



# **TIBCO® Data Virtualization**

## **Sharepoint Adapter Guide**

Version 8.7.0 | October 2023

# Contents

---

<b>Contents</b>	<b>2</b>
<b>SharePoint Adapter</b>	<b>4</b>
Getting Started	4
Basic Tab	5
Logging	14
Creating a Custom AzureAD App	16
Connecting to REST API	17
Using Kerberos	17
Fine-Tuning Data Access	19
Changelog	20
Advanced Features	27
User Defined Views	28
SSL Configuration	31
Firewall and Proxy	31
Query Processing	32
Logging	33
SQL Compliance	36
SELECT Statements	37
SELECT INTO Statements	39
INSERT Statements	40
UPDATE Statements	41
DELETE Statements	41
EXECUTE Statements	42
PIVOT and UNPIVOT	43
SOAP Data Model	44
Tables	45
Stored Procedures	61

REST Data Model .....	88
Views .....	89
Stored Procedures .....	113
Data Type Mapping .....	132
Connection String Options .....	133
Authentication .....	140
SSO .....	146
OAuth .....	148
JWT OAuth .....	156
Kerberos .....	160
SSL .....	165
Firewall .....	171
Proxy .....	174
Logging .....	181
Schema .....	181
Miscellaneous .....	183
<b>TIBCO Product Documentation and Support Services .....</b>	<b>196</b>
How to Access TIBCO Documentation .....	196
How to Contact TIBCO Support .....	197
Release Version Support .....	197
How to Join TIBCO Community .....	198
<b>Legal and Third-Party Notices .....</b>	<b>199</b>

# SharePoint Adapter

---

## SharePoint Version Support

The adapter supports all versions of Microsoft SharePoint that support the SOAP API. This includes: Windows SharePoint Services 3.0, SharePoint Server 2007+ (2010, 2013, etc.), and SharePoint Online. The adapter models the custom lists of your SharePoint site as bidirectional tables; when you connect, the adapter retrieves the metadata for these tables by calling SharePoint Web services. Supported authentication schemes are NTLM, Basic, Digest, Forms, Kerberos, SSO, STS (security token services), and SharePoint authentication cookies.

## SQL Compliance

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

# Getting Started

## Connecting to SharePoint

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

To deploy the adapter, you can execute the server\_util utility via the command line by

1. Unzip the tdv.sharepoint.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.sharepoint/tdv.sharepoint.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 SharePoint
```

## Basic Tab

### Connecting to SharePoint

SharePoint works with all Lists and Documents in the global Microsoft Sharepoint site, as well as with individual sites. Set the URL connection property to your Site Collection URL to work with all Lists and Documents, or set it to a specific Site URL to work with Lists and Documents in that site only.

URL	Example URL
Site Collection	https://teams.contoso.com
Site	https://teams.contoso.com/teamA or https://teamA.contoso.com

In addition to providing the URL, use one of the following sets of connection properties to authenticate to SharePoint. The default values make it easy to connect in most environments, as shown below. Sharepoint supports online (cloud-based) and on-premises architectures. Each architecture supports a different set of authentication schemes.

Online and on-prem authentication are discussed in turn below.

### Authenticating to SharePoint Online

Set SharePointEdition to "SharePoint Online" and set the User and Password to the credentials you use to log onto SharePoint, for example, the credentials to your Microsoft Online Services account.

The following authentication schemes are supported:

- AzureAD
- Single sign-on (SSO) schemes
- MSI
- OAuthJWT

The following SSO identity providers are supported:

- ADFS
- OneLogin
- Okta
- PingFederate

If the user account domain is different from the domain configured with the identity provider, set SSODomain to the latter. This property may be required for any SSO.

## AzureAD

Azure Active Directory (AzureAD) is a connection type that leverages OAuth to authenticate. OAuth requires the authenticating user to interact with SharePoint using an internet browser. The driver facilitates this in several ways as described below. Set your AuthScheme to **AzureAD**. The AzureAD flows described below assume that you have done so.

Your organization may require Admin Consent when authorizing a new AzureAD application for your Azure Tenant. In all AzureAD flows, any initial installation and use of an AzureAD application requires that an administrator approve the application for their Azure Tenant.

## 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.

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.

### 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. Choose one of these options:

- If you are using the Embedded OAuth Application click [SharePoint OAuth endpoint](#) to open the endpoint in your browser.
- 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.

Then call the [GetOAuthAuthorizationURL](#) stored procedure with the appropriate CallbackURL. Open the URL returned by the stored procedure in a browser.

2. Log in and grant permissions to the adapter. You are then redirected to the callback URL, which contains the verifier code.
3. Save the value of the verifier code. Later you will set this in the OAuthVerifier connection property.

Next, you need to exchange the OAuth verifier code for OAuth refresh and access tokens. Set the following properties:

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

- InitiateOAuth: Set this to **REFRESH**.

- OAuthVerifier: Set this to the verifier code.
- OAuthClientId: (custom applications only) Set this to the client Id in your custom OAuth application settings.
- OAuthClientSecret: (custom applications only) Set this to the client secret in the custom OAuth application settings.
- OAuthSettingsLocation: Set this to the path to the file where the driver saves the OAuth token values that persist across connections.

After the OAuth settings file is generated, you need to re-set the following properties to connect:

- InitiateOAuth: Set this to **REFRESH**.
- OAuthClientId: (custom applications only) Set this to the client Id assigned when you registered your application.
- OAuthClientSecret: (custom applications only) Set this to the client secret assigned when you registered your application.
- OAuthSettingsLocation: Set this to the file containing the encrypted OAuth authentication values. Make sure this file grants read and write permissions to the adapter to enable the automatic refreshing of the access token.

## Option 2: Transfer OAuth Settings

Prior to connecting on a headless machine, you need to create and install a connection with the driver on a device that supports an internet browser. Set the connection properties as described in "Desktop Applications" above.

After completing the instructions in "Desktop Applications", the resulting authentication values are encrypted and written to the path specified by OAuthSettingsLocation. The default filename is *OAuthSettings.txt*.

Once you have successfully tested the connection, copy the OAuth settings file to your headless machine.

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

- InitiateOAuth: Set this to **REFRESH**.
- OAuthClientId: (custom applications only) Set this to the client Id assigned when you registered your application.
- OAuthClientSecret: (custom applications only) Set this to the client secret assigned



when you registered your application.

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

## Azure Password

To connect using your Azure credentials directly, specify the following connection properties:

- AuthScheme: Set this to **AzurePassword**.
- User: Set this to your user account you use to connect to Azure.
- Password: Set this to the password you use to connect to Azure.
- AzureTenant: Set this to the **Directory (tenant) ID**, found on the Overview page of the OAuth app used to authenticate to SharePoint on Azure.

## OAuth: Admin Consent

*Admin consent* refers to when the Admin for an Azure Active Directory tenant grants permissions to an application that requires an administrator in your organization to consent to the use case. The embedded application within the SharePoint Adapter, contains no permissions that require administrator consent. Therefore, this information applies only to custom applications.

### Administrator Consent Permissions

When creating a new OAuth application in the Azure Portal, you must specify which permissions the application requires. Some permissions may be marked with "Admin Consent Required". For example, all Groups permissions require Admin Consent. If your application requires admin consent, there are two ways you can do this.

The easiest way to grant admin consent is to have an administrator log into the [Azure Portal](#) and navigate to the application you have created in App Registrations. Under API Permissions, click **Grant Consent**, which grants permissions on the tenant under which it was created.

If your organization has multiple tenants or you need to grant application permissions for other tenants outside your organization, use the [GetAdminConsentURL](#) stored procedure to generate the Admin Authorization URL. After the OAuth application is successfully authorized, it returns a Boolean indicating that permissions have been granted.

After the administrator has approved the OAuth Application, you can continue to authenticate.

## Credentials Using a Custom App

Follow the steps below to create a custom OAuth app and obtain the connection properties for the OAuth authentication using Sharepoint App.

### Register Add-In

1. Navigate to the Register Add-In page by entering the url as : *https://{sitename}.SharePoint.com/\_layouts/15/appregnew.aspx* .
2. In the "App Information" section, click **Generate**, which is located next to the Client Id and Client Secret textboxes to generate the respective values.
3. Set Title, App Domain and Redirect URI by filling respective text boxes.
4. Click **Create**, which registers the add-in and returns the success message with created information.

### Grant Permissions to Add-In

1. Navigate to the SharePoint site.
2. Enter the URL: *https://{sitename}.sharepoint.com/\_layouts/15/appinv.aspx* in the browser. This redirects to the Grant permission page.
3. Enter the Client ID (which you generated earlier), in **AppId** and click Lookup button. That will populate the value to other textboxes in Title, App Domain and Redirect Url.
4. Now enter the below permission request in XML format.

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest Scope="http://sharepoint/content/tenant"
    Right="FullControl"/>
</AppPermissionRequests>
```

5. When you click on Create you'll be presented with a permission consent dialog. Press Trust It to grant the permissions.

## ADFS

Set the AuthScheme to **ADFS**. You need to set the following connection properties:

- User: Set this to the ADFS user.
- Password: Set this to ADFS password for the user.
- SSODomain (optional): The domain configured with the ADFS identity provider.

Below is an example connection string:

```
AuthScheme=ADFS;User=ADFSUserName;Password=ADFSPassword;URL='http://sharepointserver/mysite';
```

## Okta

Set the AuthScheme to **Okta**. The following connection properties are used to connect to Okta:

- User: Set this to the Okta user.
- Password: Set this to Okta password for the user.
- SSODomain (optional): The domain configured with the OKTA identity provider.

The following is an example connection string:

```
AuthScheme=Okta;User=oktaUserName;Password=oktaPassword;URL='http://sharepointserver/mysite';
```

## OneLogin

Set the AuthScheme to **OneLogin**. The following connection properties are used to connect to OneLogin:

- User: Set this to the OneLogin user.
- Password: Set this to OneLogin password for the user.
- SSODomain (optional): The domain configured with the OneLogin identity provider.

The following is an example connection string:

```
AuthScheme=OneLogin;User=OneLoginUserName;Password=OneLoginPassword;URL='http://sharepointserver/mysite';
```

## PingFederate

Set the AuthScheme to **PingFederate**. The following connection properties are used to connect to PingFederate:

- User: Set this to the PingFederate user.
- Password: Set this to PingFederate password for the user.
- SSODomain (optional): The domain configured with the PingFederate identity provider.

The following is an example connection string:

```
AuthScheme=PingFederate;User=PingFederateUserName;Password=PingFederatePassword;URL='http://sharepointserver/mysite';
```

## SharePointOAuth

Set the AuthScheme to **SharePointOAuth**. The following connection properties are used to connect to SharePointOAuth:

- InitiateOAuth: Set this to **GETANDREFRESH**.
- User: Set this to the SharePointOAuth user.
- Password: Set this to SharePointOAuth password for the user.

The following is an example connection string:

```
Schema=REST;InitiateOAuth=GETANDREFRESH;AuthScheme=SharepointOAuth;URL=https://rssbuscrm.sharepoint.com;User=SharePointUserName;Password=SharePointPassword;SharePointEdition='SharepointOnline';
```

## OAuthJWT Certificate

Set the AuthScheme to **OAuthJWT**. The following connection properties are used to connect to SharePoint:

- OAuthGrantType: Set this to **CLIENT**.
- AzureTenant: Set this to the tenant you wish to connect to.
- OAuthJWTCert: Set this to the JWT Certificate store.

- OAuthJWTIssuer: Set this to the OAuth client Id.

## MSI

If you are running SharePoint on an Azure VM, you can leverage Managed Service Identity (MSI) credentials to connect:

- AuthScheme: Set this to **AzureMSI**.

The MSI credentials are automatically obtained for authentication.

## Authenticating to SharePoint On-Premises

Set SharePointEdition to "SharePoint On-Premises" to use the following authentication types.

### Windows (NTLM)

This is the most common authentication type. As such, the adapter is preconfigured to use NTLM as the default; simply set the Windows User and Password to connect.

### Kerberos

Set the AuthScheme to **NEGOTIATE**. The following connection properties are used to connect with Kerberos:

- KerberosKDC: Set this to the **host name or IP Address** of your Kerberos KDC machine.
- KerberosRealm: Set this to **the realm of the SharePoint Kerberos principal**. This is the value after the '@' symbol (for instance, EXAMPLE.COM) of the **principal value** (for instance, MyService/MyHost@EXAMPLE.COM).
- KerberosSPN: Set this to the **service and host of the SharePoint Kerberos Principal**. This is the value prior to the '@' symbol (for instance, MyService/MyHost) of the **principal value** (for instance, MyService/MyHost@EXAMPLE.COM).

Please see [Using Kerberos](#) for details on how to authenticate with Kerberos.

### ADFS

Set the AuthScheme to **ADFS**. Then set the following connection properties:

- User: Set this to the ADFS user.
- Password: Set this to ADFS password for the user.
- SSOLoginURL: Set this to the WS-trust endpoint of the ADFS server.

You also need to set SSOProperties to authenticate to ADFS. Specify the value of the RelyingParty parameter; it is located on the ADFS server for Sharepoint. Below is an example connection string:

```
AuthScheme=ADFS;User=ADFSUserName;Password=ADFSPassword;SSOLoginURL='https://<authority>/adfs/services/trust/2005/usernamemixed';SSO Properties='RelyingParty=urn:sharepoint:sp2016;';
```

## Anonymous Access

Set the AuthScheme to **NONE** along with the URL.

## MSI

If you are running SharePoint on an Azure VM, you can leverage Managed Service Identity (MSI) credentials to connect:

- AuthScheme: Set this to **AzureMSI**.

The MSI credentials are automatically obtained for authentication.

## 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 SharePoint 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 AzureAD App

### When to Create a Custom OAuth App

CData embeds OAuth Application Credentials with CData branding that can be used when connecting via either a Desktop Application or from a Headless Machine. Creating a custom OAuth application is, however, required when using a web application.

You may choose to create your own OAuth Application Credentials when you want to

- control branding of the Authentication Dialog
- control the redirect URI that the application redirects the user to after the user authenticates
- customize the permissions that you are requesting from the user

### Creating a Custom OAuth App

Follow the steps below to obtain OAuth values for your app, the OAuthClientId and OAuthClientSecret.

1. Log in to the [Azure Portal](#).
2. In the left-hand navigation pane, select **Azure Active Directory > App Registrations** and click **Add**.



3. Enter an application name and select **Any Azure AD Directory - Multi Tenant**. Then set the redirect url to `http://localhost:33333`, the adapter's default. Or, set a different port of your choice and set CallbackURL to the exact reply URL you defined.
4. After creating the app, navigate to the "Certificates & Secrets" section, create a client secret for the application, and select a duration.
5. After you save the key, key value is displayed once. Set OAuthClientSecret to the displayed value. Set OAuthClientId to the Application Id.
6. Select **API Permissions** and click **Add**. when selecting permissions, use the Delegated permissions.
7. In the API Permissions section, click on Add a permission and select Sharepoint. And choose the permissions you want your app to have. To view and edit lists you have to select at least the permission, AllSites.Manage.
8. Save your changes.
9. If you have selected to use permissions that require admin consent, you can grant them from the current tenant on the API Permissions page. Otherwise, follow the steps under "OAuth: Admin Consent" in [Establishing a Connection](#).

## Connecting to REST API

SharePoint REST API is supported both on Sharepoint OnPremise and on Sharepoint Online. To connect using the REST API set Schema to REST.

The property SharePointEdition may be used to define the edition of Sharepoint.

### Sharepoint Online

SharePoint Online uses OAuth standard to authenticate. Follow the steps under "Authenticating to SharePoint Online" in [Establishing a Connection](#) for more information.

### Sharepoint OnPremise

Follow the steps under "Authenticating to SharePoint On Premises" in [Establishing a Connection](#) for more information.

## Using Kerberos

This section shows how to use the adapter to authenticate using Kerberos.

## Kerberos

To authenticate to SharePoint using Kerberos, set the following properties:

- AuthScheme: Set this to **NEGOTIATE**.
- KerberosKDC: Set this to the host name or IP Address of your Kerberos KDC machine.
- KerberosRealm: Set this to **the realm of the SharePoint Kerberos principal**. This will be the value after the '@' symbol (for instance, EXAMPLE.COM) of the **principal value** (for instance, MyService/MyHost@EXAMPLE.COM).
- KerberosSPN: Set this to the service and host of the SharePoint Kerberos Principal. This is the value prior to the '@' symbol (for instance, MyService/MyHost) of the principal value (for instance, MyService/MyHost@EXAMPLE.COM).

## Retrieve the Kerberos Ticket

You can use one of the following options to retrieve the required Kerberos ticket.

