



How to integrate Denodo with SharePoint Online

Revision 20221229

NOTE

This document is confidential and proprietary of **Denodo Technologies**. No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2023
Denodo Technologies Proprietary and Confidential

CONTENTS

1 CONFIGURING THE APPLICATION IN AZURE.....	4
2 CONNECTING TO SHAREPOINT USING ODATA.....	8
2.1 OBTAIN THE OAUTH TOKENS.....	8
2.2 USING DENODO TEMPLATES FOR SHAREPOINT.....	10
2.3 USING THE ODATA 2 CUSTOM WRAPPER.....	10
3 CONNECTING TO SHAREPOINT USING EXCEL, DELIMITED FILE, JSON AND XML DATA SOURCE.....	14
3.1 GET THE OAUTH TOKENS USING THE OAUTH CREDENTIALS WIZARD	19
4 APPENDIX I: AUTHENTICATING WITH NTLM.....	22
5 APPENDIX II: CONNECTING TO LISTS AS RSS FEEDS.....	24
6 REFERENCES.....	25

This document describes all the possible ways to access SharePoint Online from Denodo Virtual DataPort:

- The fastest way that can be enough for many users is to use the **Denodo Templates for SharePoint**. These templates allow accessing information like Lists, Items, File and Folder Metadata of any SharePoint site. If you want to use the Denodo templates for SharePoint take a look at the [Denodo Templates for SharePoint - Quick User Guide](#). Note that, even using the templates, you have to register and configure the app in Azure as explained in this document. In case you do not want to use the templates or require information not available through them, check the other options.
- **OData2 Custom Wrapper**: to access SharePoint OData entities.
- **Excel, JSON, XML and Delimited File Data Sources**: to access SharePoint's REST API to retrieve resources like Files, Lists or Folders.
- Also, you can connect to SharePoint online through **RSS**.

The following table summarizes the possible ways to access SharePoint:

NEED	CONNECTION METHOD	AUTHENTICATION
Access metadata of Files, Folders, Lists or Items; and basic OData entities on any	Denodo Templates for Sharepoint	OAuth 2.0

Sharepoint site.		
Access other OData entities.	OData2 Custom Wrapper	
Retrieve metadata of Files, Folders, Lists or Items (without using the Denodo Templates for SharePoint).	JSON / XML data sources	OAuth 2.0
Read Excel files.	Excel data source	OAuth 2.0
Read CSV files.	Delimited File data source	OAuth 2.0
Read Sharepoint Lists.	RSS	OAuth 2.0

***NOTE:** NTLM authentication is only an option for SharePoint Server, not for SharePoint Online.

In all cases it is necessary to register and configure an app in Azure as described below.

1 CONFIGURING THE APPLICATION IN AZURE

First of all, it is necessary to register an App, provide the correct API permissions and generate a client secret in the Azure Portal in order to allow the app (in this case Denodo) to access SharePoint.

In order to register an app in Azure, follow these steps:

1. Login to the Azure portal
2. Navigate https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps to
3. Press the button New Registration
4. Give a name for the app
5. Fill in the Support Account Types field. The type of the application support depends on your requirement. For the purpose of this documentation we are using Accounts in this organizational directory only (Default Directory only - Single tenant).
6. Enter Redirect URL value. Due to a SharePoint limitation, the value you set at this time will be the only one you can use as a redirect URL for this app. Azure supports adding more redirect URLs to an app but Sharepoint only supports one. Although Azure allows its later modification, this modification will not be synchronized with SharePoint and will not work. The default value is <http://localhost:9090/oauth/2.0/redirectURL.jsp>. Note that http only can be used with localhost, if you have HTTPS enabled in your Denodo Virtual DataPort, then use your hostname instead of localhost: <https://<hostname>:9443/oauth/2.0/redirectURL.jsp>.
7. Press the button "Register".

Register an application

*** Name**
The user-facing display name for this application (this can be changed later).

Supported account types
Who can use this application or access this API?

Accounts in this organizational directory only (Denodo Technologies Inc. only - Single tenant)

Accounts in any organizational directory (Any Azure AD directory - Multitenant)

Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

Once the app is registered we will need to grant permissions for clients to access SharePoint Online using the app:


- In the newly registered application, click on Integration assistant (on the left side menu) and select Daemon for What application types are you building and select Evaluate my registration.
- Once the evaluation is done, the Recommended configurations are shown.

