CreatedTime >=
'1/1/2012' AND CreatedTime <= '10/1/2012'
```

## Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the photo.
Target	<i>String</i>	The Id or username of the target you are retrieving posts for or are posting to. This may be an album, event, page, or user.
FromId	<i>String</i>	Id of the user who uploaded the photo.
FromName	<i>String</i>	Name of the user who uploaded the photo.
FromPicture	<i>String</i>	Photo of the user who uploaded the photo.
FromCategory	<i>String</i>	Category of the user who uploaded the photo. FromCategory may only be retrieved if the other From* fields are not selected.

Link	<i>String</i>	A link to the photo on Facebook.
Name	<i>String</i>	The name of the photo.
Picture	<i>String</i>	A link a thumbnail of the photo.
Source	<i>String</i>	The source image of the photo.
Height	<i>Integer</i>	The height of the photo.
Width	<i>Integer</i>	The width of the photo.
Position	<i>Integer</i>	The position of the photo in the album.
Icon	<i>String</i>	A link to the icon Facebook displays when photos are published to the stream.
PlaceId	<i>String</i>	The Id of the location associated with the post, if any.
PlaceName	<i>String</i>	The name of the location associated with the post, if any.
Images	<i>String</i>	An aggregate of four different images for the photo.
CommentsCount	<i>Integer</i>	The number of comments for the photo.
LikesCount	<i>Integer</i>	The number of times the photo has been liked.
CommentsData	<i>String</i>	An aggregate of comments for this photo.

LikesData	<i>String</i>	An aggregate of likes data.
CreatedTime	<i>Datetime</i>	When the photo was uploaded.
UpdateTime	<i>Datetime</i>	When the photo was last updated.

## Places

Query the Places based on the supplied Id. Places are stored as Pages in Facebook.

### Table Specific Information

Places in Facebook are places that can charted on a map and visited. They will typically contain an address, coordinates, and basic information.

### Select

If you know the place Id, you may specify the Id to obtain information about the specific place. For example:

```
SELECT * FROM Places WHERE Id = '407032649344593'
```

If no Id is specified, no results will come back.

### Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the place.

Username	<i>String</i>	Username for the page of the place if any.
Name	<i>String</i>	The name of the place.
Category	<i>String</i>	The category of the place.
Categories	<i>String</i>	An aggregate of categories associated with the page if multiple categories are available.
Link	<i>String</i>	A link to the place.
LocationStreet	<i>String</i>	The street address of the place.
LocationCity	<i>String</i>	The the city for the place.
LocationState	<i>String</i>	The the state for the place.
LocationCountry	<i>String</i>	The the country for the place.
LocationZip	<i>String</i>	The the ZIP code of the place.
LocationLatitude	<i>String</i>	The latitude of the place.
LocationLongitude	<i>String</i>	The longitude of the place.
Phone	<i>String</i>	The phone number of the place, if available.

Picture	<i>String</i>	A link to the profile picture of the page for the place.
About	<i>String</i>	Basic information about the place.
Description	<i>String</i>	A description of the place, if available.
Website	<i>String</i>	A link to the website for the place.
Likes	<i>Integer</i>	Number of people who like the place.
Checkins	<i>Integer</i>	The total number of users who have checked in to the place.
TalkingAboutCount	<i>Integer</i>	The number of users talking about the place.
CanPost	<i>Boolean</i>	Boolean indicating if the authenticated user can post of the page for the place.
IsPublished	<i>Boolean</i>	A boolean indicating if the page for the place has been published.
IsCommunityPage	<i>Boolean</i>	Boolean indicating if this is a community Page.
PublicTransit	<i>String</i>	The public transit available for the place, if any.
ParkingStreet	<i>Boolean</i>	Boolean indicating if street parking is available.
ParkingLot	<i>Boolean</i>	Boolean indicating if a parking lot is available.
ParkingValet	<i>Boolean</i>	Boolean indicating if valet parking is available.

## 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. For more information, see the WHERE clause section.

Name	Type	Description
Distance	<i>String</i>	Optional input that can be specified if searching places. This is the maximum distance from the specified LocationLatitude and LocationLongitude in meters.
offset	<i>String</i>	Which result to begin returning results from. Used for manual paging of results.

## Ratings

List of ratings for a Facebook Page. This view requires authentication.

## Columns

Name	Type	Description
Target	<i>String</i>	The Id or name of the page to retrieve ratings for.
CreatedTime	<i>Datetime</i>	When the reviewer rated this object.
HasRating	<i>Boolean</i>	Boolean indicating if there was a rating included? (1-5 stars).
HasReview	<i>Boolean</i>	Boolean indicating if there was text in the rating.



OpenGraphID	<i>String</i>	Id of Open Graph story generated by the rating action.
Rating	<i>Integer</i>	Rating (1-5 stars).
ReviewText	<i>String</i>	The time the last message was posted in the conversation.
ReviewerID	<i>String</i>	Person who rated the object.
ReviewerName	<i>String</i>	Person who rated the object.

## SimpleInsights

Allows the retrieval of simple insights with a single value in the response.

### Table Specific Information

Simple insights refers to simple insights that can be retrieved from a page or post. These kinds of insights have a single value in the response and can only be filtered by a few values.

### Select

When selecting simple insights, a Target must be specified. In addition, an InsightName and Period should always be specified. For instance:

```
SELECT * FROM SimpleInsights WHERE Target = 'mypageid' AND InsightName =
'PAGE_IMPRESSIONS' AND Period = 'day'
```

Note that multiple insights may be specified via the IN operator. For example:

```
SELECT * FROM SimpleInsights WHERE Target = 'mypageid' AND InsightName
IN ('PAGE_IMPRESSIONS', 'PAGE_ENGAGED_USERS') AND Period = 'day'
```

Additionally, StartTime and EndTime can be used to specify a range where insight data should be drawn from:

```
SELECT * FROM SimpleInsights WHERE Target = 'mypageid' AND InsightName =
'PAGE_IMPRESSIONS' AND Period = 'day' AND EndTime >= '12/1/2015' AND
EndTime <= '12/31/2015'
```

Note: Please be aware that not all periods are available for all insights. Some will have day/week/days\_28 available. Others may only have lifetime. It is best to check against Facebook to see what periods are and are not supported. They can be found here: <https://developers.facebook.com/docs/graph-api/reference/insights>

## Columns

Name	Type	Description
RowNumber [KEY]	Int	The row number of the result.
EndTime	Date	The most recent date this insight data is relevant for.
Value	Long	Insight column for a singular value response.
Target	String	The target of the insight. This must always be specified.
InsightName	String	The name of the insight. This must always be specified. The available values are: PAGE_ACTIONS_POST_REACTIONS_ANGER_TOTAL,PAGE_ACTIONS_POST_REACTIONS_HAHA_TOTAL,PAGE_ACTIONS_POST_REACTIONS_LIKE_TOTAL,PAGE_ACTIONS_POST_REACTIONS_LOVE_TOTAL,PAGE_ACTIONS_POST_REACTIONS_SORRY_TOTAL,PAGE_ACTIONS_POST_REACTIONS_WOW_TOTAL,PAGE_CONSUMPTIONS,PAGE_CONSUMPTIONS_

---

UNIQUE,PAGE\_ENGAGED\_USERS,PAGE\_FAN\_ADDS,PAGE\_FAN\_ADDS\_UNIQUE,PAGE\_FAN\_REMOVES,PAGE\_FAN\_REMOVES\_UNIQUE,PAGE\_FANS,PAGE\_FANS\_ONLINE,PAGE\_FANS\_ONLINE\_PER\_DAY,PAGE\_IMPRESSIONS,PAGE\_IMPRESSIONS\_FREQUENCY\_DISTRIBUTION,PAGE\_IMPRESSIONS\_ORGANIC,PAGE\_IMPRESSIONS\_ORGANIC\_UNIQUE,PAGE\_IMPRESSIONS\_PAID,PAGE\_IMPRESSIONS\_PAID\_UNIQUE,PAGE\_IMPRESSIONS\_VIRAL,PAGE\_IMPRESSIONS\_VIRAL\_FREQUENCY\_DISTRIBUTION,PAGE\_IMPRESSIONS\_VIRAL\_UNIQUE,PAGE\_IMPRESSIONS\_NONVIRAL,PAGE\_IMPRESSIONS\_NONVIRAL\_UNIQUE,PAGE\_NEGATIVE\_FEEDBACK,PAGE\_NEGATIVE\_FEEDBACK\_UNIQUE,PAGE\_PLACES\_CHECKIN\_MOBILE,PAGE\_PLACES\_CHECKIN\_MOBILE\_UNIQUE,PAGE\_PLACES\_CHECKIN\_TOTAL,PAGE\_PLACES\_CHECKIN\_TOTAL\_UNIQUE,PAGE\_POST\_ENGAGEMENTS,PAGE\_POSTS\_IMPRESSIONS,PAGE\_POSTS\_IMPRESSIONS\_ORGANIC,PAGE\_POSTS\_IMPRESSIONS\_ORGANIC\_UNIQUE,PAGE\_POSTS\_IMPRESSIONS\_PAID,PAGE\_POSTS\_IMPRESSIONS\_PAID\_UNIQUE,PAGE\_POSTS\_IMPRESSIONS\_UNIQUE,PAGE\_POSTS\_IMPRESSIONS\_VIRAL,PAGE\_POSTS\_IMPRESSIONS\_VIRAL\_UNIQUE,PAGE\_CONTENT\_ACTIVITY,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_AUTOPLAYED,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_CLICK\_TO\_PLAY,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_ORGANIC,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_PAID,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_REPEAT\_VIEWS,PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_UNIQUE,PAGE\_VIDEO\_REPEAT\_VIEWS,PAGE\_VIDEO\_VIEWS,PAGE\_VIDEO\_VIEWS\_AUTOPLAYED,PAGE\_VIDEO\_VIEWS\_CLICK\_TO\_PLAY,PAGE\_VIDEO\_VIEWS\_ORGANIC,PAGE\_VIDEO\_VIEWS\_PAID,PAGE\_VIDEO\_VIEWS\_UNIQUE,PAGE\_VIEWS\_TOTAL,PAGE\_VIEWS\_LOGGED\_IN\_TOTAL,PAGE\_VIEWS\_LOGGED\_IN\_UNIQUE,PAGE\_VIEWS\_LOGOUT,PAGE\_VIEWS\_UNIQUE,POST\_CLICKS,POST\_CLICKS\_UNIQUE,POST\_ENGAGED\_USERS,POST\_IMPRESSIONS,POST\_IMPRESSIONS\_FAN,POST\_IMPRESSIONS\_FAN\_PAID,POST\_IMPRESSIONS\_FAN\_PAID\_UNIQUE,POST\_IMPRESSIONS\_FAN\_UNIQUE,POST\_IMPRESSIONS\_ORGANIC,POST\_IMPRESSIONS\_ORGANIC\_UNIQUE,POST\_IMPRESSIONS\_PAID,POST\_IMPRESSIONS\_PAID\_UNIQUE,POST\_IMPRESSIONS\_UNIQUE,POST\_IMPRESSIONS\_VIRAL,POST\_

---

---

OST\_VIDEO\_VIEWS\_10S,POST\_VIDEO\_VIEWS\_10S\_AUTOPLAYED,POST\_VIDEO\_VIEWS\_10S\_CLICKED\_TO\_PLAY,POST\_VIDEO\_VIEWS\_10S\_ORGANIC,POST\_VIDEO\_VIEWS\_10S\_PAID,POST\_VIDEO\_VIEWS\_10S\_SOUND\_ON,POST\_VIDEO\_VIEWS\_10S\_UNIQUE,POST\_VIDEO\_VIEWS\_ORGANIC,POST\_VIDEO\_VIEWS\_ORGANIC\_UNIQUE,POST\_VIDEO\_VIEWS\_PAID,POST\_VIDEO\_VIEWS\_PAID\_UNIQUE,POST\_VIDEO\_VIEWS\_SOUND\_ON

---

Period	<i>String</i>	The period for the insight. This controls how rows will be broken up (by day, week, 28 days, or lifetime). This must always be specified.  The allowed values are <i>day</i> , <i>week</i> , <i>days_28</i> , <i>lifetime</i> .
--------	---------------	---

---

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

---

## SimpleVideoInsights

Allows the retrieval of simple video insights with a single value in the response.

## Table Specific Information

Simple video insights refers to simple insights that can be retrieved from a video. These kinds of insights have a single value in the response and can only be filtered by a few values.

### Select

When selecting simple video insights, a Target must be specified. In addition, an InsightName should always be specified. For instance:

```
SELECT * FROM SimpleVideoInsights WHERE Target = 'myvideoid' AND
InsightName = 'TOTAL_VIDEO_VIEWS'
```

Note: Unlike other types of insights, video insights are only available for the entire lifetime of the video. Breakdowns in smaller units are not possible.

### Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
Value	<i>Long</i>	Insight column for a singular value response.
Target	<i>String</i>	The target of the insight. This must always be specified.
InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: TOTAL_VIDEO_VIEWS, TOTAL_VIDEO_VIEWS_UNIQUE, TOTAL_VIDEO_VIEWS_AUTOPLAYED, TOTAL_VIDEO_VIEWS_CLICKED_TO_PLAY, TOTAL_VIDEO_VIEWS_SOUND_ON, TOTAL_VIDEO_COMPLETE_VIEWS, TOTAL_VIDEO_COMPLETE_VIEWS_UNIQUE, TOTAL_VIDEO_COMPLETE_VIEWS_AUTO_

---

PLAYED,TOTAL\_VIDEO\_COMPLETE\_VIEWS\_CLICKED\_TO\_PLAY,TOTAL\_VIDEO\_10S\_VIEWS,TOTAL\_VIDEO\_10S\_VIEWS\_UNIQUE,TOTAL\_VIDEO\_10S\_VIEWS\_auto\_played,TOTAL\_VIDEO\_10S\_VIEWS\_CLICKED\_TO\_PLAY,TOTAL\_VIDEO\_10S\_VIEWS\_SOUND\_ON,TOTAL\_VIDEO\_AVG\_TIME\_WATCHED,TOTAL\_VIDEO\_VIEW\_TOTAL\_TIME,TOTAL\_VIDEO\_IMPRESSIONS,TOTAL\_VIDEO\_IMPRESSIONS\_UNIQUE,TOTAL\_VIDEO\_IMPRESSIONS\_VIRAL\_UNIQUE,TOTAL\_VIDEO\_IMPRESSIONS\_VIRAL,TOTAL\_VIDEO\_IMPRESSIONS\_FAN\_UNIQUE,TOTAL\_VIDEO\_IMPRESSIONS\_FAN

---

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## StoryAttachments

Query attachments from a post or comment.

### Table Specific Information

StoryAttachments in Facebook are attachments associated with a particular user or page story, like a page post or comment.

## Select

When querying StoryAttachments, the Target must be specified and set to that of a post or comment. For example, to retrieve all the attachments for a specific post, your SELECT statement could look something like this:

```
SELECT * FROM StoryAttachments WHERE Target = '15526475270_410830705612736'
```

## Columns

Name	Type	Description
Target	<i>String</i>	The Id or username of the target you are retrieving the wall for.
Type	<i>String</i>	The type of post.
Link	<i>String</i>	The link attached to the post.
Name	<i>String</i>	The name of the link.
Caption	<i>String</i>	The caption of the link, which appears beneath the link name.
Description	<i>String</i>	A description of the link, appears beneath the link caption.
Source	<i>String</i>	A URL to a flash movie or video file embedded within the post.
ObjectId	<i>String</i>	The Facebook object Id for an uploaded photo or video.

## TaggedBy

Query information about Posts, Statuses, Photos, and other entities that have tagged the User or Page. This view is a derivative of the Wall connection where only entries that have tagged the Target User or Page will be returned. In general it is only available for Pages.

## Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the entity that has tagged the user or page.
Target	<i>String</i>	The Id or username of the user you are retrieving from.
FromId	<i>String</i>	Id of the user who made the post.
FromName	<i>String</i>	Name of the user who made the post.
FromPicture	<i>String</i>	Picture of the user who made the post.
Message	<i>String</i>	The message of the post or status if available.
CommentsCount	<i>Integer</i>	The number of comments for the post.
LikesCount	<i>Integer</i>	The number of times the post has been liked.
SharesCount	<i>Integer</i>	The number of times the post has been shared.
CreatedTime	<i>Datetime</i>	When the post was created.



---

UpdateTime	<i>Datetime</i>	When the post was last updated.
------------	-----------------	---------------------------------

---

## 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. For more information, see the WHERE clause section.

---

Name	Type	Description
offset	<i>String</i>	Which result to begin returning results from. Used for manual paging of results.

---

## Users

Returns basic information about the authenticated user.

## Table Specific Information

Users in Facebook are the various user accounts on Facebook.

## Select

When selecting users, an Id must be specified. For example:

```
SELECT * FROM Posts WHERE Id = 'UserId'
```

If an Id is not specified, the information for the currently logged in user will be returned.

## Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the user.
Name	<i>String</i>	The full name of the user.
Picture	<i>String</i>	Picture of the user.
FirstName	<i>String</i>	The first name of the user.
MiddleName	<i>String</i>	The middle name of the user.
LastName	<i>String</i>	The last name of the user.
Birthday	<i>String</i>	The birthday of the user. Requires the user_birthday permission.
Email	<i>String</i>	The email address of the user. Requires the email permission.
HometownName	<i>String</i>	The hometown name of the user. Requires the user_hometown permission.
HometownId	<i>String</i>	The hometown name of the user. Requires the user_hometown permission.
LocationName	<i>String</i>	The current city name of the user. Requires the user_location permission.
LocationId	<i>String</i>	The current city Id of the user. Requires the user_location permission.

## VideoInsightsByActionType

Allows the retrieval of video insights by story action type.

## Table Specific Information

Video insights by action type refers to video insights that can be retrieved that detail totals for types of stories.

## Select

When selecting video insights, a Target must be specified.

```
SELECT * FROM VideoInsightsByActionType WHERE Target = 'myvideoid'
```

Note: Unlike other types of insights, video insights are only available for the entire lifetime of the video. Separation into smaller units is not possible.

## Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
Like	<i>Long</i>	Total likes.
Comment	<i>Long</i>	Total comments.
Share	<i>Long</i>	Total shares.
Target	<i>String</i>	The target of the insight. This is a video and must always be specified.

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## VideoInsightsByDistributionType

Allows the retrieval of video insights by distribution type.

### Table Specific Information

Video insights by reaction type refers to video insights that offer information on views your video got from different distribution sources.

### Select

When selecting video insights, a Target must be specified. In addition, an InsightName should always be specified. For instance:

```
SELECT * FROM VideoInsightsByDistributionType WHERE Target = 'myvideoid'  
AND InsightName = 'TOTAL_VIDEO_VIEWS_BY_DISTRIBUTION_TYPE'
```

Note: Unlike other types of insights, video insights are only available for the entire lifetime of the video. Separation into smaller units is not possible.

### Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
Page_Owned	<i>Long</i>	Views that were owned by the page.
Shared	<i>Long</i>	Views coming from shares.
Crossposted	<i>Long</i>	Views coming from posts that were crossposted.
Target	<i>String</i>	The target of the insight. This is a video and must always be specified.
InsightName	<i>String</i>	The name of the insight. This must always be specified. The available values are: TOTAL_VIDEO_VIEWS_BY_DISTRIBUTION_TYPE,TOTAL_VIDEO_VIEW_TIME_BY_DISTRIBUTION_TYPE

## 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. For more information, see the WHERE clause section.

Name	Type	Description
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

## VideoInsightsByReactionType

Allows the retrieval of video insights by reaction type.

## Table Specific Information

Video insights by reaction type refers to video insights that can be measured by the type of reaction.

## Select

When selecting video insights, a Target must be specified. For instance:

```
SELECT * FROM VideoInsightsByReactionType WHERE Target = 'myvideoid'
```

Note: Unlike other types of insights, video insights are only available for the entire lifetime of the video. Separation into smaller units is not possible.

## Columns

Name	Type	Description
RowNumber [KEY]	<i>Int</i>	The row number of the result.
Like	<i>Long</i>	Total like reactions.
Love	<i>Long</i>	Total love reactions.
Wow	<i>Long</i>	Total wow reactions.
Haha	<i>Long</i>	Total haha reactions.
Sorry	<i>Long</i>	Total sorry reactions.

---

Anger	<i>Long</i>	Total anger reactions.
Target	<i>String</i>	The target of the insight. This is a video and must always be specified.

---

## 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. For more information, see the WHERE clause section.

<b>Name</b>	<b>Type</b>	<b>Description</b>
FromDateTime	<i>String</i>	The earliest time for insight data to have been collected.
ToDateTime	<i>String</i>	The latest time for insight data to have been collected.

---

## Videos

Query Videos from a Target. Normally requires the user\_videos permission.

### Table Specific Information

Videos in Facebook are videos that have been uploaded by a user or to a page on Facebook.

## Select

When selecting videos, specify a target. The target represents the user or page who uploaded the video. If no target is specified, the currently authenticated user will be used as the target. For example:

```
SELECT * FROM Videos WHERE Target = 'facebook'
```

If you know the Id, you can specify the Id to obtain information about the specific video. For example:

```
SELECT * FROM Videos WHERE Id = 'VideoId'
```

When querying videos, retrieve elements by specifying either the CreatedTime or the UpdatedTime. For example:

```
SELECT * FROM Videos WHERE Target='thesimpsons' AND CreatedTime >=
'1/1/2012' AND CreatedTime <= '10/1/2012'
```

## Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the video.
Target	<i>String</i>	The Id or username of the target you are retrieving videos for.
FromId	<i>String</i>	Id of the user who uploaded the video.
FromName	<i>String</i>	Name of the user who uploaded the video.
FromPicture	<i>String</i>	Picture of the user who uploaded the video.



FromCategory	<i>String</i>	Category of the user who uploaded the video. FromCategory may only be retrieved if the other From* fields are not selected.
Description	<i>String</i>	A description of the video.
Picture	<i>String</i>	URL for the thumbnail of the video.
Source	<i>String</i>	A URL to the raw, playable video file.
EmbedHtml	<i>String</i>	The HTML element that may be embedded in an Web page to play the video.
Icon	<i>String</i>	The icon that Facebook displays when videos are published to the Feed.
Format	<i>String</i>	An aggregate of four different formats for the video.
TagsData	<i>String</i>	An aggregate of users tagged in the video, if any.
CommentsCount	<i>Integer</i>	The number of comments for the video.
LikesCount	<i>Integer</i>	The number of times the video has been liked.
CommentsData	<i>String</i>	An aggregate of comments for the video.
CreatedTime	<i>Datetime</i>	When the video was uploaded.
UpdatedTime	<i>Datetime</i>	When the video was last updated.

## Wall

Query Posts from the Wall of a Target.

### Table Specific Information

Walls in Facebook are a collection of the various posts and updates made to a user's profile or wall.

### Select

When selecting from a wall, specify a target. The target represents the user, page, group, application, or other valid entity that may have a wall associated with it. If no target is specified, the currently authenticated user will be used as the target. For example:

```
SELECT * FROM Wall WHERE Target = 'facebook'
```

When querying the wall, elements may be retrieved by specifying either the CreatedTime or the UpdatedTime. For example:

```
SELECT * FROM Wall WHERE Target='facebook' AND CreatedTime >= '1/1/2012'
AND CreatedTime <= '2/1/2012'
```

### Insert

While the wall may not be directly inserted to, you may post to a wall by issuing an insert with the correct target using the [Posts](#) table.

### Columns

Name	Type	Description
ID [KEY]	String	The Id of the post.

Target	<i>String</i>	The Id or username of the target you are retrieving the wall for.
FromId	<i>String</i>	Id of the user who made the post.
FromName	<i>String</i>	Name of the user who made the post.
FromPicture	<i>String</i>	Picture of the user who made the comment.
FromCategory	<i>String</i>	Category of the user who made the post. FromCategory may only be retrieved if the other From* fields are not selected.
ToData	<i>String</i>	An aggregate of users the post was made to.
Message	<i>String</i>	The message of the post.
MessageTags	<i>String</i>	An aggregate of objects tagged in the message such as Users, Pages, etc.
Picture	<i>String</i>	A link to the picture included in the post.
Icon	<i>String</i>	Link to an icon representing the type of post.
Actions	<i>String</i>	An aggregate of available actions on the post (such as commenting or liking).
CommentsCount	<i>Integer</i>	The number of comments for the post.
LikesCount	<i>Integer</i>	The number of times the post has been liked.

SharesCount	<i>Integer</i>	The number of times the post has been shared.
PlaceId	<i>String</i>	The Id of the location associated with the post, if any.
PlaceName	<i>String</i>	The name of the location associated with the post, if any.
Attachments	<i>String</i>	An aggregate for the attachments of the post.
ApplicationId	<i>String</i>	Id of the application this post came from.
ApplicationNamespace	<i>String</i>	Information about the application used to create the entity.
Story	<i>String</i>	Text of stories not intentionally generated by users, such as those generated when two users become friends; you must have the 'Include recent activity stories' migration enabled in your app to retrieve these stories.
StoryTags	<i>String</i>	An aggregate of objects (users, pages, etc.) associated with the story.
LikesData	<i>String</i>	An aggregate of like data.
CommentsData	<i>String</i>	An aggregate of comments for this post.
CreatedTime	<i>Datetime</i>	When the post was created.
UpdatedTime	<i>Datetime</i>	When the post was last updated.

## Stored Procedures

Stored procedures are function-like interfaces that extend the functionality of the adapter beyond simple SELECT/INSERT/UPDATE/DELETE operations with Facebook.

Stored procedures accept a list of parameters, perform their intended function, and then return, if applicable, any relevant response data from Facebook, along with an indication of whether the procedure succeeded or failed.

### Facebook Adapter Stored Procedures

Name	Description
<a href="#">CreateInsightSchema</a>	Creates a schema file for an insight.
<a href="#">GetOAuthAccessToken</a>	Gets an authentication token from Facebook.
<a href="#">GetOAuthAuthorizationURL</a>	Gets an AuthURL from Facebook. You will request the auth token from this URL.
<a href="#">RefreshOAuthAccessToken</a>	Refreshes the OAuth token.

## CreateInsightSchema

Creates a schema file for an insight.

### Input

Name	Type	Accepts Output Streams	Description
InsightName	<i>String</i>	<i>False</i>	The name of the insight to create a table for.

The allowed values are *PAGE\_CONSUMPTIONS*, *PAGE\_CONSUMPTIONS\_BY\_CONSUMPTION\_TYPE*, *PAGE\_CONSUMPTIONS\_BY\_CONSUMPTION\_TYPE\_UNIQUE*, *PAGE\_CONSUMPTIONS\_UNIQUE*, *PAGE\_ENGAGED\_USERS*, *PAGE\_FAN\_ADDS*, *PAGE\_FAN\_ADDS\_UNIQUE*, *PAGE\_FAN\_REMOVES*, *PAGE\_FAN\_REMOVES\_UNIQUE*, *PAGE\_FANS*, *PAGE\_FANS\_BY\_LIKE\_SOURCE*, *PAGE\_FANS\_BY\_LIKE\_SOURCE\_UNIQUE*, *PAGE\_FANS\_BY\_UNLIKE\_SOURCE\_UNIQUE*, *PAGE\_FANS\_CITY*, *PAGE\_FANS\_COUNTRY*, *PAGE\_FANS\_GENDER\_AGE*, *PAGE\_FANS\_LOCALE*, *PAGE\_FANS\_ONLINE*, *PAGE\_FANS\_ONLINE\_PER\_DAY*, *PAGE\_IMPRESSIONS*, *PAGE\_IMPRESSIONS\_BY\_AGE\_GENDER\_UNIQUE*, *PAGE\_IMPRESSIONS\_BY\_CITY\_UNIQUE*, *PAGE\_IMPRESSIONS\_BY\_COUNTRY\_UNIQUE*, *PAGE\_IMPRESSIONS\_BY\_LOCALE\_UNIQUE*, *PAGE\_IMPRESSIONS\_BY\_STORY\_TYPE*, *PAGE\_IMPRESSIONS\_BY\_STORY\_TYPE\_UNIQUE*, *PAGE\_IMPRESSIONS\_FREQUENCY\_DISTRIBUTION*, *PAGE\_IMPRESSIONS\_ORGANIC*, *PAGE\_IMPRESSIONS\_ORGANIC\_UNIQUE*, *PAGE\_IMPRESSIONS\_PAID*, *PAGE\_IMPRESSIONS\_PAID\_UNIQUE*, *PAGE\_IMPRESSIONS\_UNIQUE*, *PAGE\_IMPRESSIONS\_VIRAL*, *PAGE\_IMPRESSIONS\_VIRAL\_FREQUENCY\_DISTRIBUTION*, *PAGE\_IMPRESSIONS\_VIRAL\_UNIQUE*, *PAGE\_NEGATIVE\_FEEDBACK*, *PAGE\_NEGATIVE\_FEEDBACK\_BY\_TYPE*, *PAGE\_NEGATIVE\_FEEDBACK\_BY\_TYPE\_UNIQUE*, *PAGE\_NEGATIVE\_FEEDBACK\_UNIQUE*, *PAGE\_PLACES\_CHECKIN\_MOBILE*, *PAGE\_PLACES\_CHECKIN\_MOBILE\_UNIQUE*, *PAGE\_PLACES\_CHECKIN\_TOTAL*, *PAGE\_PLACES\_CHECKIN\_TOTAL\_UNIQUE*, *PAGE\_PLACES\_CHECKINS\_BY\_AGE\_GENDER*, *PAGE\_PLACES\_CHECKINS\_BY\_COUNTRY*, *PAGE\_PLACES\_CHECKINS\_BY\_LOCALE*, *PAGE\_POSITIVE\_FEEDBACK\_BY\_TYPE*, *PAGE\_POSITIVE\_FEEDBACK\_*

---

BY\_TYPE\_UNIQUE, PAGE\_POSTS\_IMPRESSIONS,  
 PAGE\_POSTS\_IMPRESSIONS\_FREQUENCY\_  
 DISTRIBUTION, PAGE\_POSTS\_IMPRESSIONS\_  
 ORGANIC, PAGE\_POSTS\_IMPRESSIONS\_ORGANIC\_  
 UNIQUE, PAGE\_POSTS\_IMPRESSIONS\_PAID,  
 PAGE\_POSTS\_IMPRESSIONS\_PAID\_UNIQUE,  
 PAGE\_POSTS\_IMPRESSIONS\_UNIQUE, PAGE\_  
 POSTS\_IMPRESSIONS\_VIRAL, PAGE\_POSTS\_  
 IMPRESSIONS\_VIRAL\_UNIQUE, PAGE\_STORIES,  
 PAGE\_STORIES\_BY\_STORY\_TYPE, PAGE\_  
 STORYTELLERS, PAGE\_CONTENT\_ACTIVITY\_BY\_  
 AGE\_GENDER\_UNIQUE, PAGE\_CONTENT\_  
 ACTIVITY\_BY\_CITY\_UNIQUE, PAGE\_CONTENT\_  
 ACTIVITY\_BY\_COUNTRY\_UNIQUE, PAGE\_  
 CONTENT\_ACTIVITY\_BY\_LOCALE\_UNIQUE, PAGE\_  
 STORYTELLERS\_BY\_STORY\_TYPE, PAGE\_TAB\_  
 VIEWS\_LOGIN\_TOP, PAGE\_TAB\_VIEWS\_LOGIN\_  
 TOP\_UNIQUE, PAGE\_TAB\_VIEWS\_LOGOUT\_TOP,  
 PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S, PAGE\_  
 VIDEO\_COMPLETE\_VIEWS\_30S\_AUTOPLAYED,  
 PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_CLICK\_TO\_  
 PLAY, PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_  
 ORGANIC, PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_  
 PAID, PAGE\_VIDEO\_COMPLETE\_VIEWS\_30S\_  
 REPEAT\_VIEWS, PAGE\_VIDEO\_COMPLETE\_VIEWS\_  
 30S\_UNIQUE, PAGE\_VIDEO\_REPEAT\_VIEWS, PAGE\_  
 VIDEO\_VIEW\_TIME, PAGE\_VIDEO\_VIEWS, PAGE\_  
 VIDEO\_VIEWS\_AUTOPLAYED, PAGE\_VIDEO\_VIEWS\_  
 CLICK\_TO\_PLAY, PAGE\_VIDEO\_VIEWS\_ORGANIC,  
 PAGE\_VIDEO\_VIEWS\_PAID, PAGE\_VIDEO\_VIEWS\_  
 UNIQUE, PAGE\_VIEWS, PAGE\_VIEWS\_EXTERNAL\_  
 REFERRALS, PAGE\_VIEWS\_LOGIN, PAGE\_VIEWS\_  
 LOGIN\_UNIQUE, PAGE\_VIEWS\_LOGOUT, PAGE\_  