## MIT Kerberos Credential Cache File

This option enables you to use the MIT Kerberos Ticket Manager or kinit command to get tickets. Note that you do not need to set the User or Password connection properties with this option.

1. Ensure that you have an environment variable created called **KRB5CCNAME**.
2. Set the **KRB5CCNAME** environment variable to a path pointing to your credential cache file (for instance, C:\krb\_cache\krb5cc\_0 or /tmp/krb5cc\_0). This file is created when generating your ticket with MIT Kerberos Ticket Manager.
3. To obtain a ticket, open the MIT Kerberos Ticket Manager application, click **Get Ticket**, enter your principal name and password, then click **OK**. If successful, ticket information appears in Kerberos Ticket Manager and is stored in the credential cache file.
4. Now that you have created the credential cache file, the adapter uses the cache file to obtain the Kerberos ticket to connect to SharePoint.

As an alternative to setting the **KRB5CCNAME** environment variable, you can directly set the file path using the KerberosTicketCache property. When set, the adapter uses the specified cache file to obtain the Kerberos ticket to connect to SharePoint.

## Keytab File

If the **KRB5CCNAME environment variable has not been set**, you can retrieve a Kerberos ticket using a Keytab File. To do so, set the User property to the desired username and set the KerberosKeytabFile property to a file path pointing to the keytab file associated with the user.

## User and Password

If both the **KRB5CCNAME** environment variable and the KerberosKeytabFile property have not been set, you can retrieve a ticket using a user and password combination. To do this, set the User and Password properties to the user/password combination that you use to authenticate with SharePoint.

## Cross-Realm

More complex Kerberos environments may require cross-realm authentication where multiple realms and KDC servers are used (e.g., where one realm/KDC is used for user authentication and another realm/KDC is used for obtaining the service ticket).

In such an environment, set the KerberosRealm and KerberosKDC properties to the values required for user authentication. Also set the KerberosServiceRealm and KerberosServiceKDC properties to the values required to obtain the service ticket.

# Fine-Tuning Data Access

## Fine Tuning the SharePoint Connection

To make it easier to access data in advanced integrations, use the following connection properties to control column name identifiers and other aspects of data access:

- UseDisplayNames: Set this to true to return column names that match field names in the underlying API  
. By default, the adapter uses column names that match the field names defined in SharePoint.
- UseSimpleNames: Set this to true to perform substitutions on special characters in column names that SharePoint allows but that many databases typically do not.

- **ShowPredefinedColumns:** Set this to false to exclude fields derived from fields in the list; for example, Author and CreatedAt. This setting excludes the predefined fields from being returned in *SELECT \** statements and schema discovery.
- **ShowHiddenColumns:** When true, columns marked as hidden in SharePoint will be displayed by the adapter.

## Changelog

### General Changes

Date	Build Number	Change Type	Description
03/01/2023	8460	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>• Added ServerRelativeUrl column in Lists View.</li> </ul>
01/19/2023	8419	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>• Added the links on the Connecting to REST API page under Sharepoint Online and Sharepoint OnPremise sections.</li> </ul>
12/22/2022	8391	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>• Added RenameAttachmentOrDocument and MoveAttachmentOrDocument Stored Procedures in SOAP and REST Schema.</li> </ul>
12/21/2022	8390	SharePoint	<b>Changed</b> <ul style="list-style-type: none"> <li>• Changed the datatype of PertainingToTerm column from Boolean to String in GetValidTerms View.</li> </ul>

12/19/2022	8388	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Add support for the ReadTimeout option to the download stored procedures in the REST schema. ReadTimeout can be used to force a download to fail after a certain time, unlike Timeout which only triggers if the download stalls for that amount of time.</li> </ul>
12/14/2022	8383	General	<b>Changed</b> <ul style="list-style-type: none"> <li>Added the Default column to the sys_procedureparameters table.</li> </ul>
12/08/2022	8377	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added the WriteToFile parameter for CreateSchema stored procedure in SOAP schema. This defaults to true and must be disabled to write the schema to FileStream or FileData.</li> </ul> <b>Removed</b> <ul style="list-style-type: none"> <li>Removed SchemaDirectory parameter from CreateSchema stored procedure in SOAP schema. Instead, the Location connection property path will be used to create the schema.</li> </ul>
10/25/2022	8369	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added UseEntityTypeName property to determine if the table name should be EntityTypeName instead of the title in the REST schema.</li> </ul>
10/25/2022	8333	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added the FileStream input attribute to add outputstream and FileData output attribute to print the response in</li> </ul>

			<p>DownloadAttachment, DownloadDocument stored procedures in SOAP and REST schema.</p> <ul style="list-style-type: none"> <li>Added the FileStream input attribute to add outputstream and FileData output attribute to print the response in CreateSchema stored procedure in SOAP schema.</li> <li>Added the Content input attribute to add inputstream in AddAttachment, UploadDocument stored procedures in SOAP and REST schema.</li> </ul>
09/30/2022	8308	General	<p><b>Changed</b></p> <ul style="list-style-type: none"> <li>Added the IsPath column to the sys_procedureparameters table.</li> </ul>
08/17/2022	8264	General	<p><b>Changed</b></p> <ul style="list-style-type: none"> <li>We now support handling the keyword "COLLATE" as standard function name as well.</li> </ul>
2022/07/29	8245	SharePoint	<p><b>Removed</b></p> <ul style="list-style-type: none"> <li>Removed the Enum value Germany from AzureEnvironments as Microsoft has retired its Germany-based cloud.</li> </ul>
05/24/2022	8179	SharePoint	<p><b>Changed</b></p> <ul style="list-style-type: none"> <li>Changed provider name to Microsoft SharePoint.</li> </ul>
05/19/2022	8174	SharePoint	<p><b>Deprecated</b></p> <ul style="list-style-type: none"> <li>OAuthGrantType has been deprecated. Use the AuthScheme connection property instead.</li> </ul>

03/15/2022	8109	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added a new Id column for the Users view in the REST Schema.</li> </ul> <b>Added</b> <ul style="list-style-type: none"> <li>Added the ItemId column to get RoleAssignments, RoleAssignmentMember, RoleDefinitionBindings.</li> </ul>
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>
08/23/2021	7905	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added the stored procedures CreateFolder, UploadDocument, DeleteDocument, CopyDocument, CheckInDocument, CheckOutDocument, DiscardCheckOutDocument, AddAttachment, DeleteAttachment to the REST Schema.</li> </ul>
08/10/2021	7892	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added new AuthScheme "SharePointOAuth" to support client credentials flow using SharePoint App.</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>

08/05/2021	7887	SharePoint	<b>Removed</b> <ul style="list-style-type: none"> <li>Removed the Permissions view from the REST schema.</li> </ul> <b>Replacements</b> <ul style="list-style-type: none"> <li>Replaced Permissions with RoleAssignments, RoleAssignmentMember and RoleDefinitionBindings views in the REST schema.</li> </ul>
07/29/2021	7880	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added new columns LoginName, Title, IsHiddenInUI for the Users view and Id for the Groups view in the REST schema.</li> </ul>
07/28/2021	7878	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added support for Attachments, Permissions and SubSites views in the REST schema.</li> </ul> <b>Added</b> <ul style="list-style-type: none"> <li>Added support for OAuth Authentication without JWT cert for Client Credentials by when using a SharePoint App.</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</li> </ul>



			<p>match.</p> <ul style="list-style-type: none"> <li>As a replacement for the previous behavior, the relative date/datetime functions in the criteria may have an 'L' appended to them. I.e: 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	<p><b>Added</b></p> <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>
06/16/2021	7837	SharePoint	<p><b>Added</b></p> <ul style="list-style-type: none"> <li>Added support for PingFederate identity provider in Sharepoint Online SOAP schema.</li> </ul>
06/05/2021	7826	SharePoint	<p><b>Added</b></p> <ul style="list-style-type: none"> <li>Added support for the AzureServicePrinciple authentication scheme only using a JWT certs instead of the OAuthClientSecret.</li> <li>Added support to authenticate submitting JWT certs instead of the OAuthClientSecret for the AzureServicePrinciple, OAuth and AzureAD authentication schemes.</li> </ul>
04/23/2021	7785	General	<p><b>Added</b></p> <ul style="list-style-type: none"> <li>Added support for handling client side</li> </ul>

			<p>formulas during insert / update. For example: UPDATE Table SET Col1 = Concat(Col1, " - ", Col2) WHERE Col2 LIKE 'A%'</p>
04/23/2021	7783	General	<p><b>Changed</b></p> <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	<p><b>Added</b></p> <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> <p><b>Changed</b></p> <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>
03/31/2021	7760	SharePoint	<b>Deprecated</b> <ul style="list-style-type: none"> <li>The UseSSO connection property is deprecated. You should select the preferred SSO scheme directly from the AuthScheme property instead.</li> <li>The URNAddress is deprecated. This property is used inside the SSOProperties and it should be specified when authenticating to ADFS on Sharepoint On-Premise. Instead RelyingParty should be used as in other drivers.</li> </ul>
11/03/2020	7612	SharePoint	<b>Added</b> <ul style="list-style-type: none"> <li>Added support for retrieving information regarding the current logged in user via the GetCurrentUser stored procedure.</li> </ul>

## Advanced Features

This section details a selection of advanced features of the SharePoint 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 SharePoint 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 SharePoint 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 Calendar 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.

### Client SSL Certificates

The SharePoint adapter also supports setting client certificates. Set the following to connect using a client certificate.

- [SSLClientCert](#): The name of the certificate store for the client certificate.
- [SSLClientCertType](#): The type of key store containing the TLS/SSL client certificate.
- [SSLClientCertPassword](#): The password for the TLS/SSL client certificate.
- [SSLClientCertSubject](#): The subject of the TLS/SSL client certificate.

## 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 SharePoint 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 SharePoint Query Evaluation component examines SQL queries and returns information indicating what parts of the query the adapter is not capable of executing natively.

The SharePoint 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 verbosities, 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 SharePoint Adapter supports several operations on data, including querying, deleting, modifying, and inserting.

### SELECT Statements

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

See [SOAP Data Model](#) for information on the capabilities of the SharePoint 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 SharePoint adapter:

```
SELECT {  
  [ TOP <numeric_literal> ]
```

```

{
  *
  | {
    <expression> [ [ AS ] <column_reference> ]
    | { <table_name> | <correlation_name> } .*
    } [ , ... ]
  }
[ INTO csv:// [ filename= ] <file_path> [ ;delimiter=tab ] ]
{
  FROM <table_reference> [ [ AS ] <identifier> ]
}
[ WHERE <search_condition> ]
[
  ORDER BY
  <column_reference> [ ASC | DESC ] [ NULLS FIRST | NULLS LAST ]
]
[
  LIMIT <expression>
]
} | SCOPE_IDENTITY()
<expression> ::=
  | <column_reference>
  | @ <parameter>
  | ?
  | COUNT( * | { [ DISTINCT ] <expression> } )
  | { AVG | MAX | MIN | SUM | COUNT } ( <expression> )
  | NULLIF ( <expression> , <expression> )
  | COALESCE ( <expression> , ... )
  | CASE <expression>
    WHEN { <expression> | <search_condition> } THEN { <expression> |
NULL } [ ... ]
    [ ELSE { <expression> | NULL } ]
    END
  | <literal>
  | <sql_function>
<search_condition> ::=
  {
    <expression> { = | != | <> | > | < | >= | <= | BEGINSWITH |
CONTAINS | IN | IS NULL | IS NOT NULL | AND | OR } [ <expression> ]
  } [ { AND | OR } ... ]

```

## Examples

1. Return all columns:

```
SELECT * FROM Calendar
```

2. Rename a column:

```
SELECT "Location" AS MY_Location FROM Calendar
```

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

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

4. Search data:

```
SELECT * FROM Calendar WHERE Location <> 'Chapel Hill'
```

5. The SharePoint APIs support the following operators in the WHERE clause: =, !=, <>, >, <, >=, <=, BEGINSWITH, CONTAINS, IN, IS NULL, IS NOT NULL, AND, OR.

```
SELECT * FROM Calendar WHERE Location <> 'Chapel Hill';
```

6. Sort a result set in ascending order:

```
SELECT Id, Location FROM Calendar ORDER BY Location ASC
```

## SELECT INTO Statements

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

### Data Export with an SQL Query

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

```
boolean ret = stat.execute("SELECT Id, Location INTO  
"csv://c:/Calendar.txt" FROM "Calendar" WHERE Location <> 'Chapel  
Hill'");  
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 "Calendar" IN
'csv://filename=c:/Calendar.csv;delimiter=tab' FROM "Calendar" WHERE
Location <> 'Chapel Hill'");
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 Calendar (Location) VALUES (?)";
PreparedStatement pstmt = connection.prepareStatement
(cmd,Statement.RETURN_GENERATED_KEYS);
pstmt.setString(1, "U.S.A.");
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 Calendar SET Location='U.S.A.' WHERE Id = ?";
PreparedStatement pstmt = connection.prepareStatement(cmd);
pstmt.setString(1, "1");
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:sharepoint:User=MyUserAccount;Password=MyPassword;Auth  
Scheme=NTLM;URL=http://sharepointserver/mysite;");  
String cmd = "DELETE FROM Calendar WHERE Id = ?";  
PreparedStatement pstmt = connection.prepareStatement(cmd);  
pstmt.setString(1, "1");  
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 Syntax

```
"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 \(Transact-SQL\)](#)

## SOAP Data Model

The SharePoint Adapter models SharePoint entities in relational Tables, Views, and Stored Procedures. The table definitions are dynamically obtained based on your SharePoint site. Any changes you make, such as adding a custom field or changing a field's data type, are automatically reflected when you connect.

### Customizing the Data Model

The adapter sets defaults to facilitate the maximum number of integrations; however, the following connection properties allow a greater granularity of customization useful in advanced integrations:

- CalculatedDataType: The data type to be used for calculated fields.
- CreateIDColumns: Indicates whether or not to create supplemental ID columns for SharePoint columns that use values from information stored in other Lists.
- FolderOption: An option to determine how to display folders in results. Enter either FilesOnly, FilesAndFolders, Recursive, or RecursiveAll.
- PseudoColumns: Indicates whether or not to report pseudo columns as columns in the table metadata.

### Tables

[Tables](#) describes the available tables.

The adapter can expose custom lists from SharePoint that are not mentioned in the [Tables](#). The data model illustrates a sample of what your SharePoint site might look like. The

actual data model will be obtained dynamically based on your user credentials and SharePoint site.

## Views

Typically, entities that cannot be modified are represented as [Views](#), or read-only tables. You can also access custom views of a list as relational views.

To get data from a custom view of a list, you can set the ViewID pseudo column in the WHERE clause.

```
SELECT * FROM ListName WHERE ViewID='ID of the view'
```

You can get the ID of the view from the [Views](#) list. You must specify the List pseudo column to get a list of views for that list. For instance:

```
SELECT * FROM Views WHERE List ='ListName'
```

## Stored Procedures

[Stored Procedures](#) are function-like interfaces to the data source. They surface additional capabilities of the SharePoint API such as searching, updating, and modifying information.

## Tables

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

Generally, querying SharePoint 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.

## SharePoint Adapter Tables

Name	Description
------	-------------

<a href="#">Attachments</a>	Read or delete Attachments for the specified item on the specified list.
<a href="#">Groups</a>	Create, update, delete, and query Groups from SharePoint.
<a href="#">Roles</a>	Create, update, delete, and query Roles from SharePoint.
<a href="#">Users</a>	Update, delete, and query Users from SharePoint.
<a href="#">Views</a>	Create, update, delete, and query the available lists in SharePoint.

## Attachments

Read or delete Attachments for the specified item on the specified list.

### Table Specific Information

#### Select

List and ItemId are required to return Attachments.

#### Insert

Call the AddAttachments stored procedure to add new attachments to a list item.

#### Columns

Name	Type	ReadOnly	Description
Url [KEY]	<i>String</i>	True	Description of the term set.
List	<i>String</i>	True	The internal name of the list to retrieve attachments from.

ListDisplayName	<i>String</i>	True	The display name of the list to retrieve attachments from.
ItemID	<i>String</i>	True	The ID of the item on the list to retrieve attachments from.
Name	<i>String</i>	True	The name of the attachment on the item.

## Groups

Create, update, delete, and query Groups from SharePoint.

### Table Specific Information

#### Insert

The Name, DefaultLogin, and OwnerLogin columns are required to insert to this table.

To use the UserName pseudo column, you must set the value to the LoginName of the user. You can obtain the LoginName by querying the Users table.