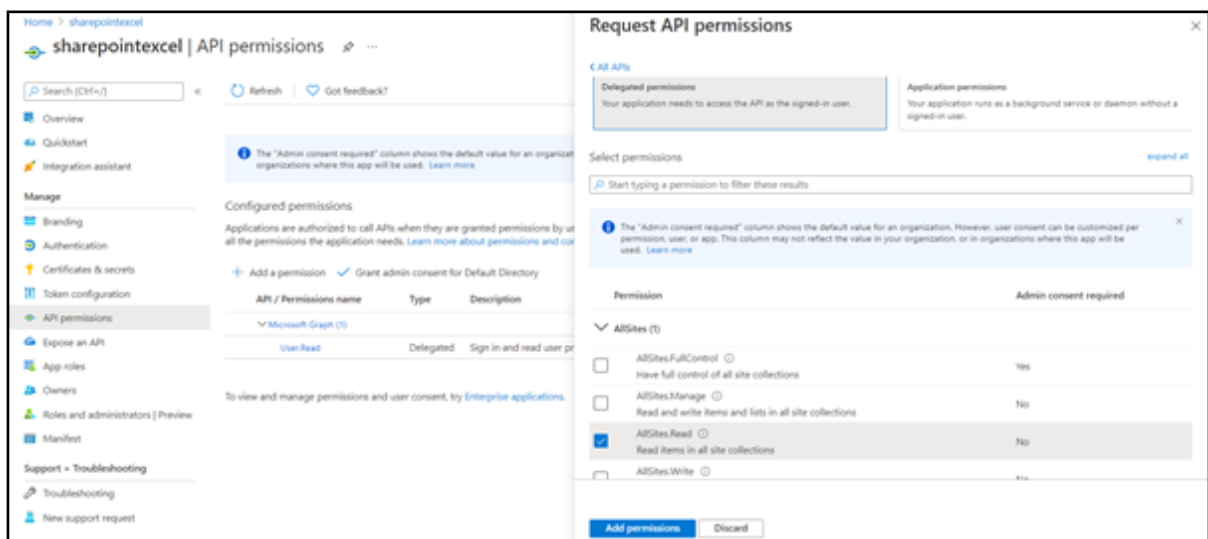
Item	Status
Configure API permissions.	Action required ***
Configure a valid credential.	Action required ***

There are 2 needed configurations that have to be performed to allow the Denodo

Platform to access the data in SharePoint: API permissions and client secret generation.

In order to configure API permissions follow the following steps:

- Click on the  icon on Configure API permissions and select Go to page.
- Select Add Permissions. In the Request API permissions pop-up select SharePoint.
- Select Delegated Permissions and select AllSites.Read as the permissions.



After the API permissions are set we have to generate a new client secret key. In order to generate the key:

- In the Integration assistance section, select Configure a valid credential.
- In the Certificates & secrets page select the option New client secret.
- After generating the Client secret, copy the **Value**.

Certificates & secrets ✕

[Got feedback?](#)

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

i Application registration certificates, secrets and federated credentials can be found in the tabs below. ✕

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[+ New client secret](#)

Description	Expires	Value	Secret ID
client_secret	10/1/2023	[REDACTED]	[REDACTED]

- After having followed those sections, you should have gathered these values:
- **Application (client) ID** which will be the **Client Id** in the data sources configuration.
 - **Value of the Client Secret** from step “Client secret key generation”
 - The **Directory (tenant) ID** which will be the **tenant_id** needed later.

^ Essentials

Display name Denodo Platform	Client credentials 0 certificate, 1 secret
Application (client) ID e: [REDACTED]	Redirect URIs 1 web, 0 spa, 0 public client
Object ID b: [REDACTED]	Application ID URI Add an Application ID URI
Directory (tenant) ID 8: [REDACTED]	Managed application in local directory Denodo Platform
Supported account types My organization only	