 VIEWS\_UNIQUE, POST\_CONSUMPTIONS, POST\_  
 CONSUMPTIONS\_BY\_TYPE, POST\_  
 CONSUMPTIONS\_BY\_TYPE\_UNIQUE, POST\_  
 CONSUMPTIONS\_UNIQUE, POST\_ENGAGED\_  
 USERS, POST\_IMPRESSIONS, POST\_

---

Target	<i>String</i>	<i>False</i>	The default target for this insight.
FromDateTime	<i>String</i>	<i>False</i>	The earliest date and time value for which to return insight data. This will override the Periods input and in some insights is required.
ToDateTime	<i>String</i>	<i>False</i>	The latest date and time value for which to return insight data. This will override the Periods input and in some insights is required.
Description	<i>String</i>	<i>False</i>	An optional description for the insight. This will override the description that comes from the data source.
FileStream	<i>String</i>	<i>True</i>	An instance of an output stream where file data is written to.

## Result Set Columns

<b>Name</b>	<b>Type</b>	<b>Description</b>
Result	<i>String</i>	Returns Success or Failure.
SchemaFile	<i>String</i>	The generated schema file. You will need to set the Location connection property to the folder containing your schema files.
Columns	<i>String</i>	The number of columns found.

## GetOAuthAccessToken

Gets an authentication token from Facebook.



## Input

Name	Type	Description
AuthMode	<i>String</i>	<p>The type of authentication mode to use. Select App for getting authentication tokens via a windows forms app. Select Web for getting authentication tokens via a web app.</p> <p>The allowed values are <i>APP</i>, <i>WEB</i>.</p> <p>The default value is <i>APP</i>.</p>
Scope	<i>String</i>	<p>A comma-separated scope of permissions to request from the user. Please check the Facebook API for scope of available permissions. Permissions that may be required depending on your use case are: user_birthday,user_photos,user_videos,user_likes,user_hometown,user_location,read_insights,pages_manage_metadata,pages_read_engagement,pages_read_user_content,pages_messaging,business_management,instagram_basic,instagram_manage_insights.</p>
CallbackUrl	<i>String</i>	<p>The URL the user will be redirected to after authorizing your application. This value must match the Site URL and App Domain you have specified in the Facebook app settings. Only needed when the Authmode parameter is Web.</p>
Verifier	<i>String</i>	<p>The verifier returned from Facebook after the user has authorized your app to have access to their data. This value will be returned as a parameter to the callback URL in GetAuthURL.</p>
AuthType	<i>String</i>	<p>An optional parameter to pass Facebook. Set the value to rerequest to request the same permissions again if the user declined some of them.</p>
State	<i>String</i>	<p>Any value that you wish to be sent with the callback.</p>

## Result Set Columns

Name	Type	Description
OAuthAccessToken	<i>String</i>	The access token used for communication with Facebook.
OAuthRefreshToken	<i>String</i>	This is actually the same as the OAuthAccessToken. It can be used internally to determine if the token can be refreshed. Tokens are refreshed by Facebook automatically.
ExpiresIn	<i>String</i>	The remaining lifetime on the access token. A -1 denotes that it will not expire.

## GetOAuthAuthorizationURL

Gets an AuthURL from Facebook. You will request the auth token from this URL.

### Input

Name	Type	Description
CallbackUrl	<i>String</i>	The URL the user will be redirected to after authorizing your application. This value must match the Site URL and App Domain in the Facebook app settings.
Scope	<i>String</i>	A comma-separated scope of permissions to request from the user. Please check the Facebook API for scope of available permissions. Permissions that may be required depending on your use case are: user_birthday,user_photos,user_videos,user_likes,user_hometown,user_location,read_insights,pages_manage_metadata,pages_read_engagement,pages_read_user_content,pages_messaging,business_management,instagram_basic,instagram_manage_insights.
AuthType	<i>String</i>	An optional parameter to pass Facebook. Set the value to rerequest to request the same permissions again if the user declined some of them.

State	<i>String</i>	Any value that you wish to be sent with the callback.
-------	---------------	---

## Result Set Columns

Name	Type	Description
URL	<i>String</i>	The authorization URL, entered into a Web browser to obtain the verifier token and authorize your app.

## RefreshOAuthAccessToken

Refreshes the OAuth token.

### Input

Name	Type	Description
OAuthRefreshToken	<i>String</i>	The refresh token returned when the OAuth Token was first created.

## Result Set Columns

Name	Type	Description
OAuthAccessToken	<i>String</i>	The authentication token returned from Facebook.
OAuthRefreshToken	<i>String</i>	A token that may be used to obtain a new access token.

---

ExpiresIn	String	The remaining lifetime on the access token.
-----------	--------	---

---

## 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](#).

### Authentication

---

Property	Description
<a href="#">Version</a>	The Facebook Graph API version to use.
<a href="#">AuthenticateAsPage</a>	The name or Id of a page to authenticate as when making requests to Facebook.

---

### 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="#">CallbackURL</a>	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
<a href="#">Scope</a>	The scope to use when authenticating to Facebook.
<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.

## 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.

## Miscellaneous

Property	Description
<a href="#">AggregateFormat</a>	The format aggregate or collection columns should return in.
<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="#">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 Facebook.
<a href="#">Readonly</a>	You can use this property to enforce read-only access to Facebook from the provider.
<a href="#">Target</a>	A default target if none is specified. Used for some tables, such as Comments, where a target may be specified.
<a href="#">Timeout</a>	The value in seconds until the timeout error is thrown, canceling the operation.

<a href="#">UploadLinkedMedia</a>	Upload linked photos or videos before inserting a new Post.
<a href="#">UserDefinedViews</a>	A filepath pointing to the JSON configuration file containing your custom views.

## 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="#">Version</a>	The Facebook Graph API version to use.
<a href="#">AuthenticateAsPage</a>	The name or Id of a page to authenticate as when making requests to Facebook.

## Version

The Facebook Graph API version to use.