#### Columns

<b>Name</b>	<b>Type</b>	<b>ReadOnly</b>	<b>Description</b>
Name# [KEY]	<i>String</i>	False	The name of the group.
Description#	<i>String</i>	False	A description of the group.
OwnerLogin#	<i>String</i>	False	The user name of the owner of the group. This value should be in the format DOMAIN\\username.

OwnerType#	<i>String</i>	False	The type of owner. User or group.
DefaultLogin#	<i>String</i>	False	The user name of the default user for the group. This value should be in the format DOMAIN\\username.

## Pseudo-Columns

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

Name	Type	Description
UserName	<i>String</i>	The logical name of the user to return groups for. Specify this value on the SELECT statement to return only groups the specified User is in.

## Roles

Create, update, delete, and query Roles from SharePoint.

### Table Specific Information

#### Select

To use the UserName pseudo column, you must set the value to the LoginName of the user. You can obtain the LoginName by querying the Users table.

#### Insert

To insert a Role, at least the Name is required:



```
INSERT INTO Roles (Name) VALUES ('My Role')
```

## Columns

Name	Type	ReadOnly	Description
Name# [KEY]	<i>String</i>	False	The name of the role.
Description#	<i>String</i>	False	A description of the role.
Permissions#	<i>Long</i>	False	A long representing the permissions for the role.
RoleType	<i>String</i>	True	The type of role.
IsHidden	<i>Boolean</i>	True	A boolean indicating if the role is hidden.

## Pseudo-Columns

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

Name	Type	Description
UserName	<i>String</i>	The login name of the user to return roles for. Specify this value on a SELECT statement to return only roles assigned to the specified user.

---

GroupName	<i>String</i>	The name of the group to return roles for. Specify this value on a SELECT statement to return only roles assigned to the specified group.
-----------	---------------	---

---

## Users

Update, delete, and query Users from SharePoint.

### Table Specific Information

#### Select

Retrieve all users created for the SharePoint Account:

```
SELECT * FROM Users
```

You can retrieve Users that belong to a specific Group. In this case specify the Group Name

```
SELECT * FROM Users WHERE [Group] = "GroupName"
```

Or you can retrieve Users that have a specific Role. In this case specify the Role Name

```
SELECT * FROM Users WHERE [Role] = "RoleName"
```

#### Columns

---

Name	Type	ReadOnly	Description
ID	<i>String</i>	True	The ID of the user.
LoginName# [KEY]	<i>String</i>	False	The login name of the user.

---



---

Name#	<i>String</i>	False	The name of the user.
Email#	<i>String</i>	False	The email address of the user.
IsInDomainGroup	<i>Boolean</i>	True	A boolean indicating if the user is in the domain group.
IsSiteAdmin	<i>Boolean</i>	True	A boolean indicating if the user is a site admin.
Notes#	<i>String</i>	False	Optional notes concerning the user.
SecurityId	<i>String</i>	True	The security Id (SID) for the user.

## 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.

<b>Name</b>	<b>Type</b>	<b>Description</b>
Group	<i>String</i>	The group you are adding a user to and selecting or deleting the user from. This is an input-only value and either Group or Role must be specified for inserts and selects, but may be optionally specified for deletions.
Role	<i>String</i>	The role you are adding a user to and selecting or deleting the user from. This is an input-only value and either Group or Role must be specified for inserts and selects, but may be optionally specified for deletions.

## Views

Create, update, delete, and query the available lists in SharePoint.

## Table Specific Information

Views is a special table. It may be used to get, update, insert, and delete views from a specified List.

### Select

In order to return results from Views, either the Id or List must be specified in the SELECT statement. For example:

```
SELECT * FROM Views WHERE List='MyListName'
```

### Insert

The List, Name, Type, and Fields columns are required to insert to this table.

### Columns

Name	Type	ReadOnly	Description
ID [KEY]	String	True	The Id of the view.
List	String	True	The list the view is associated with. A list must be specified when performing SELECT statements if the Id is not specified.
ViewID	String	True	The Id of the view. May only be unique for the specific list.
Name	String	False	The name of the view.

Type	<i>String</i>	False	<p>The type of view. This must have a value on inserts and updates.</p> <p>The allowed values are <i>CALENDAR</i>, <i>GRID</i>, <i>HTML</i>.</p> <p>The default value is <i>HTML</i>.</p>
Fields	<i>String</i>	False	A comma separated list of the fields associated with the view. This is space-sensitive.
IsDefault	<i>Boolean</i>	False	A boolean indicating if the view is the default view for the list.
Query	<i>String</i>	False	A query for the view.

## 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.

## SharePoint Adapter Views

Name	Description
<a href="#">FileVersions</a>	Lists the versions of files available on SharePoint.
<a href="#">GetValidTerms</a>	Gets a list of valid terms for the specified column on the specified table.
<a href="#">Lists</a>	Lists the available lists in SharePoint.

<b>Permissions</b>	The permissions for a site or list. Note: If ItemID is empty, the ObjectType should be set to List or web (an ObjectName must be specified when the ObjectType is list). If not, the ObjectName must be specified along with the ItemID.
<b>Subsites</b>	This lists the available subsites.

## FileVersions

Lists the versions of files available on SharePoint.

### View-Specific Information

Library and File must be specified to return results from this view.

### Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The ID of the version.
Comments	<i>String</i>	Comments about the particular version.
CreateBy	<i>String</i>	The username of the SharePoint user who modified this version of the file.
Date	<i>Datetime</i>	When the file was modified.
Size	<i>String</i>	The size of this version of the file.
Url	<i>String</i>	The URL to this version of the file.

Library	<i>String</i>	The library name on SharePoint you are listing versions from. A library must be specified to retrieve the versions for a file.  The default value is Shared Documents.
File	<i>String</i>	The name of the file on SharePoint to list versions for. A file must be specified to retrieve the versions for a file.

## GetValidTerms

Gets a list of valid terms for the specified column on the specified table.

### Table Specific Information

GetValidTerms is a special view. It may be used to get valid terms for a Taxonomy or Managed Metadata column of a given list. To use the view, supply both the name of the table and the column for which you are looking to get valid terms. For example:

```
SELECT * FROM GetValidTerms WHERE List='MyListName' AND
ColumnName='MyManagedMetadataColumn'
```

### Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The identifier of the term.
TermLabelValue	<i>String</i>	The label of the term.

Description	<i>String</i>	Description of the term set.
NameInRequestedLang	<i>String</i>	The name of the term set in the language requested by the client.
IsOpen	<i>Boolean</i>	Boolean indicating if the term set is open.
Deprecated	<i>Boolean</i>	Boolean indicating if the term is deprecated.
InternalId	<i>String</i>	Internal identifier for the term.
TermSetContact	<i>String</i>	Term set contact.
ContainerDesc	<i>String</i>	Container node for the description.
SingleTermLabelDesc	<i>String</i>	This fully describes a single term label.
IsDefaultLabel	<i>Boolean</i>	True if the term label is the default term label.
BelongsTo	<i>String</i>	This item describes a term set to which a term belongs.
IsTaggingAvailable	<i>Boolean</i>	If the term set is available for tagging, this value is true.
TermPath	<i>String</i>	Term path of the term with term labels.
TermpathoftermwithIds	<i>String</i>	Term path of term with identifiers.
ChildTerms	<i>String</i>	A string value that indicates a custom sort order for



		the child terms of the term identified by PertainingToTerm.
HasChildTerms	<i>Boolean</i>	True if the term has child terms.
PertainingToTerm	<i>String</i>	Identifier of the term that this term set information is pertaining to.

## 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.

<b>Name</b>	<b>Type</b>	<b>Description</b>
List	<i>String</i>	The name of the list to get valid terms for.
ColumnName	<i>String</i>	The name of the column to get valid terms for.
LocaleId	<i>String</i>	The locale Id for the term. Defaults to 1033.

## Lists

Lists the available lists in SharePoint.

### Table Specific Information

Lists can be used to list the tables in SharePoint. This will only return actual lists in SharePoint and not any special tables associated with the adapter.

The following columns can be used in the WHERE clause: Title and BaseTemplate.

## Columns

Name	Type	Description
ID [KEY]	<i>String</i>	The Id of the list.
Title	<i>String</i>	The title of the list. This column may be used in the WHERE clause and may be used with a wild card (*) character.
Description	<i>String</i>	A description for the list.
BaseTemplate	<i>String</i>	Indicates the type of template used to create the list. This column may be used in the WHERE clause.
Version	<i>Double</i>	The version of the list.
Url	<i>String</i>	The default URL of the list.
EmailAlias	<i>String</i>	The email alias of the list.
ImageUrl	<i>String</i>	The image URL of the list.
ItemCount	<i>Integer</i>	The number of items in the list.
Item_Deleted	<i>Datetime</i>	The last time an item was deleted from this list.
Item_Modified	<i>Datetime</i>	The last time an item was modified from this list.

SendToUrl	<i>String</i>	The send-to URL of the list.
Created	<i>Datetime</i>	The time when the list was created.
AllowDeletion	<i>String</i>	Whether items can be deleted.
AllowMultiResponses	<i>Boolean</i>	Boolean indicating if multiple responses are enabled for the survey.
Direction	<i>String</i>	A string that contains LTR if the reading order is left-to-right, RTL if it is right-to-left, or None.
EnableAssignedToEmail	<i>Boolean</i>	Boolean indicating if assigned-to emails are enabled. Only applies to issues lists.
EnableAttachments	<i>Boolean</i>	Boolean indicating if attachments may be added to items in the list. Does not apply to document libraries.
EnableModeration	<i>Boolean</i>	Boolean indicating if content approval is enabled for the list.
EnableVersioning	<i>Boolean</i>	Boolean indicating if versioning is enabled for the list.
Hidden	<i>Boolean</i>	Boolean indicating if the list is hidden so that it does not appear on the Documents and Lists page, Quick Launch bar, Modify Site Content page, or Add Column page as an option for lookup fields.
MultipleDataList	<i>Boolean</i>	Boolean indicating if a meeting-workspace site contains data for multiple meeting instances within the site.
Ordered	<i>Boolean</i>	Boolean indicating if items in the list can be sorted on the Edit View page.

Showuser	<i>Boolean</i>	Boolean indicating if the names of users are shown in the results of the survey.
----------	----------------	--

## Permissions

The permissions for a site or list. Note: If ItemID is empty, the ObjectType should be set to List or web (an ObjectName must be specified when the ObjectType is list). If not, the ObjectName must be specified along with the ItemID.

## Columns

Name	Type	Description
MemberID [KEY]	<i>String</i>	The ID of the permission.
Mask	<i>Long</i>	A 32-bit integer in 0x00000000 format that represents a Microsoft.SharePoint.SPRights value and defines the permission. Use the pipe symbol (' ') in C# or Or in Microsoft Visual Basic to delimit values when creating a custom permission mask that combines permissions.
MemberIsUser	<i>Bool</i>	Indicate whether it is the permission for user.
MemberGlobal	<i>Bool</i>	Indicate whether it is the permission for group.
RoleName	<i>String</i>	A string that contains the name of the site group, the name of the cross-site group, or the user name (DOMAIN\User_Alias) of the user to whom the permission applies.

## Pseudo-Columns

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

Name	Type	Description
ObjectName	<i>String</i>	A string that contains the name of the list or site.
ObjectType	<i>String</i>	A string that specifies either List or Web.
ItemID	<i>String</i>	ID of the item.

## Subsites

This lists the available subsites.

## Columns

Name	Type	Description
Title	<i>String</i>	The name of the subsite.
Url	<i>String</i>	The url of the subsite.

## Stored Procedures

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

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

## SharePoint Adapter Stored Procedures

Name	Description
<a href="#">AddAttachment</a>	Add an Attachment to a SharePoint List item.
<a href="#">AddList</a>	Creates a list on a SharePoint site.
<a href="#">AddListColumn</a>	Adds a new column to the specified list.
<a href="#">AddUserToGroup</a>	Add the user to specified group.
<a href="#">AddUserToRole</a>	Add the user to specified role.
<a href="#">CheckInDocument</a>	Checks in a document to SharePoint and releases the lock on the document.
<a href="#">CheckOutDocument</a>	Checks out a document from SharePoint.
<a href="#">CopyDocument</a>	Copies a document from the SharePoint library.
<a href="#">CreateFolder</a>	Adds a folder to a document library on a SharePoint site.
<a href="#">CreateSchema</a>	Creates the schema file for the specified SharePoint list. The schema file may be customized manually to exclude unwanted columns or include additional information about columns.
<a href="#">DeleteAttachment</a>	Delete an attachment from a SharePoint list item.
<a href="#">DeleteDocument</a>	Delete a document on the SharePoint library.
<a href="#">DeleteList</a>	Permanently deletes a list from a SharePoint site.

<a href="#">DeleteListColumn</a>	Deletes a column from the specified list.
<a href="#">DeleteUserFromGroup</a>	Delete the user from specified group.
<a href="#">DeleteUserFromRole</a>	Delete the user from specified role.
<a href="#">DiscardCheckOutDocument</a>	Discards a check out on a document in SharePoint. This does not check a new file into SharePoint. It only releases the lock on the document.
<a href="#">DownloadAttachment</a>	Download a document from the SharePoint list.
<a href="#">DownloadDocument</a>	Download a document from the SharePoint library.
<a href="#">MoveAttachmentOrDocument</a>	Moves a document or attachment from source folder to destination folder.
<a href="#">RenameAttachmentOrDocument</a>	Renames a document or attachment from source folder to destination folder.
<a href="#">UpdateList</a>	Updates a list on a SharePoint site.
<a href="#">UpdateListColumn</a>	Updates a column to the specified list.
<a href="#">UploadDocument</a>	Upload a document to the SharePoint library.

## AddAttachment

Add an Attachment to a SharePoint List item.

### Input

Name	Type	Accepts Input Streams	Description
File	<i>String</i>	<i>False</i>	The path of the local file to be added.

List	<i>String</i>	<i>False</i>	The name of the List on the SharePoint server.
ItemID	<i>String</i>	<i>False</i>	The ID of the item on the List to add attachments for.
Content	<i>String</i>	<i>True</i>	The content as InputStream to be uploaded when File is not specified.
FileName	<i>String</i>	<i>False</i>	Name of the file to be uploaded. This will be used if content is not null. For example: test.csv

## Result Set Columns

Name	Type	Description
URL	<i>String</i>	The URL of the newly created item.
Result	<i>String</i>	Boolean value indicating whether the stored procedure was successful.

## AddList

Creates a list on a SharePoint site.

### Input

Name	Type	Description
Name	<i>String</i>	The name of the list on the SharePoint server.



Template	<i>String</i>	<p>The name of the template to use for the list creation.</p> <p>The allowed values are <i>GenericList</i>, <i>DocumentLibrary</i>, <i>Survey</i>, <i>Links</i>, <i>Announcements</i>, <i>Contacts</i>, <i>Events</i>, <i>Tasks</i>, <i>DiscussionBoard</i>, <i>PictureLibrary</i>, <i>DataSources</i>, <i>WebTemplateCatalog</i>, <i>UserInfo</i>, <i>WebPartCatalog</i>, <i>ListTemplateCatalog</i>, <i>XMLForm</i>, <i>MasterPageCatalog</i>, <i>NoCodeWorkflows</i>, <i>WorkflowProcess</i>, <i>WebPageLibrary</i>, <i>CustomGrid</i>, <i>DataConnectionLibrary</i>, <i>WorkflowHistory</i>, <i>GanttTasks</i>, <i>Meetings</i>, <i>Agenda</i>, <i>MeetingUser</i>, <i>Decision</i>, <i>MeetingObjective</i>, <i>TextBox</i>, <i>ThingsToBring</i>, <i>HomePageLibrary</i>, <i>Posts</i>, <i>Comments</i>, <i>Categories</i>, <i>IssueTracking</i>, <i>AdminTasks</i>.</p> <p>The default value is <i>GenericList</i>.</p>
Description	<i>String</i>	The description of the list to add.
Columns	<i>String</i>	The definition of the columns to add, support json/xml/tempTable

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## AddListColumn

Adds a new column to the specified list.

## Input

Name	Type	Description
List	<i>String</i>	The name of the list on the SharePoint server.
ColumnName	<i>String</i>	The name of the column to add.
DisplayName	<i>String</i>	The display name of the column to add.
DefaultValue	<i>String</i>	The default value of the column to add.
ColumnType	<i>String</i>	The data type of the column to add. The valid options are defined by the FieldTypes available in the SharePoint API: <a href="https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-csom/ee540543(v=office.15)">https://docs.microsoft.com/en-us/previous-versions/office/sharepoint-csom/ee540543(v=office.15)</a> .
MaxLength	<i>String</i>	The values' maximum length of the column to add.
PrimaryKey	<i>String</i>	Boolean value indicating whether or not the column should be primary key.
ReadOnly	<i>String</i>	Boolean value indicating whether or not the column is read only.
Required	<i>String</i>	Boolean value indicating whether or not the column is required.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## AddUserToGroup

Add the user to specified group.