2 CONNECTING TO SHAREPOINT USING ODATA

2.1 OBTAIN THE OAUTH TOKENS

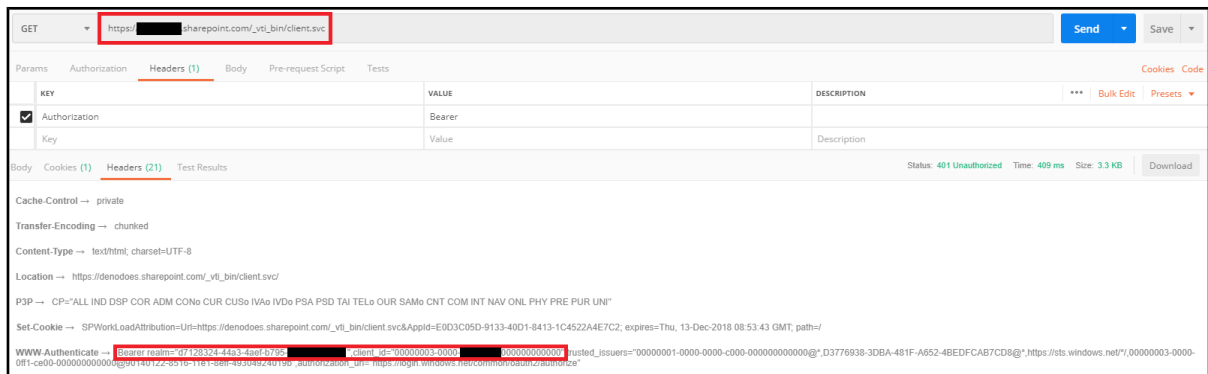
NOTE: If you are not going to use the OAuth authentication you can skip this step.

As you have obtained the client id, client secret and tenant id, you need to obtain the access token and refresh token by using those values.

2.1.1 Get the Realm of the site

Realm is a constant GUID for a site. In order to retrieve it, a tool such as Postman can be used. To obtain the Realm, it is necessary to carry out the following steps:

- Make a GET request like this:
 - `https://<tenant_name>.sharepoint.com/_vti_bin/ListData.svc`
 - Header:
 - Authorization: Bearer
- Get the Bearer realm component from the WWW-Authenticate response header and save it. This value is the same as the tenant id.
- Get the client_id component from the WWW-Authenticate response header and save it. This value is what later we will call Audience Principal ID.



NOTE: If you are using cookies you might not get back the WWW-Authenticate header with the Bearer realm. Delete the cookies in the browser or use the Postman Interceptor to avoid this issue.

2.1.2 Get the authorization code

Construct an authorization url as follows:

```
https://<tenant_name>.sharepoint.com/_layouts/15/OAuthAuthorize.aspx?
client_id=<client_id>&scope=AllSites.Read&response_type=code&redirect_uri
=<redirect_uri>
```

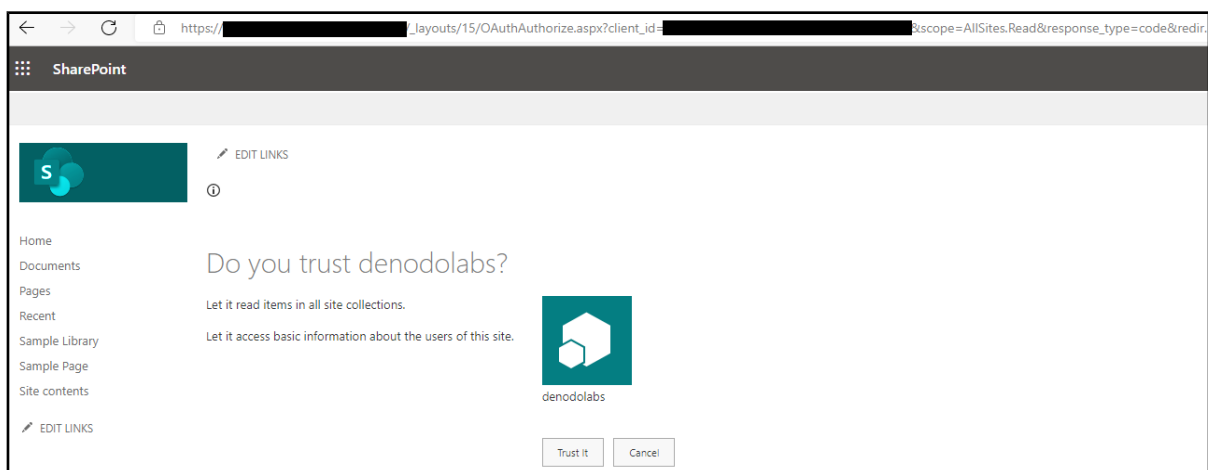

Change the parameters of the URL to fit your connection data. For the `client_id` use the **Application (client) ID** from the application registration in the Azure portal (this is different from the client id obtained in the previous step that will be used as Audience Principal ID).