### Data Type

string

### Default Value

"15.0"

### Remarks

The Facebook Graph API version to use. Generally this property does not need to be set.

## AuthenticateAsPage



The name or Id of a page to authenticate as when making requests to Facebook.

## Data Type

string

## Default Value

""

## Remarks

The Id of a page to retrieve data from. The page must be managed by the authenticated user; you can obtain the Ids for all such pages by querying the [Pages](#) view.

# 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://.

CallbackURL	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.
Scope	The scope to use when authenticating to Facebook.
OAuthVerifier	The verifier code returned from the OAuth authorization URL.
OAuthRefreshToken	The OAuth refresh token for the corresponding OAuth access token.
OAuthExpiresIn	The lifetime in seconds of the OAuth AccessToken.
OAuthTokenTimestamp	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%\\CData\\Facebook Data Provider\\OAuthSettings.txt"

## Remarks

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

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

If left unspecified, the default location is "%APPDATA%\\CData\\Facebook 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

## CallbackURL

The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your app settings.

## Data Type

string

## Default Value

""

## Remarks

During the authentication process, the OAuth authorization server redirects the user to this URL. This value must match the callback URL you specify in your app settings.

## Scope

The scope to use when authenticating to Facebook.

## Data Type

string

## Default Value

""

## Remarks

The scope that are used when authenticating to Facebook give your App access to request additional information that may be required for certain tables. Individual scope may be entered in a comma separated list. Leaving it blank will use the defaults. The Scope is only used if you have specified your own [OAuthClientId](#) and [OAuthClientSecret](#).

## Requesting Additional Permissions

You may find while using the adapter that Facebook returns an error stating your app does not have permissions to do a certain action. To resolve this, you will need to generate a

new OAuth access token with the required scope. Set the Scope property in the authentication step. You can find a list of available Facebook scopes here: <http://developers.facebook.com/docs/authentication/permissions/>.

## 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.

# 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----- MIICHTCCAe4CAQAwDQYJKoZIhvd.....Qw == -----END CERTIFICATE-----
A path to a local file containing the certificate	C:\cert.cer
The public key (example shortened for brevity)	-----BEGIN RSA PUBLIC KEY----- MIGfMA0GCSq.....AQAB -----END RSA PUBLIC KEY-----
The MD5 Thumbprint (hex values can also be either space or colon separated)	34e929226ae0819f2ec14b4a3d904f801c
The SHA1 Thumbprint (hex values can also be either space or colon separated)	bb150d

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 Facebook

		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.

---

## Location

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

## Data Type

string

## Default Value

"%APPDATA%\\CData\\Facebook 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\\Facebook 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

## 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="#">AggregateFormat</a>	The format aggregate or collection columns should return in.
<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="#">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 Facebook.
<a href="#">Readonly</a>	You can use this property to enforce read-only access to Facebook from the provider.
<a href="#">Target</a>	A default target if none is specified. Used for some tables, such as Comments, where a target may be specified.
<a href="#">Timeout</a>	The value in seconds until the timeout error is thrown, canceling the operation.
<a href="#">UploadLinkedMedia</a>	Upload linked photos or videos before inserting a new Post.
<a href="#">UserDefinedViews</a>	A filepath pointing to the JSON configuration file containing your custom views.

## AggregateFormat

The format aggregate or collection columns should return in.

### Possible Values

JSON, XML

### Data Type

string

### Default Value

"JSON"

### Remarks

The format aggregate or collection columns should return in.

## 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.

## 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 Facebook.

### Data Type

string

### Default Value

""

### Remarks

The Pagesize property affects the maximum number of results to return per page from Facebook. Sometimes you may get an error asking you to request less data. The frequency of such errors can be reduced by reducing the pagesize. The maximum pagesize tends to be about 100 per page.

## Readonly

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

## Target

A default target if none is specified. Used for some tables, such as Comments, where a target may be specified.

## Data Type

string

## Default Value

""

## Remarks

A default target if none is specified. Used for some tables, such as Comments, where a target may be specified.

## 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.

## UploadLinkedMedia

Upload linked photos or videos before inserting a new Post.

### Data Type

bool

### Default Value

false

## Remarks

The UploadLinkedMedia determines whether to upload media before inserting a Post. If set to True, when you attempt to insert a new Post with the Link column, the driver will first attempt to resolve the URL and determine if the URL is referencing a photo or a video. If so, the photo or video will be uploaded first, then a new Post containing the media will be created. If False, then the new Post will be created as a Link Post.

## UserDefinedViews

A filepath pointing to the JSON configuration file containing your custom views.

### Data Type

string

### Default Value

""

## Remarks

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

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

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

# 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.

## 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.

## 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 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, TIBCO O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and 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.