### Input

Name	Type	Description
LoginName	<i>String</i>	The login name of the user.
Group	<i>String</i>	The group you are adding a user to and selecting or deleting the user from. This is an input-only value and either Group or Role must be specified for inserts and selects, but may be optionally specified for deletions.

### Result Set Columns

Name	Type	Description
Result	<i>Boolean</i>	Boolean value indicating whether the stored procedure was successful.

## AddUserToRole

Add the user to specified role.

### Input

Name	Type	Description
LoginName	<i>String</i>	The login name of the user.
Role	<i>String</i>	The role you are adding a user to and selecting or deleting the user from. This is an input-only value and either Group or Role must be specified for inserts and selects, but may be optionally specified for deletions.

## Result Set Columns

Name	Type	Description
Result	<i>Boolean</i>	Boolean value indicating whether the stored procedure was successful.

## CheckInDocument

Checks in a document to SharePoint and releases the lock on the document.

### Input

Name	Type	Description
File	<i>String</i>	The path of the file you are using to overwrite the document on SharePoint with. For example: C:\myfolder\myfile.txt.
Library	<i>String</i>	The name of the library on the SharePoint server. For example: Shared Documents.
Comment	<i>String</i>	A comment to leave when checking the file in.

RemoteFile	<i>String</i>	The path of the file on the server. This can be the full URL or simply the file name.
------------	---------------	---

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## CheckOutDocument

Checks out a document from SharePoint.

### Input

Name	Type	Description
Library	<i>String</i>	The name of the library on the SharePoint server.
RemoteFile	<i>String</i>	The path of the file on the server. This can be the full URL or simply the file name.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## CopyDocument

Copies a document from the SharePoint library.

**Note:** This procedure makes use of **indexed parameters**. These input parameters are denoted with a '#' character at the end of their names.

Indexed parameters facilitate providing multiple instances a single parameter as inputs for the procedure.

Suppose there is an input parameter named Param#. Input multiple instances of an indexed parameter like this:

```
EXEC ProcedureName Param#1 = "value1", Param#2 = "value2", Param#3 = "value3"
```

## Input

Name	Type	Description
DocumentName	<i>String</i>	The name of the document in the document library to be copied.
DocumentLibrary	<i>String</i>	The name of the document library the document is currently stored on.
NewDocumentLibrary	<i>String</i>	The name of the document library the document is being copied to.
NewDocumentName	<i>String</i>	The new name of the document once it has been copied. If left blank, this will be the same as the DocumentName.
MetadataName#	<i>String</i>	The name of a metadata field to be set for the document.
ReturnID	<i>String</i>	The return ID of the document.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.
_dlc_DocId	<i>String</i>	The document ID.
_dlc_DocIdUrl	<i>String</i>	The URL of the document ID.
Vti_author	<i>String</i>	The creator of the document.
Vti_etag	<i>String</i>	The e-tag of the document.
ID	<i>String</i>	The ID of the document.
FileRef	<i>String</i>	The file reference of the document.

## CreateFolder

Adds a folder to a document library on a SharePoint site.

### Input

Name	Type	Description
Library	<i>String</i>	The name of the library on the SharePoint server.

---

Name	<i>String</i>	Name of the folder to which the document is to be added.
------	---------------	--

---

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the stored procedure was successful.

---

## CreateSchema

Creates the schema file for the specified SharePoint list. The schema file may be customized manually to exclude unwanted columns or include additional information about columns.

### Input

Name	Type	Accepts Output Streams	Description
Table	<i>String</i>	<i>False</i>	The name of the table for which to create a schema.
TableDescription	<i>String</i>	<i>False</i>	An optional description of the table.
FileStream	<i>String</i>	<i>True</i>	OutputStream to write the created schema

---



WriteToFile	<i>String</i>	<i>False</i>	Whether to write to an output file or not. Defaults to true, must be set to false to write to FileStream or FileData.
-------------	---------------	--------------	---

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Returns Success or Failure.
SchemaFile	<i>String</i>	The generated schema file.
Columns	<i>String</i>	The number of columns found.
FileData	<i>String</i>	The generated schema encoded in base64. Only returned if WriteToFile set to false and FileStream is not set.

## DeleteAttachment

Delete an attachment from a SharePoint list item.

### Input

Name	Type	Description
URL	<i>String</i>	Full URL to attachment to be deleted.
List	<i>String</i>	The name of the List on the SharePoint server.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the stored procedure was successful.

## DeleteDocument

Delete a document on the SharePoint library.

### Input

Name	Type	Description
Library	<i>String</i>	The name of the library on the SharePoint server.
Path	<i>String</i>	The path of the file (or folder) to remove from the document library.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the stored procedure was successful.

## DeleteList

Permanently deletes a list from a SharePoint site.

### Input

Name	Type	Description
List	<i>String</i>	The name of the list on the SharePoint server.

### Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## DeleteListColumn

Deletes a column from the specified list.

### Input

Name	Type	Description
List	<i>String</i>	The name of the list on the SharePoint server.
ColumnName	<i>String</i>	The name of the column to delete.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## DeleteUserFromGroup

Delete the user from specified group.

### Input

Name	Type	Description
LoginName	<i>String</i>	The login name of the user.
Group	<i>String</i>	The group you are adding a user to and selecting or deleting the user from. This is an input-only value and either Group or Role must be specified for inserts and selects, but may be optionally specified for deletions.

## Result Set Columns

Name	Type	Description
Result	<i>Boolean</i>	Boolean value indicating whether the stored procedure was successful.

## DeleteUserFromRole

Delete the user from specified role.

### Input

Name	Type	Description
LoginName	<i>String</i>	The login name of the user.
Role	<i>String</i>	The role you are adding a user to and selecting or deleting the user from. This is an input-only value and either Group or Role must be specified for inserts and selects, but may be optionally specified for deletions.

### Result Set Columns

Name	Type	Description
Result	<i>Boolean</i>	Boolean value indicating whether the stored procedure was successful.

## DiscardCheckOutDocument

Discards a check out on a document in SharePoint. This does not check a new file into SharePoint. It only releases the lock on the document.

### Input

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

Library	<i>String</i>	The name of the library on the SharePoint server.
RemoteFile	<i>String</i>	The name of the file being checked out.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## DownloadAttachment

Download a document from the SharePoint list.

### Input

Name	Type	Accepts Output Streams	Description
File	<i>String</i>	<i>False</i>	The path of the file to be saved.
RemoteFile	<i>String</i>	<i>False</i>	The path of the file on the server. This can be the full URL or simply the file name. If you use the name of the file, the latest version will be downloaded.
FileStream	<i>String</i>	<i>True</i>	OutputStream to write the downloaded attachment. Only returned if File is not set.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.
FileData	<i>String</i>	The downloaded BASE64 encoded file content. Only returned if File and FileStream is not set.

## DownloadDocument

Download a document from the SharePoint library.

### Input

Name	Type	Accepts Output Streams	Description
File	<i>String</i>	<i>False</i>	The path of the file to be saved.
Library	<i>String</i>	<i>False</i>	The name of the library on the SharePoint server.
RemoteFile	<i>String</i>	<i>False</i>	The path of the file on the server. This can be the full URL or simply the file name. If you use the name of the file, the latest version will be downloaded.
FileStream	<i>String</i>	<i>True</i>	OutputStream to write the downloaded document. Only returned if File is not set.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.
FileData	<i>String</i>	The downloaded file content. Only returned if File and FileStream is not set.

## MoveAttachmentOrDocument

Moves a document or attachment from source folder to destination folder.

The MoveAttachmentOrDocument stored procedure requires List, SourceFileURL and DestinationFolderURL parameters to move an attachment or document.

For Example:

```
EXEC MoveAttachmentOrDocument List = 'Test134', SourceFileURL = '/Shared Documents/Dummy.txt', DestinationFolderURL = '/Shared Documents/MySite'
```

## Input

Name	Type	Description
List	<i>String</i>	Name of the list from which you want to move the document or attachment.
SourceFileURL	<i>String</i>	<p>URL of the source file, relative to the base <u>Url</u> supplied in the adapter's connection properties.</p> <p>For example:  <b>Root Directory file:</b> /Shared Documents/filename.txt  <b>Sub-directory file:</b> /Shared Documents/MyFolder/filename.txt</p>



If you set the Url connection property to a site collection, the relative URL will correspond with a path on the **base site**.

If the Url connection property points to a specific site, the relative URL will be relative to the site supplied in the Url.

DestinationFolderURL      *String*

URL of the destination folder where you want to move the file, relative to the base Url supplied in the adapter's connection properties.

For example:

**Root Directory:** /Shared Documents/

**Sub-directory:** /Shared Documents/MyFolder/

If you set the Url connection property to a site collection, the relative URL will correspond with a path on the **base site**.

If the Url connection property points to a specific site, the relative URL will be relative to the site supplied in the Url.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Value indicating whether the operation was successful.

## RenameAttachmentOrDocument

Renames a document or attachment from source folder to destination folder.

The RenameAttachmentOrDocument stored procedure requires List, SourceFileURL and NewFileName parameters to rename an attachment or document.

For Example:

```
EXEC RenameAttachmentOrDocument List = 'Test134', SourceFileURL =
'/Shared Documents/Dummy.txt', NewFileName = 'Dummy1.txt'
```

## Input

Name	Type	Description
List	<i>String</i>	Name of the list from which you want to move the document or attachment.
SourceFileURL	<i>String</i>	<p>URL of the source file, relative to the base <u>Url</u> supplied in the adapter's connection properties.</p> <p>For example:  <b>Root Directory file:</b> /Shared Documents/filename.txt  <b>Sub-directory file:</b> /Shared Documents/MyFolder/filename.txt            If you set the <u>Url</u> connection property to a site collection, the relative URL will correspond with a path on the <b>base site</b>.            If the <u>Url</u> connection property points to a specific site, the relative URL will be relative to the site supplied in the <u>Url</u>.</p>
NewFileName	<i>String</i>	New name of the file with extension.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Value indicating whether the operation was successful.

## UpdateList

Updates a list on a SharePoint site.

## Input

Name	Type	Description
List	<i>String</i>	The name of the list on the SharePoint server.
AllowMultiResponses	<i>String</i>	Set to True to allow multiple responses to the survey.
Description	<i>String</i>	A string that contains the description for the list.
Direction	<i>String</i>	A string that contains LTR if the reading order is left-to-right, RTL if it is right-to-left, or None.
EnableAssignedToEmail	<i>String</i>	Set to True to enable assigned-to e-mail for the issues list.
EnableAttachments	<i>String</i>	Set to True to enable attachments to items in the list. Does not apply to document libraries.
EnableModeration	<i>String</i>	Set to True to enable Content Approval for the list.
EnableVersioning	<i>String</i>	Set to True to enable versioning for the list.
Hidden	<i>String</i>	Set to True to hide the list so that it does not appear on the Documents and Lists page, Quick Launch bar, Modify Site Content page, or Add Column page as an option for lookup fields.
MultipleDataList	<i>String</i>	Set to True to specify that the list in a Meeting Workspace site contains data for multiple meeting instances within the site.
Ordered	<i>String</i>	Set to True to specify that the option to allow users to reorder items in the list is available on the Edit

View page for the list.		
ShowUser	<i>String</i>	Set to True to specify that names of users are shown in the results of the survey.
Title	<i>String</i>	A string that contains the title of the list.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

## UpdateListColumn

Updates a column to the specified list.

### Input

Name	Type	Description
List	<i>String</i>	The name of the list on the SharePoint server.
ColumnName	<i>String</i>	The name of the column to update.
DisplayName	<i>String</i>	The updated value of the display name.
DefaultValue	<i>String</i>	The updated default value of the specified column.

---

ColumnType	<i>String</i>	The updated data type of the specified column.
MaxLength	<i>String</i>	The updated maximum length of the specified column.
PrimaryKey	<i>String</i>	Use this to make or not the existing column, primary key.
ReadOnly	<i>String</i>	Use this to make or not the existing column, readonly.
Required	<i>String</i>	Use this to make or not the existing column, required.

---

## Result Set Columns

---

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.

---

## UploadDocument

Upload a document to the SharePoint library.

**Note:** This procedure makes use of **indexed parameters**. These input parameters are denoted with a '#' character at the end of their names.

Indexed parameters facilitate providing multiple instances a single parameter as inputs for the procedure.

Suppose there is an input parameter named Param#. Input multiple instances of an indexed parameter like this:

```
EXEC ProcedureName Param#1 = "value1", Param#2 = "value2", Param#3 = "value3"
```

## Input

Name	Type	Accepts Input Streams	Description
File	<i>String</i>	<i>False</i>	The path of the file to be added.
FileContent	<i>String</i>	<i>False</i>	Base64 encoded content of the file to be added. If specified, the value of 'File' input will be ignored.
Library	<i>String</i>	<i>False</i>	<p>The URL of the directory you would like to upload file(s) to, relative to the base <u>Url</u> supplied in the adapter's connection properties.</p> <p>For example:  <b>Root directory:</b> <i>Shared Documents</i></p> <p>If you set the <u>Url</u> connection property to a site collection, the relative URL will correspond with a path on the <b>base site</b>.  If the <u>Url</u> connection property points to a specific site, the relative URL will be relative to the site supplied in the <u>Url</u>.</p>
Name	<i>String</i>	<i>False</i>	<p>The name assigned to the new file.</p> <p>If uploading to the root directory of the site, supply only the filename.  If uploading to a folder nested at some level inside the root directory, <b>prepend the full parent subdirectory</b>.</p>

			For example: <b>Root Directory:</b> <i>filename.txt</i> <b>Sub-directory:</b> <i>MyFolder/filename.txt</i>
Content	<i>String</i>	<i>True</i>	The content as InputStream to be uploaded when File is not specified.
ReturnID	<i>String</i>	<i>False</i>	Boolean value indicating whether to return the ID and other metadata fields of the newly created record. An extra API request is needed to get this value.  The default value is <i>false</i> .
MetadataName#	<i>String</i>	<i>False</i>	The name of a metadata field to be set for the document. It must be the column name present in the table of library.
MetadataValue#	<i>String</i>	<i>False</i>	The value of a metadata field to be set for the document.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the stored procedure was successful.
ID	<i>String</i>	The internal ID of the document.
_dlc_DocId	<i>String</i>	The document ID.
_dlc_DocIdUrl	<i>String</i>	The URL of the document ID.

Vti_author	<i>String</i>	The creator of the document.
Vti_etag	<i>String</i>	The e-tag of the document.
FileRef	<i>String</i>	The file reference of the document.

## REST Data Model

The SharePoint Adapter models SharePoint entities in relational Tables, Views, and Stored Procedures. The table definitions are dynamically obtained based on your SharePoint site. Any changes you make, such as adding a custom field or changing a field's data type, are automatically reflected when you connect.

### Tables

Tables describes the available tables.

Lists in your sharepoint site are exposed as relational tables dynamically. Which means any change you make in your lists, i.e adding new list or adding new fields, will be reflected on the driver.

### Views

Views are tables that cannot be modified. Typically, read-only data are shown as views.

### Stored Procedures

[Stored Procedures](#) are function-like interfaces to the data source. They surface additional capabilities of the SharePoint API such as searching, updating, and modifying information.

### Using OData standard

Since the REST API is OData based, server side filters, are done using OData standard. So the driver takes the most of the server filtering, by reading the metadata file and determining



which filters can be done on the server.

**NOTE:** When executing "SELECT \*" queries, the SharePoint REST API response, does not return all the available fields. So to avoid too many null values, the provider will select all the columns explicitly using the **\$select** filter. However, the provider will do this only if the **\$select** filter's length is not bigger than 1500, to avoid an error from SharePoint REST API regarding the URL length. This is a limitation of the SharePoint REST API, so in these cases, the only way to see the actual value of some columns, is to explicitly select them in your query.

## 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.

### SharePoint Adapter Views

Name	Description
<a href="#">Attachments</a>	Read Attachments for the specified item on the specified list.
<a href="#">Files</a>	Query the available files on your sharepoint site.
<a href="#">Groups</a>	Query the available groups on your sharepoint site.
<a href="#">Lists</a>	Query the available lists on your sharepoint site.
<a href="#">RoleAssignmentMember</a>	Get Web RoleAssignments member.
<a href="#">RoleAssignments</a>	Get Web RoleAssignments.

<a href="#">RoleDefinitionBindings</a>	Get Web Role definition binding.
<a href="#">Roles</a>	Query the roles your users can have.
<a href="#">Subsites</a>	This lists the available subsites.
<a href="#">Users</a>	Query the available users on your sharepoint site.