NOTE: For this step the Azure Portal user will need to have the necessary permissions to consent to grant permissions to the app. In the previous sample URL we are requesting read permissions for all SharePoint sites, this would require [Manage permissions](#) on the resources requested.

To make a request restricted to a specific site a URL like the following could be used requesting read permissions on an individual site:

```
https://<tenant_name>.sharepoint.com/sites/<sitename>/_layouts/15/OAuthAuthorize.aspx?
client_id=<client_id>&scope=Site.Read&response_type=code&redirect_uri=<re
direct_uri>
```

Navigate to the URL from your browser. Login to the site if you have not logged in already. This opens a consent page that prompts the user to grant (or deny) the app the permissions that the app requests.



Once you grant the permission (by clicking trust), the SharePoint Online site asks the Access Control Service (ACS) to create a short-lived (approximately 5 minutes) authorization code unique to this combination of user and app. ACS sends the authorization code to the SharePoint site.

SharePoint Online site redirects the browser back to the redirect URI that was specified when the app was registered. It also includes the authorization code as a query string. The redirect URL is structured like the following:

`http://localhost:9090/oauth/2.0/redirectURL.jsp?code=<authcode>` or
`https://<hostname>:9443/oauth/2.0/redirectURL.jsp?code=<authcode>`

2.1.3 Get the access token and refresh token

Construct the below POST request:

- URL:
 - `https://accounts.accesscontrol.windows.net/<site_realm>/tokens/OAuth/2`
- Header:

- Content-Type = "application/x-www-form-urlencoded"
- Post parameters (in the body of the request):
 - grant_type=authorization_code
 - client_id=<client_id>@<site_realm>
 - client_secret=<client_secret>
 - code=<auth_code>
 - redirect_uri=<redirect_uri>
 - resource=<audience_principal_ID>/<site_host>@<site_realm>
 - Where:
 - <site_realm> is the Bearer realm obtained in the step "Get the Realm of the site".
 - <client_id> is <client id when registering the app>@<site realm from the step "Get the Realm of the site>.
 - <client_secret> is the client_secret obtained when registering the app.
 - <auth_code> is the auth code obtained in step "Get the authorization code".
 - <resource> is <audience principal ID>/<sharepoint domain>@<site realm>.
 - <redirect_uri> is the redirect URL set when registering the app.
 - <audience_principal_ID> is a permanent security principal ID for SharePoint. <audience_principal_ID> is obtained in step "Get the Realm of the site" (the value "client_id" in the response header WWW-Authenticate).

NOTE: all values need to be URL encoded (including the client_secret)

2.2 USING DENODO TEMPLATES FOR SHAREPOINT

If you are using the Denodo Templates for Sharepoint you can skip the following steps and return to the [Denodo Templates for SharePoint - Quick User Guide](#).

2.3 USING THE ODATA 2 CUSTOM WRAPPER

First of all, you need to download the **Denodo OData 2 Custom Wrapper** from the Support Site and import it into the Virtual DataPort Server following the instructions provided in the documentation.

Once the OData Custom Wrapper is imported, create a new Custom Data Source selecting New > Data Source > Custom in the contextual menu.

- On "Select Jars" select the imported extension.

- Fill the "Class name" field with: `com.denodo.connect.odata.wrapper.ODataWrapper`
- Click the **Refresh Input Parameters** to show the parameters related to the wrapper to be configured.

The configuration of these parameters depends on the type of the authentication chosen.