## Attachments

Read Attachments for the specified item on the specified list.

### Table Specific Information

#### Select

**Note:** List and ItemId are required to return Attachments.

List can be fetched from the Lists view(Title column).

```
SELECT * FROM Attachments WHERE List = 'TestApp' AND ItemID = 1
```

#### Columns

Name	Type	References	Description
Id [KEY]	<i>String</i>		The Id of the attachment on the item.
Updated	<i>Datetime</i>		The updated date of the attachment on the item.
FileName	<i>String</i>		The FileName of the attachment on the item.

ServerRelativePath_DecodedUrl	<i>String</i>	The ServerRelativePath DecodedUrl of the attachment on the item.
FileNameAsPath_DecodedUrl	<i>String</i>	The FileNameAsPath DecodedUrl of the attachment on the item.
ServerRelativeUrl	<i>String</i>	The ServerRelativeUrl of the attachment on the item.
List	<i>String</i>	The internal name of the list to retrieve attachments from.
ItemID	<i>String</i>	The ID of the item on the list to retrieve attachments from.

## Files

Query the available files on your sharepoint site.

## Columns

<b>Name</b>	<b>Type</b>	<b>References</b>	<b>Description</b>
Id [KEY]	<i>String</i>		
CreatedBy_Id	<i>String</i>		
CreatedBy_Name	<i>String</i>		

CreatedBy_Puid	<i>String</i>
ETag	<i>String</i>
LastModifiedBy_Id	<i>String</i>
LastModifiedBy_Name	<i>String</i>
LastModifiedBy_Puid	<i>String</i>
Name	<i>String</i>
Size	<i>Int</i>
TimeCreated	<i>Datetime</i>
TimeLastModified	<i>Datetime</i>
Url	<i>String</i>

## Groups

Query the available groups on your sharepoint site.

### Table Specific Information

#### Select

```
SELECT * FROM Groups
```

## Columns

Name	Type	References	Description
Id [KEY]	<i>Int</i>		The Group Id.
AllowMembersEditMembership	<i>Bool</i>		
AllowRequestToJoinLeave	<i>Bool</i>		
AutoAcceptRequestToJoinLeave	<i>Bool</i>		
CanCurrentUserEditMembership	<i>Bool</i>		
CanCurrentUserManageGroup	<i>Bool</i>		
CanCurrentUserViewMembership	<i>Bool</i>		
Description	<i>String</i>		
OnlyAllowMembersViewMembership	<i>Bool</i>		
OwnerTitle	<i>String</i>		
RequestToJoinLeaveEmailSetting	<i>String</i>		

LinkedOwner	<i>String</i>
LinkedUsers	<i>String</i>

## Lists

Query the available lists on your sharepoint site.

## Columns

Name	Type	References	Description
HasUniqueRoleAssignments [KEY]	<i>Bool</i>		
LinkedFirstUniqueAncestorSecurableObject	<i>String</i>		
LinkedRoleAssignments	<i>String</i>		
ServerRelativeUrl	<i>String</i>		This column will be available when the URL is a subsite.
Id [KEY]	<i>String</i>		
AllowContentTypes	<i>Bool</i>		

---

AllowDeletion	<i>Bool</i>
BaseTemplate	<i>Int</i>
BaseType	<i>Int</i>
BrowserFileHandling	<i>Int</i>
ContentTypesEnabled	<i>Bool</i>
CrawlNonDefaultViews	<i>Bool</i>
Created	<i>Datetime</i>
CurrentChangeToken_StringValue	<i>String</i>
CustomActionElements_Items	<i>String</i>
DataSource_Properties	<i>String</i>
DefaultContentApprovalWorkflowId	<i>String</i>
DefaultDisplayFormUrl	<i>String</i>
DefaultEditFormUrl	<i>String</i>

---

DefaultItemOpenUseListSetting	<i>Bool</i>
DefaultNewFormUrl	<i>String</i>
DefaultViewPath_DecodedUrl	<i>String</i>
DefaultViewUrl	<i>String</i>
Description	<i>String</i>
Direction	<i>String</i>
DisableGridEditing	<i>Bool</i>
DocumentTemplateUrl	<i>String</i>
DraftVersionVisibility	<i>Int</i>
EffectiveBasePermissions_High	<i>Long</i>
EffectiveBasePermissions_Low	<i>Long</i>
EffectiveBasePermissionsForUI_High	<i>Long</i>
EffectiveBasePermissionsForUI_Low	<i>Long</i>
EnableAssignToEmail	<i>Bool</i>



EnableAttachments	<i>Bool</i>
EnableFolderCreation	<i>Bool</i>
EnableMinorVersions	<i>Bool</i>
EnableModeration	<i>Bool</i>
EnableRequestSignOff	<i>Bool</i>
EnableVersioning	<i>Bool</i>
EntityTypeNames	<i>String</i>
ExcludeFromOfflineClient	<i>Bool</i>
ExemptFromBlockDownloadOfNonViewableFiles	<i>Bool</i>
FileSavePostProcessingEnabled	<i>Bool</i>
ForceCheckout	<i>Bool</i>
HasExternalDataSource	<i>Bool</i>
Hidden	<i>Bool</i>

ImagePath_DecodedUrl	<i>String</i>
ImageUrl	<i>String</i>
IrmEnabled	<i>Bool</i>
IrmExpire	<i>Bool</i>
IrmReject	<i>Bool</i>
IsApplicationList	<i>Bool</i>
IsCatalog	<i>Bool</i>
IsEnterpriseGalleryLibrary	<i>Bool</i>
IsPrivate	<i>Bool</i>
IsSiteAssetsLibrary	<i>Bool</i>
IsSystemList	<i>Bool</i>
ItemCount	<i>Int</i>
LastItemDeletedDate	<i>Datetime</i>
LastItemModifiedDate	<i>Datetime</i>

---

---

LastItemUserModifiedDate	<i>Datetime</i>
ListExperienceOptions	<i>Int</i>
ListItemEntityTypeFullName	<i>String</i>
MajorVersionLimit	<i>Int</i>
MajorWithMinorVersionsLimit	<i>Int</i>
MultipleDataList	<i>Bool</i>
NoCrawl	<i>Bool</i>
OnQuickLaunch	<i>Bool</i>
PageRenderType	<i>Int</i>
ParentWebPath_DecodedUrl	<i>String</i>
ParentWebUrl	<i>String</i>
ParserDisabled	<i>Bool</i>
ReadSecurity	<i>Int</i>

---

---

SchemaXml	<i>String</i>
ServerTemplateCanCreateFolders	<i>Bool</i>
TemplateFeatureId	<i>String</i>
Title	<i>String</i>
ValidationFormula	<i>String</i>
ValidationMessage	<i>String</i>
WriteSecurity	<i>Int</i>
LinkedContentTypes	<i>String</i>
LinkedCreatablesInfo	<i>String</i>
LinkedDefaultView	<i>String</i>
LinkedDescriptionResource	<i>String</i>
LinkedEventReceivers	<i>String</i>
LinkedFields	<i>String</i>
LinkedForms	<i>String</i>

---

LinkedInformationRightsManagementSettings	<i>String</i>
LinkedItems	<i>String</i>
LinkedParentWeb	<i>String</i>
LinkedRootFolder	<i>String</i>
LinkedSubscriptions	<i>String</i>
LinkedTitleResource	<i>String</i>
LinkedUserCustomActions	<i>String</i>
LinkedViews	<i>String</i>
LinkedWorkflowAssociations	<i>String</i>

---

## RoleAssignmentMember

Get Web RoleAssignments member.

### Table Specific Information

#### Select

**Note:** PrincipalId is required to return RoleAssignmentMember.

```

SELECT * FROM RoleAssignmentMember WHERE PrincipalId = 3
SELECT * FROM RoleAssignmentMember WHERE List = 'TestApp' AND
PrincipalId = 3
SELECT * FROM RoleAssignmentMember WHERE PrincipalId = 5 AND list =
'KatsunariMatsumoto' AND ItemId = '3'

```

## Columns

Name	Type	References	Description
ID [KEY]	<i>String</i>		The ID of the role assigned member.
Updated	<i>Datetime</i>		The updated date for role assigned member.
IsHiddenInUI	<i>Boolean</i>		A boolean indicating if the assigned role member is hidden in UI.
LoginName	<i>String</i>		The login name of the role assigned member.
Title	<i>String</i>		The title of the role assigned member.
AllowMembersEditMembership	<i>Boolean</i>		A boolean indicating whether to allow members

			edit membership.
AllowRequestToJoinLeave	<i>Boolean</i>		A boolean indicating whether to allow request to join leave.
AutoAcceptRequestToJoinLeave	<i>Boolean</i>		A boolean indicating whether to auto accept request to join leave.
Description	<i>String</i>		The description of the role assigned member.
OnlyAllowMembersViewMembership	<i>Boolean</i>		A boolean indicating whether to only allow members view membership.
OwnerTitle	<i>String</i>		The owner title of the role assigned member.
RequestToJoinLeaveEmailSetting	<i>String</i>		The request to join leave email setting of the role assigned member.
PrincipalId	<i>String</i>	<a href="#">RoleAssignments</a> .PrincipalId	The Principal Id.

List	<i>String</i>	<a href="#">Lists</a> .Title	The internal name of the list to retrieve role assigned member from.
ItemId	<i>String</i>		List item id for role assignment.

## RoleAssignments

Get Web RoleAssignments.

### Table Specific Information

#### Select

```
SELECT * FROM RoleAssignments WHERE List = 'TestApp'
SELECT * FROM RoleAssignments WHERE PrincipalId = 5 AND list =
'KatsunariMatsumoto' AND ItemId = '3'
```

#### Columns

Name	Type	References	Description
ID [KEY]	<i>String</i>		The ID of the role assigned.
PrincipalId	<i>Long</i>		The Principal ID for role assigned.
Updated	<i>Datetime</i>		The updated date for role assigned.



List	String	<a href="#">Lists.Title</a>	The internal name of the list to retrieve role assigned from.
ItemId	String		List item id for role assignment.

## RoleDefinitionBindings

Get Web Role definition binding.

### Table Specific Information

#### Select

**Note:** PrincipalId is required to return RoleDefinitionBindings.

```
SELECT * FROM RoleDefinitionBindings WHERE PrincipalId = 3
SELECT * FROM RoleDefinitionBindings WHERE List = 'TestApp' AND
PrincipalId = 3
SELECT * FROM RoleDefinitionBindings WHERE PrincipalId = 5 AND list =
'KatsunariMatsumoto' AND ItemId = '3'
```

#### Columns

Name	Type	References	Description
ID [KEY]	String		The ID of the role assigned member.
BasePermissions_High	Int64		The base permissions high.

BasePermissions_Low	<i>Int64</i>		The base permissions low.
Description	<i>String</i>		The description.
Hidden	<i>Boolean</i>		A boolean indicating if it is hidden.
Name	<i>String</i>		The name.
Order	<i>Int</i>		The order.
RoleTypeKind	<i>Int</i>		The role type kind.
PrincipalId	<i>String</i>	<a href="#">RoleAssignments</a> .PrincipalId	The Principal Id.
List	<i>String</i>	<a href="#">Lists</a> .Title	The internal name of the list to retrieve role assigned member from.
ItemId	<i>String</i>		List item id for role defintion.

## Roles

Query the roles your users can have.

## Columns

Name	Type	References	Description
Id	<i>Int</i>		

[KEY]	
BasePermissions_High	<i>Long</i>
BasePermissions_Low	<i>Long</i>
Description	<i>String</i>
Hidden	<i>Bool</i>
Name	<i>String</i>
Order	<i>Int</i>
RoleTypeKind	<i>Int</i>

## Subsites

This lists the available subsites.

### Table Specific Information

#### Select

```
SELECT * FROM Subsites
```

#### Columns

Name	Type	References	Description
Id [KEY]	String		The Id of the subsite.
AllowRssFeeds	Boolean		The AllowRssFeeds of the subsite.
AlternateCssUrl	String		The AlternateCssUrl of the subsite.
AppInstanceId	String		The AppInstanceId of the subsite.
ClassicWelcomePage	String		The ClassicWelcomePage of the subsite.
Configuration	Int		The Configuration of the subsite.
Created	Datetime		The Created of the subsite.
CurrentChangeToken_StringValue	String		The CurrentChangeToken_StringValue of the subsite.
CustomMasterUrl	String		The CustomMasterUrl of the subsite.
Description	String		The Description of the subsite.
DesignPackageId	String		The DesignPackageId of the subsite.
DocumentLibraryCalloutOfficeWebAppPreviewersDisabled	Boolean		The DocumentLibraryCalloutOfficeWebAppPreviewersDisabled of the

		subsite.
EnableMinimalDownload	<i>Boolean</i>	The EnableMinimalDownload of the subsite.
FooterEmphasis	<i>Int</i>	The FooterEmphasis of the subsite.
FooterEnabled	<i>Boolean</i>	The FooterEnabled of the subsite.
FooterLayout	<i>Int</i>	The FooterLayout of the subsite.
HeaderEmphasis	<i>Int</i>	The HeaderEmphasis of the subsite.
HeaderLayout	<i>Int</i>	The HeaderLayout of the subsite.
HideTitleInHeader	<i>Boolean</i>	The HideTitleInHeader of the subsite.
HorizontalQuickLaunch	<i>Boolean</i>	The HorizontalQuickLaunch of the subsite.
IsHomepageModernized	<i>Boolean</i>	The IsHomepageModernized of the subsite.
IsMultilingual	<i>Boolean</i>	The IsMultilingual of the subsite.
IsRevertHomepageLinkHidden	<i>Boolean</i>	The IsRevertHomepageLinkHidden of the subsite.
KeepFieldUserResources	<i>Boolean</i>	The KeepFieldUserResources of the subsite.
Language	<i>Int</i>	The Language of the subsite.

LastItemModifiedDate	<i>DateTime</i>	The LastItemModifiedDate of the subsite.
LastItemUserModifiedDate	<i>DateTime</i>	The LastItemUserModifiedDate of the subsite.
LogoAlignment	<i>Int</i>	The LogoAlignment of the subsite.
MasterUrl	<i>String</i>	The MasterUrl of the subsite.
MegaMenuEnabled	<i>Boolean</i>	The MegaMenuEnabled of the subsite.
NavAudienceTargetingEnabled	<i>Boolean</i>	The NavAudienceTargetingEnabled of the subsite.
NoCrawl	<i>Boolean</i>	The NoCrawl of the subsite.
ObjectCacheEnabled	<i>Boolean</i>	The ObjectCacheEnabled of the subsite.
OverwriteTranslationsOnChange	<i>Boolean</i>	The OverwriteTranslationsOnChange of the subsite.
ResourcePath_DecodedUrl	<i>String</i>	The ResourcePath_DecodedUrl of the subsite.
QuickLaunchEnabled	<i>Boolean</i>	The QuickLaunchEnabled of the subsite.
RecycleBinEnabled	<i>Boolean</i>	The RecycleBinEnabled of the subsite.
SearchScope	<i>Int</i>	The SearchScope of the subsite.

ServerRelativeUrl	<i>String</i>	The ServerRelativeUrl of the subsite.
SiteLogoUrl	<i>String</i>	The SiteLogoUrl of the subsite.
SyndicationEnabled	<i>Boolean</i>	The SyndicationEnabled of the subsite.
TenantAdminMembersCanShare	<i>Int</i>	The TenantAdminMembersCanShare of the subsite.
Title	<i>String</i>	The Title of the subsite.
TreeViewEnabled	<i>Boolean</i>	The TreeViewEnabled of the subsite.
UIVersion	<i>Int</i>	The UIVersion of the subsite.
UIVersionConfigurationEnabled	<i>Boolean</i>	The UIVersionConfigurationEnabled of the subsite.
Url	<i>String</i>	The Url of the subsite.
WebTemplate	<i>String</i>	The WebTemplate of the subsite.
WelcomePage	<i>String</i>	The WelcomePage of the subsite.

## Users

Query the available users on your sharepoint site.

## Table Specific Information

### Select

```
SELECT * FROM Users // Fetch all the Users
SELECT * FROM Users WHERE GroupId = 5 // Fetch a user for a particular Group
```

### Columns

Name	Type	References	Description
AadObjectId_Nameld	<i>String</i>		
AadObjectId_NameldIssuer	<i>String</i>		
Email	<i>String</i>		
Expiration	<i>String</i>		
IsEmailAuthenticationGuestUser	<i>Bool</i>		
IsShareByEmailGuestUser	<i>Bool</i>		
IsSiteAdmin	<i>Bool</i>		
UserId_Nameld	<i>String</i>		



UserId_NameldIssuer	<i>String</i>	
UserPrincipalName	<i>String</i>	
LinkedAlerts	<i>String</i>	
LinkedGroups	<i>String</i>	
LoginName	<i>String</i>	The login name of the user.
Title	<i>String</i>	The Title of the user.
IsHiddenInUI	<i>Boolean</i>	A boolean indicating if the user is hidden in UI.
GroupId	<i>Int</i>	The group id the user is added to.

## Stored Procedures

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

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

### SharePoint Adapter Stored Procedures

<b>Name</b>	<b>Description</b>

<a href="#">AddAttachment</a>	Add an attachment into a sharepoint list.
<a href="#">CheckInDocument</a>	Check in specific document inside document library.
<a href="#">CheckOutDocument</a>	Check out specific document inside document library.
<a href="#">CopyDocument</a>	Copies the file to the destination URL..
<a href="#">CreateFolder</a>	Adds a folder to a document library on a SharePoint site.
<a href="#">DeleteAttachment</a>	Deletes an attachment from Sharepoint list.
<a href="#">DeleteDocument</a>	Delete a Document from the sharepoint document library.
<a href="#">DiscardCheckOutDocument</a>	Reverts an existing checkout for the file.
<a href="#">DownloadAttachment</a>	Download a document from the SharePoint list.
<a href="#">DownloadDocument</a>	Download a document from the SharePoint library.
<a href="#">GetAdminConsentURL</a>	Gets the admin consent URL that must be opened separately by an admin of a given domain to grant access to your application. Only needed when using custom OAuth credentials.
<a href="#">GetCurrentUser</a>	Retrieves information about the current logged in user.
<a href="#">GetOAuthAccessToken</a>	Gets the OAuth access token from SharePoint.
<a href="#">GetOAuthAuthorizationURL</a>	Gets the SharePoint authorization URL. Access the URL returned in the output in a Web browser. This requests the access token that can be used as part of the connection string to SharePoint.
<a href="#">MoveAttachmentOrDocument</a>	Moves a document or attachment from source folder to destination folder.
<a href="#">RefreshOAuthAccessToken</a>	Refreshes the OAuth access token used for

	authentication with SharePoint.
<a href="#">RenameAttachmentOrDocument</a>	Renames a document or attachment from source folder to destination folder.
<a href="#">UploadDocument</a>	Uploads a document into the SharePoint library.

## AddAttachment

Add an attachment into a sharepoint list.

### Input

Name	Type	Accepts Input Streams	Description
ListTitle	<i>String</i>	<i>False</i>	Title of the list item.
ItemId	<i>String</i>	<i>False</i>	Id of the list item.
FileName	<i>String</i>	<i>False</i>	Name of the File to be added in document library
InputFilePath	<i>String</i>	<i>False</i>	Location of the file to be attached.
Content	<i>String</i>	<i>True</i>	The content as InputStream to be uploaded when InputFilePath is not specified.

### Result Set Columns

Name	Type	Description
RelativeUrl	<i>String</i>	Relative URL of the attachment that has been added.
Status	<i>String</i>	Message indicating whether the operation was successful or not.

## CheckInDocument

Check in specific document inside document library.

### Input

Name	Type	Description
RelativeURL	<i>String</i>	Relative URL of the folder.
DocumentName	<i>String</i>	Name of the File to be Checked in.
Comment	<i>String</i>	Optional message while Checking in Document.

### Result Set Columns

Name	Type	Description
Status	<i>String</i>	Message indicating whether the operation was successful or not.

## CheckOutDocument

Check out specific document inside document library.

### Input

Name	Type	Description
RelativeURL	<i>String</i>	Relative URL of the folder.
DocumentName	<i>String</i>	Name of the File to be Checked in.

### Result Set Columns

Name	Type	Description
Status	<i>String</i>	Message indicating whether the operation was successful or not.

## CopyDocument

Copies the file to the destination URL..

### Input

Name	Type	Description
SourceFileRelativeUrl	<i>String</i>	Relative URL of the source file.

DestFileRelativeUrl	<i>String</i>	Relative URL of the destination file.
---------------------	---------------	---------------------------------------

## Result Set Columns

Name	Type	Description
Status	<i>String</i>	Message indicating whether the operation was successful or not.

## CreateFolder

Adds a folder to a document library on a SharePoint site.

### Input

Name	Type	Description
RelativeURL	<i>String</i>	Relative URL of the folder.
FolderName	<i>String</i>	Name of the folder to be created.

## Result Set Columns

Name	Type	Description
Id	<i>String</i>	Unique identifier Value Returned after completing the operation.

---

Status	<i>String</i>	Message indicating whether the operation was successful or not.
--------	---------------	---

---

## DeleteAttachment

Deletes an attachment from Sharepoint list.

### Input

Name	Type	Description
ListTitle	<i>String</i>	Title of the list item.
ItemId	<i>String</i>	Id of the list item.
FileName	<i>String</i>	Name of the File to be added in document library

### Result Set Columns

Name	Type	Description
Status	<i>String</i>	Message indicating whether the operation was successful or not.

---

## DeleteDocument

Delete a Document from the sharepoint document library.

## Input

Name	Type	Description
RelativeURL	<i>String</i>	Relative URL of the Folder.
DocumentName	<i>String</i>	Name of the File to be Deleted.

## Result Set Columns

Name	Type	Description
Status	<i>String</i>	Message indicating whether the operation was successful or not.

## DiscardCheckoutDocument

Reverts an existing checkout for the file.

## Input

Name	Type	Description
RelativeURL	<i>String</i>	Relative URL of the folder.
DocumentName	<i>String</i>	Name of the File to be Checked in.



## Result Set Columns

Name	Type	Description
Status	<i>String</i>	Message indicating whether the operation was successful or not.

## DownloadAttachment

Download a document from the SharePoint list.

### Stored Procedure Specific Information

#### Insert

RemoteFile can be both relative to the server, or it can be the full URL of the file. Below are some examples:

```
EXECUTE DownloadAttachment File =
'C:/Users/User/Desktop/DownloadedFile.txt', RemoteFile =
'https://mysite.sharepoint.com/Lists/MyCustomList/Attachments/1/FileToDownload.txt';
EXECUTE DownloadAttachment File =
'C:/Users/User/Desktop/DownloadedFile.txt', RemoteFile =
'/Lists/MyCustomList/Attachments/1/FileToDownload.txt';
```

#### Input

Name	Type	Accepts Output Streams	Description
File	<i>String</i>	<i>False</i>	The path of the file to be saved.

RemoteFile	<i>String</i>	<i>False</i>	The path of the file on the server. This can be the full URL or simply the file name. If you use the name of the file, the latest version will be downloaded.
Encoding	<i>String</i>	<i>False</i>	<p>The data will be output to FileData in the specified encoding.</p> <p>The allowed values are <i>NONE</i>, <i>BASE64</i>.</p> <p>The default value is <i>BASE64</i>.</p>
FileStream	<i>String</i>	<i>True</i>	OutputStream to write the downloaded attachment. Only used if File is not provided.
ReadTimeout	<i>String</i>	<i>False</i>	How many seconds the download may run. If given the download always fails if it runs longer than this, unlike Timeout which only triggers if the download stalls.

## Result Set Columns

<b>Name</b>	<b>Type</b>	<b>Description</b>
Result	<i>String</i>	Boolean value indicating whether the operation was successful.
FileData	<i>String</i>	The downloaded file content. Only returned if File and FileStream is not set.

## DownloadDocument

Download a document from the SharePoint library.

## Stored Procedure Specific Information

### Insert

RemoteFile can be both relative to the library, or it can be the full URL of the file. Below are some examples:

```
EXECUTE DownloadDocument File =
'C:/Users/User/Desktop/DownloadedFile.txt', Library = 'Shared
Documents', RemoteFile = 'https://mysite.sharepoint.com/Shared
Documents/newFolder/FileToDownload.txt';
EXECUTE DownloadDocument File =
'C:/Users/User/Desktop/DownloadedFile.txt', Library = 'Shared
Documents', RemoteFile = '/newFolder/FileToDownload.txt';
```

### Input

Name	Type	Accepts Output Streams	Description
File	<i>String</i>	<i>False</i>	The path of the file to be saved. You should include the new filename. For example, 'C:/Users/User/Desktop/DownloadedFile.txt'.
Library	<i>String</i>	<i>False</i>	The name of the library on the SharePoint server. For example, 'Shared Documents'.
RemoteFile	<i>String</i>	<i>False</i>	This can be either the relative path to the library or the full URL of the file.
Encoding	<i>String</i>	<i>False</i>	The data will be output to FileData in the specified encoding.  The allowed values are <i>NONE</i> , <i>BASE64</i> .  The default value is <i>BASE64</i> .
FileStream	<i>String</i>	<i>True</i>	OutputStream to write the downloaded

			document. Only returned if File is not set.
ReadTimeout	<i>String</i>	<i>False</i>	How many seconds the download may run. If given the download always fails if it runs longer than this, unlike Timeout which only triggers if the download stalls.

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Boolean value indicating whether the operation was successful.
FileData	<i>String</i>	The downloaded file content. Only returned if File and FileStream is not set.

## GetAdminConsentURL

Gets the admin consent URL that must be opened separately by an admin of a given domain to grant access to your application. Only needed when using custom OAuth credentials.

## 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 Reply URL in the Azure AD app settings.
State	<i>String</i>	The same value for state that you sent when you requested the authorization code.

Scope	<i>String</i>	The scope or permissions you are requesting from the Admin The default value is <i>AllSites.Manage</i> .
-------	---------------	---

## 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.

## GetCurrentUser

Retrieves information about the current logged in user.

## Result Set Columns

Name	Type	Description
Id	<i>Int</i>	The id of the user.
Title	<i>String</i>	The title of the user.
Email	<i>String</i>	The email of the user.
IsSiteAdmin	<i>Boolean</i>	Whether the user is a site admin or not.

## GetOAuthAccessToken

Gets the OAuth access token from SharePoint.

### Input

Name	Type	Description
AuthMode	<i>String</i>	The type of authentication mode to use. The allowed values are APP, WEB.
Verifier	<i>String</i>	The verifier token returned by SharePoint after using the URL obtained with GetOAuthAuthorizationURL. Required for only the Web AuthMode.
CallbackUrl	<i>String</i>	The URL the user will be redirected to after authorizing your application.
Scope	<i>String</i>	The scope or permissions you are requesting from the user.
State	<i>String</i>	Any value that you wish to be sent with the callback.
Prompt	<i>String</i>	Defaults to 'select_account' which prompts the user to select account while authenticating. Set to 'None', for no prompt, 'login' to force user to enter their credentials or 'consent' to trigger the OAuth consent dialog after the user signs in, asking the user to grant permissions to the app.

### Result Set Columns

Name	Type	Description

OAuthAccessToken	<i>String</i>	The authentication token returned from SharePoint.
OAuthRefreshToken	<i>String</i>	A token that may be used to obtain a new access token.
ExpiresIn	<i>String</i>	The remaining lifetime for the access token in seconds.

## GetOAuthAuthorizationURL

Gets the SharePoint authorization URL. Access the URL returned in the output in a Web browser. This requests the access token that can be used as part of the connection string to SharePoint.

### Input

<b>Name</b>	<b>Type</b>	<b>Description</b>
CallbackUrl	<i>String</i>	The URL that Sharepoint will return to after the user has authorized your app.
Scope	<i>String</i>	The scope or permissions you are requesting from the user. The default value is <i>AllSites.Manage</i> .
State	<i>String</i>	Any value that you wish to be sent with the callback.
Prompt	<i>String</i>	Defaults to 'select_account' which prompts the user to select account while authenticating. Set to 'None', for no prompt, 'login' to force user to enter their credentials or 'consent' to trigger the OAuth consent dialog after the user signs in, asking the user to grant permissions to the app.

## Result Set Columns

Name	Type	Description
URL	<i>String</i>	The URL to be entered into a Web browser to obtain the verifier token and authorize the data provider with.

## MoveAttachmentOrDocument

Moves a document or attachment from source folder to destination folder.

The MoveAttachmentOrDocument stored procedure requires SourceFileURL and DestinationFolderURL parameters to move an attachment or document.

For Example:

```
EXEC MoveAttachmentOrDocument SourceFileURL = '/Shared
Documents/Dummy.txt', DestinationFolderURL = '/Shared Documents/MySite'
```

## Input

Name	Type	Description
SourceFileURL	<i>String</i>	<p>URL of the source file, relative to the base <u>Url</u> supplied in the adapter's connection properties.</p> <p>For example:  <b>Root Directory file:</b> /Shared Documents/filename.txt  <b>Sub-directory file:</b>/Shared Documents/MyFolder/filename.txt</p> <p>If you set the <u>Url</u> connection property to a site collection, the relative URL will correspond with a path on the <b>base site</b>.          If the <u>Url</u> connection property points to a specific site, the relative URL will be relative to the site supplied in the <u>Url</u>.</p>



DestinationFolderURL	String	URL of the destination folder where you want to move the file ,relative to the base <u>Url</u> supplied in the adapter's connection properties.
----------------------	--------	---

For example:

**Root Directory:** /Shared Documents/

**Sub-directory:**/Shared Documents/MyFolder/

If you set the Url connection property to a site collection, the relative URL will correspond with a path on the **base site**.

If the Url connection property points to a specific site, the relative URL will be relative to the site supplied in the Url.

## Result Set Columns

Name	Type	Description
Result	String	Value indicating whether the operation was successful.

## RefreshOAuthAccessToken

Refreshes the OAuth access token used for authentication with SharePoint.

### Input

Name	Type	Description
OAuthRefreshToken	String	The old token to be refreshed.

## Result Set Columns

Name	Type	Description
OAuthAccessToken	<i>String</i>	The authentication token returned from SharePoint.
ExpiresIn	<i>String</i>	The remaining lifetime on the access token.

## RenameAttachmentOrDocument

Renames a document or attachment from source folder to destination folder.

The RenameAttachmentOrDocument stored procedure requires SourceFileURL and NewFileName parameters to rename an attachment or document.

For Example:

```
EXEC RenameAttachmentOrDocument SourceFileURL = '/Shared
Documents/Dummy.txt', NewFileName = 'Dummy1.txt'
```

## Input

Name	Type	Description
SourceFileURL	<i>String</i>	<p>URL of the source file, relative to the base <u>Url</u> supplied in the adapter's connection properties.</p> <p>For example:  <b>Root Directory file:</b> /Shared Documents/filename.txt  <b>Sub-directory file:</b> /Shared Documents/MyFolder/filename.txt            If you set the <u>Url</u> connection property to a site collection, the relative URL will correspond with a path on the <b>base site</b>.            If the <u>Url</u> connection property points to a specific site, the relative URL will be relative to the site supplied in the <u>Url</u>.</p>

NewFileName	<i>String</i>	New name of the file with extension.
-------------	---------------	--------------------------------------

## Result Set Columns

Name	Type	Description
Result	<i>String</i>	Value indicating whether the operation was successful.

## UploadDocument

Uploads a document into the SharePoint library.

### Input

Name	Type	Accepts Input Streams	Description
RelativeUrl	<i>String</i>	<i>False</i>	<p>The URL of the folder you would like to upload file(s) to, relative to the base <u>Url</u> supplied in the adapter's connection properties.</p> <p>For example:            Root folder: <i>Shared Documents</i>            Sub-folder: <i>Shared Documents/MyFolder</i></p> <p>If you set the <u>Url</u> connection property to a site collection, the relative URL will correspond with a path on the <b>base site</b>.            If the <u>Url</u> connection property points to a specific site, the relative URL will be relative to the site</p>

			supplied in the <u>Url</u> .
InputFilePath	<i>String</i>	<i>False</i>	The location of the file to be uploaded.
FileName	<i>String</i>	<i>False</i>	Name of the File to be created.
Overwrite	<i>String</i>	<i>False</i>	Boolean value to overwrite the existing file data.
Content	<i>String</i>	<i>True</i>	The content as InputStream to be uploaded when InputFilePath is not specified.

## Result Set Columns

Name	Type	Description
Id	<i>String</i>	Unique identifier Value Returned after completing the operation.
Status	<i>String</i>	Message indicating whether the operation was successful or not.

# Data Type Mapping

## Data Type Mappings

The adapter maps types from the data source to the corresponding data type available in the schema. The table below documents these mappings.

SharePoint	CData Schema
------------	--------------

Choice (menu)	string
Currency	float
Date and Time	datetime
Hyperlink or Picture	string
Lookup	string
Multiple lines of text	string
Number	float
Person or Group	string
Single line of text	string
Task Outcome	string
Yes/No	bool

## 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="#">AuthScheme</a>	The scheme used for authenticating to SharePoint.
<a href="#">URL</a>	The base URL for the site.