2.3.1 Connecting through OData 2 using OAuth 2.0

The following wrapper parameters need to be configured as follow to define datasource:


Service Endpoint	<code>https://<tenant_name>.sharepoint.com/_vti_bin/ListData.svc</code> Replace <tenant_name> with your actual tenant name
Service Format	JSON or XML-ATOM
Service Version	V2
Use OAuth2	true
Access Token	access_token obtained in the Obtain OAuth tokens section
Refresh Token	refresh_token obtained in the Obtain OAuth tokens section
Client Id	<code><client_id>\@<tenant_id></code> <client_id> and <tenant id> from registering the app in Configuration the application in Azure section
Client Secret	client secret from registering the app in Configuration the application in Azure section
Token Endpoint URL	<code>https://accounts.accesscontrol.windows.net/<tenant_id>/tokens/OAuth/2</code> Replace <tenant_id> with the actual directory tenant id obtained in Configuration the application in Azure section
OAuth Extra Parameters	<code>resource="<audience_principal_id>/<tenant_name>.sharepoint.com\@<tenant_id>"</code> Replace <tenant_id> with the actual directory tenant id obtained in Configuration the application in Azure section and <audience_principal_id> with the value obtained in Obtain OAuth tokens section
Refr. Token Auth. Method	Include the client credentials in the body of the request

Fill in Proxy fields if required.

NOTE: the '@' must be escaped with '\' to prevent it from being confused with an interpolation variable.

This is how it will look like in the OData 2 Custom Wrapper:

Input parameters of the data source

Click to refresh the input parameters of the data source 

Service Endpoint *:

Service Format *: ▼

Service Version: ▼

Pass-through session credentials

User:

Password:

Proxy Host:

Proxy Port:

Proxy User:

Proxy Password:

Use NTLM Authentication

NTLM Domain:

Timeout:

Use OAuth2

Access Token:

Refresh Token:

Client Id:

Client Secret:

Token Endpoint URL:

OAuth Extra Parameters: resource=

Refr. Token Auth. Method: ▼

HTTP Headers:

Save the changes and select the Create base view option. This will display the Edit Wrapper Parameter Values dialog. For example you could use Documents as Entity Collection which will list you the elements you have in that site:

Edit Wrapper Parameter values

Enter values for the following wrapper parameters:

Entity Collection *:

Expand Related Entities

Enable Pagination

Summary Edit Options VQL Execution panel Used by Associations Publish Export Drop

View type: Base

Schema:

Field Name	Field Type	Description
contenttypeid	text	
name	text	
complianceassetid	text	
title	text	

Owner: admin Last modifier: admin

Creation: 19.01.2022 20:28:20 Last modification: 19.01.2022 20:28:20

Swap status: default Cache status: off

Folder: /1 - data sources

Description:

Execute Query Results

Results Execution Trace Query: SELECT * FROM ds_odata_sharepoint LIMIT 150 CONTEXT ('118n'=de_euro, 'cache_wai

Total rows received: 3 (shown 3)

contenttyp...	name	complan...	title	desc...	id	contenttype	created	createdbyid	modified
0x010100...	Invoices_2015.xlsx	<null>	<null>	<null>	3	Document	2021-09-28 09:30:51	10	2021-12-2...
0x010100...	Invoices_2015_tax.xlsx	<null>	<null>	<null>	2	Document	2021-09-28 09:30:51	10	2021-09-2...
0x010100...	Mappe.xlsx	<null>	<null>	<null>	4	Document	2021-12-27 03:19:11	10	2021-12-2...

You can define any available collection name in the Entity Collection textbox .To know the list of available collections, browse the service endpoint URL used in the OData 2 Custom Wrapper datasource configuration

URL	https://<tenant>.sharepoint.com/_vti_bin/ListData.svc/
-----	--

or provide a non-existent collection and the wrapper will list the available ones:

There was an error while creating this base view: Error while executing custom wrapper method 'getSchemaParameters': Entity Collection not found for the requested service. Available Entity Collections are [SolutionGallery, ThemeGallery, Appfiles, Appdata, MasterPageGalleryCompatibleSearchDataTypes, MasterPageGallery, Attachments, SitePages, Events, ListTemplateGallery, Books, MasterPageGalleryTargetControlType, MasterPageGalleryTemplateLevel, SearchConfigList, WebPartGalleryGroup, ComposedLooks, MasterPageGalleryStandalone, WebPartGalleryRecommendationSettings, SharePointHomeOrgLinks, StyleLibrary, WebPartGallery, SampleLibrary, UserInformationList, MasterPageGalleryCompatibleUIVersionS, Documents, SearchConfigListScope, SitePagesSitePageFlags, TaxonomyHiddenList, FormTemplates, EventsCategory, MasterPageGalleryTargetControlTypeSearch]