<a href="#">SharePointEdition</a>	The edition of SharePoint being used. Set either SharePoint Online or SharePoint On-Premise.
<a href="#">User</a>	The SharePoint user account used to authenticate.
<a href="#">Password</a>	The password used to authenticate the user.

## Azure Authentication

Property	Description
<a href="#">AzureTenant</a>	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
<a href="#">AzureEnvironment</a>	The Azure Environment to use when establishing a connection.

## SSO

Property	Description
<a href="#">SSOLoginURL</a>	The identity provider's login URL.
<a href="#">SSODomain</a>	The domain of the user when using single sign-on (SSO).
<a href="#">SSOProperties</a>	Additional properties required to connect to the identity provider in a semicolon-separated list.

## 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="#">CallbackURL</a>	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your Add-In settings.
<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="#">Scope</a>	The scope used for the OAuth flow to access data from the Application.
<a href="#">OAuthGrantType</a>	The grant type for the OAuth flow.
<a href="#">OAuthVerifier</a>	The verifier code returned from the OAuth authorization URL.
<a href="#">OAuthRefreshToken</a>	The OAuth refresh token for the corresponding OAuth access token.
<a href="#">OAuthExpiresIn</a>	The lifetime in seconds of the OAuth AccessToken.
<a href="#">OAuthTokenTimestamp</a>	The Unix epoch timestamp in milliseconds when the current Access Token was created.

## JWT OAuth

Property	Description

<a href="#">OAuthJWTCert</a>	The JWT Certificate store.
<a href="#">OAuthJWTCertType</a>	The type of key store containing the JWT Certificate.
<a href="#">OAuthJWTIssuer</a>	The issuer of the Java Web Token.

## Kerberos

Property	Description
<a href="#">KerberosKDC</a>	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
<a href="#">KerberosRealm</a>	The Kerberos Realm used to authenticate the user.
<a href="#">KerberosSPN</a>	The service principal name (SPN) for the Kerberos Domain Controller.
<a href="#">KerberosKeytabFile</a>	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
<a href="#">KerberosServiceRealm</a>	The Kerberos realm of the service.
<a href="#">KerberosServiceKDC</a>	The Kerberos KDC of the service.
<a href="#">KerberosTicketCache</a>	The full file path to an MIT Kerberos credential cache file.

## SSL

Property	Description
<a href="#">SSLClientCert</a>	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).



<a href="#">SSLClientCertType</a>	The type of key store containing the TLS/SSL client certificate.
<a href="#">SSLClientCertPassword</a>	The password for the TLS/SSL client certificate.
<a href="#">SSLClientCertSubject</a>	The subject of the TLS/SSL client certificate.
<a href="#">SSLServerCert</a>	The certificate to be accepted from the server when connecting using TLS/SSL.

## Firewall

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

## Proxy

Property	Description
<a href="#">ProxyAutoDetect</a>	This indicates whether to use the system proxy settings or not. This takes precedence over other proxy settings, so you'll need to set ProxyAutoDetect to FALSE in order use custom proxy settings.
<a href="#">ProxyServer</a>	The hostname or IP address of a proxy to route HTTP traffic through.
<a href="#">ProxyPort</a>	The TCP port the ProxyServer proxy is running on.

<a href="#">ProxyAuthScheme</a>	The authentication type to use to authenticate to the ProxyServer proxy.
<a href="#">ProxyUser</a>	A user name to be used to authenticate to the ProxyServer proxy.
<a href="#">ProxyPassword</a>	A password to be used to authenticate to the ProxyServer proxy.
<a href="#">ProxySSLType</a>	The SSL type to use when connecting to the ProxyServer proxy.
<a href="#">ProxyExceptions</a>	A semicolon separated list of destination hostnames or IPs that are exempt from connecting through the ProxyServer .

## Logging

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

## Schema

Property	Description
<a href="#">Location</a>	A path to the directory that contains the schema files defining tables, views, and stored procedures.
<a href="#">Schema</a>	The type of schema to use.

## Miscellaneous

Property	Description
<a href="#">CalculatedDataType</a>	The data type to be used for calculated fields.

<a href="#">ContinueOnError</a>	Indicates whether or not to continue updating items in a batch after an error.
<a href="#">CreateIDColumns</a>	Indicates whether or not to create supplemental ID columns for SharePoint columns that use values from information stored in other Lists.
<a href="#">FolderOption</a>	An option to determine how to display folders in results. Enter either FilesOnly, FilesAndFolders, Recursive, or RecursiveAll.
<a href="#">IncludeLookupColumns</a>	This option controls whether the driver returns the lookup columns defined on a table.
<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 Microsoft SharePoint.
<a href="#">Readonly</a>	You can use this property to enforce read-only access to Microsoft SharePoint from the provider.
<a href="#">ShowHiddenColumns</a>	Boolean determining if hidden columns should be shown or not. If false, all hidden columns will be removed from the column listing.
<a href="#">ShowPredefinedColumns</a>	Boolean determining if predefined columns should be shown or not. If false, all columns derived from a base type will be removed from the column listing.
<a href="#">ShowVersionViews</a>	Indicate whether to display the view of list versions. Such as ListA_Versions.
<a href="#">STSURL</a>	The URL of the security token service (STS) when using single sign-on (SSO).
<a href="#">Timeout</a>	The value in seconds until the timeout error is thrown,

	canceling the operation.
<a href="#">UseDisplayNames</a>	Boolean determining if the display names for the columns should be used instead of the API names.
<a href="#">UseEntityTypeName</a>	Boolean determining if the table name should be EntityTypeName instead of the title in the REST schema.
<a href="#">UseNTLMV1</a>	Determines whether the driver will attempt to connect with NTLMv1 or NTLMv2 (default).
<a href="#">UserDefinedViews</a>	A filepath pointing to the JSON configuration file containing your custom views.
<a href="#">UseSimpleNames</a>	Boolean determining if simple names should be used for tables and columns.

## Authentication

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

Property	Description
<a href="#">AuthScheme</a>	The scheme used for authenticating to SharePoint.
<a href="#">URL</a>	The base URL for the site.
<a href="#">SharePointEdition</a>	The edition of SharePoint being used. Set either SharePoint Online or SharePoint On-Premise.
<a href="#">User</a>	The SharePoint user account used to authenticate.
<a href="#">Password</a>	The password used to authenticate the user.

## AuthScheme

The scheme used for authenticating to SharePoint.

## Possible Values

AzureAD, AzurePassword, OAuthJWT, AzureMSI, PingFederate, ADFS, OneLogin, Okta, OAuth, SharePointOAuth, NTLM, Basic, Negotiate, None

## Data Type

string

## Default Value

"NTLM"

## Remarks

If authenticating to Sharepoint On-Premise instance, together with [Password](#) and [User](#), this field is used to authenticate against the server. NTLM is the default option. Use the following options to select your authentication scheme:

- NTLM: Set this to use your Windows credentials for authentication.
- Negotiate: If [AuthScheme](#) is set to Negotiate, the adapter will Negotiate an authentication mechanism with the server. Set [AuthScheme](#) to Negotiate if you want to use Kerberos authentication.
- None: Set this to use anonymous authentication; for example, to access a public site.
- Basic: Set this to use HTTP Basic authentication.
- ADFS: Set this to use Single Sign-On authentication with ADFS.

If authenticating to Sharepoint Online, OAuth is the default option. Use the following options to select your authentication scheme:

- AzureAD: Set this to perform Azure Active Directory OAuth Authentication.
- AzurePassword: Set this to authenticate using OAuth with Password GrantType.
- OAuthJWT: If [AuthScheme](#) is set to OAuthJWT, the adapter will try to connect with a self signed certificate with Azure AD.
- PingFederate: Set this to use Single Sign-On authentication with PingFederate.

- ADFS: Set this to use Single Sign-On authentication with ADFS.
- OneLogin: Set this to use Single Sign-On authentication with OneLogin.
- OKTA: Set this to use Single Sign-On authentication with OKTA.
- OAuth: Set this to use OAuth 2 authentication with GrantType CODE or PASSWORD.
- AzureMSI: Set this to automatically obtain Managed Service Identity credentials when running on an Azure VM.
- SharePointOAuth: Set this to use OAuth authentication with Sharepoint APP for Client Credentials.

## URL

The base URL for the site.

## Data Type

string

## Default Value

""

## Remarks

The following are examples of valid URLs:

- http://server/SharePoint/
- http://server/Sites/mysite/
- http://server:90/

The provider will use URL to derive URLs for other calls to the server.

## SharePointEdition

The edition of SharePoint being used. Set either SharePoint Online or SharePoint On-Premise.

## Possible Values

SharePoint Online, SharePoint OnPremise

## Data Type

string

## Default Value

"SharePoint OnPremise"

## Remarks

The edition of SharePoint being used. Set either SharePoint Online or SharePoint On-Premise.

## User

The SharePoint user account used to authenticate.

## Data Type

string

## Default Value

""

## Remarks

Together with [Password](#), this field is used to authenticate against the SharePoint server.

For SharePoint On-Premise, User should include the domain and will look similar to the following: DOMAIN\Username.

For SharePoint Online, User will look similar to the following:  
username@domain.onmicrosoft.com.

## Password

The password used to authenticate the user.

## Data Type

string

## Default Value

""

## Remarks

The [User](#) and Password are together used to authenticate with the server.

# Azure Authentication

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

---

Property	Description
<a href="#">AzureTenant</a>	The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.
<a href="#">AzureEnvironment</a>	The Azure Environment to use when establishing a connection.

---

## AzureTenant

The Microsoft Online tenant being used to access data. If not specified, your default tenant will be used.

## Data Type

string



## Default Value

""

## Remarks

The Microsoft Online tenant being used to access data. For instance, contoso.onmicrosoft.com. Alternatively, specify the tenant Id. This value is the directory Id in the Azure Portal > Azure Active Directory > Properties.

Typically it is not necessary to specify the Tenant. This can be automatically determined by Microsoft when using the [OAuthGrantType](#) set to CODE (default). However, it may fail in the case that the user belongs to multiple tenants. For instance, if an Admin of domain A invites a user of domain B to be a guest user. The user will now belong to both tenants. It is a good practice to specify the Tenant, although in general things should normally work without having to specify it.

The [AzureTenant](#) is required when setting [OAuthGrantType](#) to CLIENT. When using client credentials, there is no user context. The credentials are taken from the context of the app itself. While Microsoft still allows client credentials to be obtained without specifying which Tenant, it has a much lower probability of picking the specific tenant you want to work with. For this reason, we require [AzureTenant](#) to be explicitly stated for all client credentials connections to ensure you get credentials that are applicable for the domain you intend to connect to.

## AzureEnvironment

The Azure Environment to use when establishing a connection.

## Possible Values

GLOBAL, CHINA, USGOVT, USGOVTDOD

## Data Type

string

## Default Value

"GLOBAL"

## Remarks

In most cases, leaving the environment set to global will work. However, if your Azure Account has been added to a different environment, the AzureEnvironment may be used to specify which environment. The available values are GLOBAL, CHINA, USGOVT, USGOVTDOD.

## SSO

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

Property	Description
<a href="#">SSOLoginURL</a>	The identity provider's login URL.
<a href="#">SSODomain</a>	The domain of the user when using single sign-on (SSO).
<a href="#">SSOProperties</a>	Additional properties required to connect to the identity provider in a semicolon-separated list.

## SSOLoginURL

The identity provider's login URL.

### Data Type

string

### Default Value

""

## Remarks

The identity provider's login URL.

## SSODomain

The domain of the user when using single sign-on (SSO).

### Data Type

string

### Default Value

""

### Remarks

This property is only applicable when using single sign-on ([AuthScheme](#) is configured to use an SSO authentication scheme) and if the domain of the [User](#) (e.g. user@mydomain.com) is different than the domain configured within the SSO service (e.g. user@myssodomain.com).

This property may be required when using the ADFS, OneLogin, or OKTA SSO services.

## SSOProperties

Additional properties required to connect to the identity provider in a semicolon-separated list.

### Data Type

string

### Default Value

""

### Remarks

Additional properties required to connect to the identity provider in a semicolon-separated list. is used in conjunction with the [SSOLoginURL](#).

SSO configuration is discussed further in .

## 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="#">CallbackURL</a>	The OAuth callback URL to return to when authenticating. This value must match the callback URL you specify in your Add-In settings.
<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="#">Scope</a>	The scope used for the OAuth flow to access data from the Application.
<a href="#">OAuthGrantType</a>	The grant type for the OAuth flow.
<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.

---

**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.

## CallbackURL

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

## OAuthSettingsLocation

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

value starting with memory://.

## Data Type

string

## Default Value

"%APPDATA%\CDData\SharePoint Data Provider\OAuthSettings.txt"

## Remarks

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

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

If left unspecified, the default location is "%APPDATA%\CDData\SharePoint 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

## Scope



The scope used for the OAuth flow to access data from the Application.

## Possible Values

NONE, AllSites.Manage, AllSites.Read, AllSites.Write, .default

## Data Type

string

## Default Value

"NONE"

## Remarks

The scope used for the OAuth flow to access data from the Application.

**Note:** If not specified or left to 'NONE', the adapter sets the default scope dependant upon [OAuthGrantType](#). Defaults to 'AllSites.Manage' unless using JWT OAuth, Azure Service Principal, or CLIENT grant type. The available values are:

- AllSites.Read: Enables reading custom lists
- AllSites.Write: Enables reading & writing custom lists
- AllSites.Manage: Enables reading, writing and creating custom lists
- .default: Requests application permissions without a user context. All the application permissions that have been granted for that web API are included in the retrieved [OAuthAccessToken](#).

## OAuthGrantType

The grant type for the OAuth flow.

## Possible Values

CODE, PASSWORD, SAML\_1, CLIENT

## Data Type

string

## Default Value

"CODE"

## Remarks

The following options are available: CODE,PASSWORD,SAML\_1,CLIENT

## OAuthVerifier

The verifier code returned from the OAuth authorization URL.

## Data Type

string

## Default Value

""

## Remarks

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

## Authentication on Headless Machines

See to obtain the OAuthVerifier value.

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

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

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

## OAuthRefreshToken

The OAuth refresh token for the corresponding OAuth access token.

### Data Type

string

### Default Value

""

### Remarks

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

## OAuthExpiresIn

The lifetime in seconds of the OAuth AccessToken.

### Data Type

string

### Default Value

""

### Remarks

Pair with OAuthTokenTimestamp to determine when the AccessToken will expire.

## OAuthTokenTimestamp

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

## Data Type

string

## Default Value

""

## Remarks

Pair with OAuthExpiresIn to determine when the AccessToken will expire.

# JWT OAuth

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

---

Property	Description
<a href="#">OAuthJWTCert</a>	The JWT Certificate store.
<a href="#">OAuthJWTCertType</a>	The type of key store containing the JWT Certificate.
<a href="#">OAuthJWTIssuer</a>	The issuer of the Java Web Token.

## OAuthJWTCert

The JWT Certificate store.

## Data Type

string

## Default Value

""

## Remarks

The name of the certificate store for the client certificate.

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

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

Designations of certificate stores are platform-dependent.

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

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

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

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

## OAuthJWTCertType

The type of key store containing the JWT Certificate.

### Possible Values

USER, MACHINE, PFXFILE, PFXBLOB, JKSFILE, JKSFILE, PEMKEY\_FILE, PEMKEY\_BLOB, PUBLIC\_KEY\_FILE, PUBLIC\_KEY\_BLOB, SSHPUBLIC\_KEY\_FILE, SSHPUBLIC\_KEY\_BLOB,

P7BFILE, PPKFILE, XMLFILE, XMLBLOB

## Data Type

string

## Default Value

"USER"

## Remarks

This property can take one of the following values:

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

	private key and an optional certificate.
PUBLIC_KEY_FILE	The certificate store is the name of a file that contains a PEM- or DER-encoded public key certificate.
PUBLIC_KEY_BLOB	The certificate store is a string (base-64-encoded) that contains a PEM- or DER-encoded public key certificate.
SSHPUBLIC_KEY_FILE	The certificate store is the name of a file that contains an SSH-style public key.
SSHPUBLIC_KEY_BLOB	The certificate store is a string (base-64-encoded) that contains an SSH-style public key.
P7BFILE	The certificate store is the name of a PKCS7 file containing certificates.
PPKFILE	The certificate store is the name of a file that contains a PPK (PuTTY Private Key).
XMLFILE	The certificate store is the name of a file that contains a certificate in XML format.
XMLBLOB	The certificate store is a string that contains a certificate in XML format.

## OAuthJWTIssuer

The issuer of the Java Web Token.

### Data Type

string

### Default Value

""

## Remarks

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

## Kerberos

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

Property	Description
<a href="#">KerberosKDC</a>	The Kerberos Key Distribution Center (KDC) service used to authenticate the user.
<a href="#">KerberosRealm</a>	The Kerberos Realm used to authenticate the user.
<a href="#">KerberosSPN</a>	The service principal name (SPN) for the Kerberos Domain Controller.
<a href="#">KerberosKeytabFile</a>	The Keytab file containing your pairs of Kerberos principals and encrypted keys.
<a href="#">KerberosServiceRealm</a>	The Kerberos realm of the service.
<a href="#">KerberosServiceKDC</a>	The Kerberos KDC of the service.
<a href="#">KerberosTicketCache</a>	The full file path to an MIT Kerberos credential cache file.

## KerberosKDC

The Kerberos Key Distribution Center (KDC) service used to authenticate the user.

## Data Type

string



## Default Value

""

## Remarks

The Kerberos properties are used when using SPNEGO or Windows Authentication. The adapter will request session tickets and temporary session keys from the Kerberos KDC service. The Kerberos KDC service is conventionally colocated with the domain controller.

If Kerberos KDC is not specified, the adapter will attempt to detect these properties automatically from the following locations:

- **KRB5 Config File (krb5.ini/krb5.conf):** If the KRB5\_CONFIG environment variable is set and the file exists, the adapter will obtain the KDC from the specified file. Otherwise, it will attempt to read from the default MIT location based on the OS: *C:\ProgramData\MIT\Kerberos5\krb5.ini* (Windows) or */etc/krb5.conf* (Linux).
- **Java System Properties:** Using the system properties *java.security.krb5.realm* and *java.security.krb5.kdc*.
- **Domain Name and Host:** If the Kerberos Realm and Kerberos KDC could not be inferred from another location, the adapter will infer them from the configured domain name and host.

**Note:** Windows authentication is supported in JRE 1.6 and above only.

The Kerberos properties are used when using SPNEGO or Windows Authentication. The adapter will request session tickets and temporary session keys from the Kerberos KDC service. The Kerberos KDC service is conventionally colocated with the domain controller.

If Kerberos KDC is not specified, the adapter will attempt to detect these properties automatically from the following locations:

- **KRB5 Config File (krb5.ini/krb5.conf):** If the KRB5\_CONFIG environment variable is set and the file exists, the adapter will obtain the KDC from the specified file. Otherwise, it will attempt to read from the default MIT location based on the OS: *C:\ProgramData\MIT\Kerberos5\krb5.ini* (Windows) or */etc/krb5.conf* (Linux).
- **Java System Properties:** Using the system properties *java.security.krb5.realm* and *java.security.krb5.kdc*.
- **Domain Name and Host:** If the Kerberos Realm and Kerberos KDC could not be inferred from another location, the adapter will infer them from the configured domain name and host.

**Note:** Windows authentication is supported in JRE 1.6 and above only.

## KerberosRealm

The Kerberos Realm used to authenticate the user.

### Data Type

string

### Default Value

""

### Remarks

The Kerberos properties are used when using SPNEGO or Windows Authentication. The Kerberos Realm is used to authenticate the user with the Kerberos Key Distribution Service (KDC). The Kerberos Realm can be configured by an administrator to be any string, but conventionally it is based on the domain name.

If Kerberos Realm is not specified, the adapter will attempt to detect these properties automatically from the following locations:

- **KRB5 Config File (krb5.ini/krb5.conf):** If the KRB5\_CONFIG environment variable is set and the file exists, the adapter will obtain the default realm from the specified file. Otherwise, it will attempt to read from the default MIT location based on the OS: *C:\ProgramData\MIT\Kerberos5\krb5.ini* (Windows) or */etc/krb5.conf* (Linux)
- **Java System Properties:** Using the system properties *java.security.krb5.realm* and *java.security.krb5.kdc*.
- **Domain Name and Host:** If the Kerberos Realm and Kerberos KDC could not be inferred from another location, the adapter will infer them from the user-configured domain name and host. This might work in some Windows environments.

**Note:** Kerberos-based authentication is supported in JRE 1.6 and above only.

## KerberosSPN

The service principal name (SPN) for the Kerberos Domain Controller.

### Data Type

string

## Default Value

""

## Remarks

If the SPN on the Kerberos Domain Controller is not the same as the URL that you are authenticating to, use this property to set the SPN.

If the SPN on the Kerberos Domain Controller is not the same as the URL that you are authenticating to, use this property to set the SPN.

## KerberosKeytabFile

The Keytab file containing your pairs of Kerberos principals and encrypted keys.

## Data Type

string

## Default Value

""

## Remarks

The Keytab file containing your pairs of Kerberos principals and encrypted keys.

## KerberosServiceRealm

The Kerberos realm of the service.

## Data Type

string

## Default Value

""

## Remarks

The KerberosServiceRealm is the specify the service Kerberos realm when using cross-realm Kerberos authentication.

In most cases, a single realm and KDC machine are used to perform the Kerberos authentication and this property is not required.

This property is available for complex setups where a different realm and KDC machine are used to obtain an authentication ticket (AS request) and a service ticket (TGS request).

## KerberosServiceKDC

The Kerberos KDC of the service.

## Data Type

string

## Default Value

""

## Remarks

The KerberosServiceKDC is used to specify the service Kerberos KDC when using cross-realm Kerberos authentication.

In most cases, a single realm and KDC machine are used to perform the Kerberos authentication and this property is not required.

This property is available for complex setups where a different realm and KDC machine are used to obtain an authentication ticket (AS request) and a service ticket (TGS request).

## KerberosTicketCache

The full file path to an MIT Kerberos credential cache file.

## Data Type

string

## Default Value

""

## Remarks

This property can be set if you wish to use a credential cache file that was created using the MIT Kerberos Ticket Manager or kinit command.

## 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="#">SSLClientCert</a>	The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).
<a href="#">SSLClientCertType</a>	The type of key store containing the TLS/SSL client certificate.
<a href="#">SSLClientCertPassword</a>	The password for the TLS/SSL client certificate.
<a href="#">SSLClientCertSubject</a>	The subject of the TLS/SSL client certificate.
<a href="#">SSLServerCert</a>	The certificate to be accepted from the server when connecting using TLS/SSL.

## SSLClientCert

The TLS/SSL client certificate store for SSL Client Authentication (2-way SSL).

## Data Type

string

## Default Value

""

## Remarks

The name of the certificate store for the client certificate.

The [SSLClientCertType](#) field specifies the type of the certificate store specified by [SSLClientCert](#). If the store is password protected, specify the password in [SSLClientCertPassword](#).

[SSLClientCert](#) is used in conjunction with the [SSLClientCertSubject](#) field in order to specify client certificates. If [SSLClientCert](#) has a value, and [SSLClientCertSubject](#) is set, a search for a certificate is initiated. See [SSLClientCertSubject](#) for more information.

Designations of certificate stores are platform-dependent.

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

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

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

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

## SSLClientCertType

The type of key store containing the TLS/SSL client certificate.

## Possible Values

USER, MACHINE, PFXFILE, PFXBLOB, JKSFILE, JKSLOB, PEMKEY\_FILE, PEMKEY\_BLOB, PUBLIC\_KEY\_FILE, PUBLIC\_KEY\_BLOB, SSHPUBLIC\_KEY\_FILE, SSHPUBLIC\_KEY\_BLOB, P7BFILE, PPKFILE, XMLFILE, XMLBLOB

## Data Type

string

## Default Value

"USER"

## Remarks

This property can take one of the following values:

USER - default	For Windows, this specifies that the certificate store is a certificate store owned by the current user. Note that this store type is not available in Java.
MACHINE	For Windows, this specifies that the certificate store is a machine store. Note that this store type is not available in Java.
PFXFILE	The certificate store is the name of a PFX (PKCS12) file containing certificates.
PFXBLOB	The certificate store is a string (base-64-encoded) representing a certificate store in PFX (PKCS12) format.
JKSFILE	The certificate store is the name of a Java key store (JKS) file containing certificates. Note that this store type is only available in Java.
JKSBLOB	The certificate store is a string (base-64-encoded) representing a certificate store in JKS format. Note that this store type is only available in Java.
PEMKEY_FILE	The certificate store is the name of a PEM-encoded file that

	contains a private key and an optional certificate.
PEMKEY_BLOB	The certificate store is a string (base64-encoded) that contains a private key and an optional certificate.
PUBLIC_KEY_FILE	The certificate store is the name of a file that contains a PEM- or DER-encoded public key certificate.
PUBLIC_KEY_BLOB	The certificate store is a string (base-64-encoded) that contains a PEM- or DER-encoded public key certificate.
SSHPUBLIC_KEY_FILE	The certificate store is the name of a file that contains an SSH-style public key.
SSHPUBLIC_KEY_BLOB	The certificate store is a string (base-64-encoded) that contains an SSH-style public key.
P7BFILE	The certificate store is the name of a PKCS7 file containing certificates.
PPKFILE	The certificate store is the name of a file that contains a PuTTY Private Key (PPK).
XMLFILE	The certificate store is the name of a file that contains a certificate in XML format.
XMLBLOB	The certificate store is a string that contains a certificate in XML format.

## SSLClientCertPassword

The password for the TLS/SSL client certificate.

### Data Type

string

### Default Value

""



## Remarks

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

## SSLClientCertSubject

The subject of the TLS/SSL client certificate.

## Data Type

string

## Default Value

""

## Remarks

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

If an exact match is not found, the store is searched for subjects containing the value of the property. If a match is still not found, the property is set to an empty string, and no certificate is selected.

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

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

Field	Meaning
CN	Common Name. This is commonly a host name like www.server.com.
O	Organization
OU	Organizational Unit
L	Locality

S	State
C	Country
E	Email Address

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

## 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----- MIICHtCC Ae4CAQAwDQYJKoZIh v.....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)

34a929226ae0819f2ec14b4a3d904f801c

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 SharePoint 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
Location	A path to the directory that contains the schema files defining tables, views, and stored procedures.
Schema	The type of schema to use.

---

## Location

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

## Data Type

string

## Default Value

"%APPDATA%\CDData\SharePoint 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.

**Note:** Given that this adapter supports multiple schemas, the structure for SharePoint custom schema files is as follows:

- Each schema is given a folder corresponding to that schema name.
- These schema folders are contained in a parent folder.
- The **parent folder** should be set as the Location, not an individual schema's folder.

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

## Schema

The type of schema to use.

### Possible Values

REST, SOAP

### Data Type

string

### Default Value

"SOAP"

### Remarks

The schemas available are REST (to use SharePoint REST API) and SOAP (to use Sharepoint SOAP API).

## 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="#">CalculatedDataType</a>	The data type to be used for calculated fields.
<a href="#">ContinueOnError</a>	Indicates whether or not to continue updating items in a batch after an error.
<a href="#">CreateIDColumns</a>	Indicates whether or not to create supplemental ID columns for SharePoint columns that use values from information stored in other Lists.
<a href="#">FolderOption</a>	An option to determine how to display folders in results. Enter either FilesOnly, FilesAndFolders, Recursive, or RecursiveAll.
<a href="#">IncludeLookupColumns</a>	This option controls whether the driver returns the lookup columns defined on a table.
<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 Microsoft SharePoint.
<a href="#">Readonly</a>	You can use this property to enforce read-only access to Microsoft SharePoint from the provider.
<a href="#">ShowHiddenColumns</a>	Boolean determining if hidden columns should be shown or not. If false, all hidden columns will be removed from the column listing.
<a href="#">ShowPredefinedColumns</a>	Boolean determining if predefined columns should be shown or not. If false, all columns derived from a base type will be removed from the column listing.
<a href="#">ShowVersionViews</a>	Indicate whether to display the view of list versions. Such as ListA_Versions.



<b>STSURL</b>	The URL of the security token service (STS) when using single sign-on (SSO).
<b>Timeout</b>	The value in seconds until the timeout error is thrown, canceling the operation.
<b>UseDisplayNames</b>	Boolean determining if the display names for the columns should be used instead of the API names.
<b>UseEntityTypeName</b>	Boolean determining if the table name should be EntityTypeName instead of the title in the REST schema.
<b>UseNTLMV1</b>	Determines whether the driver will attempt to connect with NTLMv1 or NTLMv2 (default).
<b>UserDefinedViews</b>	A filepath pointing to the JSON configuration file containing your custom views.
<b>UseSimpleNames</b>	Boolean determining if simple names should be used for tables and columns.

## CalculatedDataType

The data type to be used for calculated fields.

### Possible Values

Calculated, Boolean, Date, DateTime, Double, Integer, String

### Data Type

string

### Default Value

"Calculated"

## Remarks

The data type to be used for calculated fields. By default (Calculated), the data type is determined by the type of calculated field in SharePoint. However, in some cases these calculated fields may return values that are not appropriate for the specified type. In these instances, you may wish to set the Calculated Data Type to the appropriate type.

## ContinueOnError

Indicates whether or not to continue updating items in a batch after an error.

### Data Type

bool

### Default Value

true

## Remarks

If this property is set to True (default), the adapter will continue adding, updating, or deleting items when an error is encountered on one of the items. When set to False, the adapter will stop adding, updating, or deleting items after an error is encountered (entries preceeding the problematic entry will still be added, updated, or deleted).

## CreateIDColumns

Indicates whether or not to create supplemental ID columns for SharePoint columns that use values from information stored in other Lists.

### Data Type

bool

### Default Value

true

## Remarks

Indicates whether or not to create supplemental ID columns for SharePoint columns that use values from information stored in other Lists (like "Lookup" or "Person or Group" columns). The ID column that is created will contain the related entry's ID (in the context of its original List). If set to false, the ID columns will not be created, the ID will be ignored, and only the value of the referenced column will be returned.

## FolderOption

An option to determine how to display folders in results. Enter either FilesOnly, FilesAndFolders, Recursive, or RecursiveAll.

### Possible Values

FilesOnly, FilesAndFolders, RecursiveAll

### Data Type

string

### Default Value

"RecursiveAll"

## Remarks

An option to determine how to display folders in results. FilesOnly will display only files in specified lists or libraries. FilesAndFolders will display files and folders in the specified list. RecursiveAll will display files in the specified list and all subfolders.

## IncludeLookupColumns

This option controls whether the driver returns the lookup columns defined on a table.

### Data Type

bool

## Default Value

true

## Remarks

This option controls whether the driver returns the lookup columns defined on a table. The SharePoint server may reject the request if too many lookup columns are included in a single query.

## 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 Microsoft SharePoint.

## Data Type

int

## Default Value

1000

## Remarks

The Pagesize property affects the maximum number of results to return per page from SharePoint. Setting a higher value may result in better performance at the cost of

additional memory allocated per page consumed.

## Readonly

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

## ShowHiddenColumns

Boolean determining if hidden columns should be shown or not. If false, all hidden columns will be removed from the column listing.

### Data Type

bool

### Default Value

false

### Remarks

Boolean determining if hidden columns should be shown or not. If false, all hidden columns will be removed from the column listing.

## ShowPredefinedColumns

Boolean determining if predefined columns should be shown or not. If false, all columns derived from a base type will be removed from the column listing.

### Data Type

bool

### Default Value

true

### Remarks

Boolean determining if predefined columns should be shown or not. If false, all columns derived from a base type will be removed from the column listing. These columns are normally system columns such as CreatedBy and Author. But, predefined columns may also include common columns such as Title.

## ShowVersionViews

Indicate whether to display the view of list versions. Such as ListA\_Versions.

### Data Type

bool

### Default Value

false

### Remarks

Indicate whether to display the view of list versions. Such as ListA\_Versions.

## STSURL

The URL of the security token service (STS) when using single sign-on (SSO).

## Data Type

string

## Default Value

""

## Remarks

The URL of the security token service (STS) when using single sign-on (SSO). This rarely needs to be set explicitly.

## 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.

## UseDisplayNames

Boolean determining if the display names for the columns should be used instead of the API names.



## Data Type

bool

## Default Value

true

## Remarks

Boolean determining if the display names for the columns should be used instead of the API names.

## UseEntityTypeName

Boolean determining if the table name should be EntityTypeName instead of the title in the REST schema.

## Data Type

bool

## Default Value

false

## Remarks

Boolean determining if the table name should be EntityTypeName instead of the title in the REST schema.

## UseNTLMV1

Determines whether the driver will attempt to connect with NTLMv1 or NTLMv2 (default).

## Data Type

bool

## Default Value

false

## Remarks

Determines whether the driver will attempt to connect with NTLMv1 or NTLMv2 (default).

## 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 Calendar 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"

```

## UseSimpleNames

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

### Data Type

bool

### Default Value

false

### Remarks

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

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

# 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.