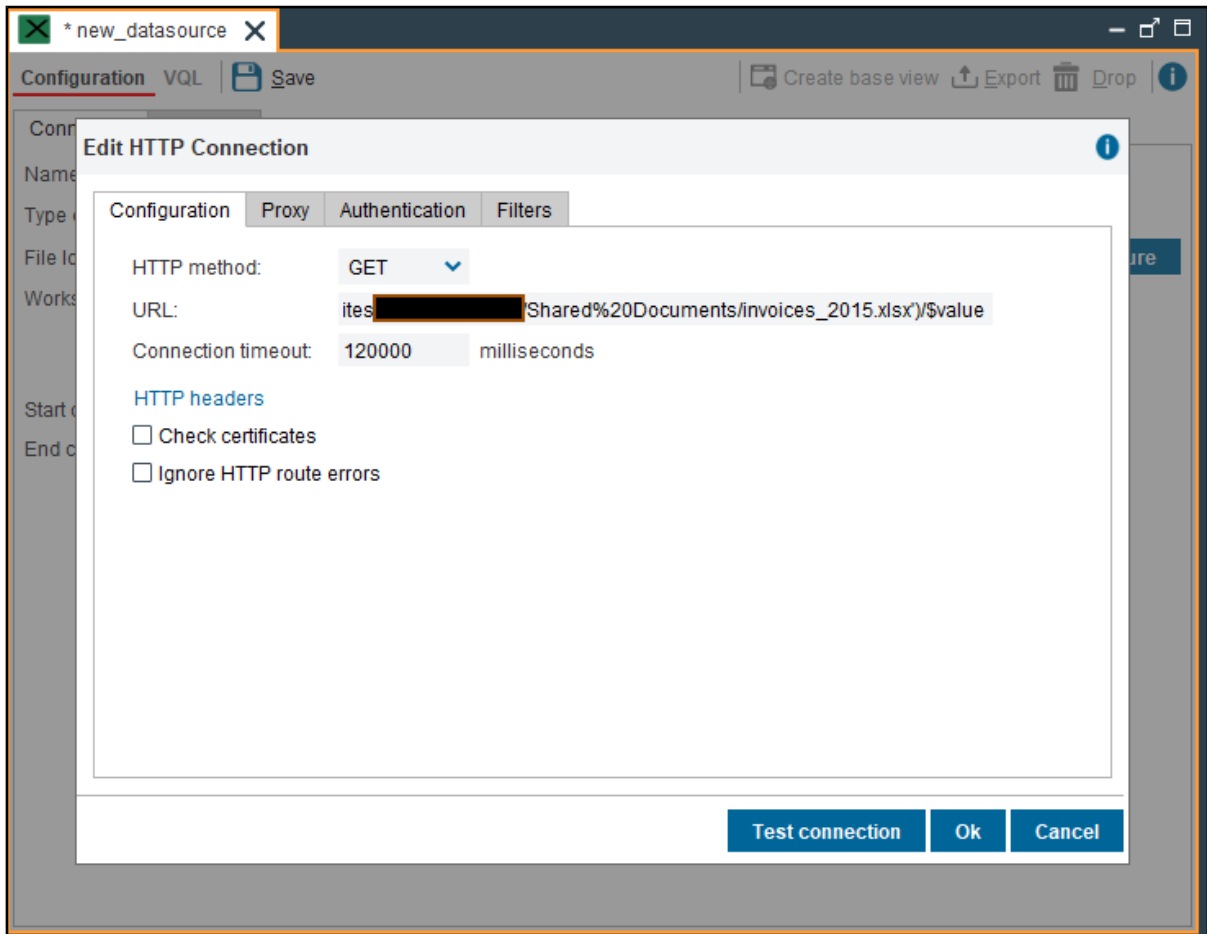
3 CONNECTING TO SHAREPOINT USING EXCEL, DELIMITED FILE, JSON AND XML DATA SOURCE

In this section we will explain how to create an Excel data source in Denodo to read Excel files from SharePoint. The same steps can be performed for JSON, XML or Delimited files as well.

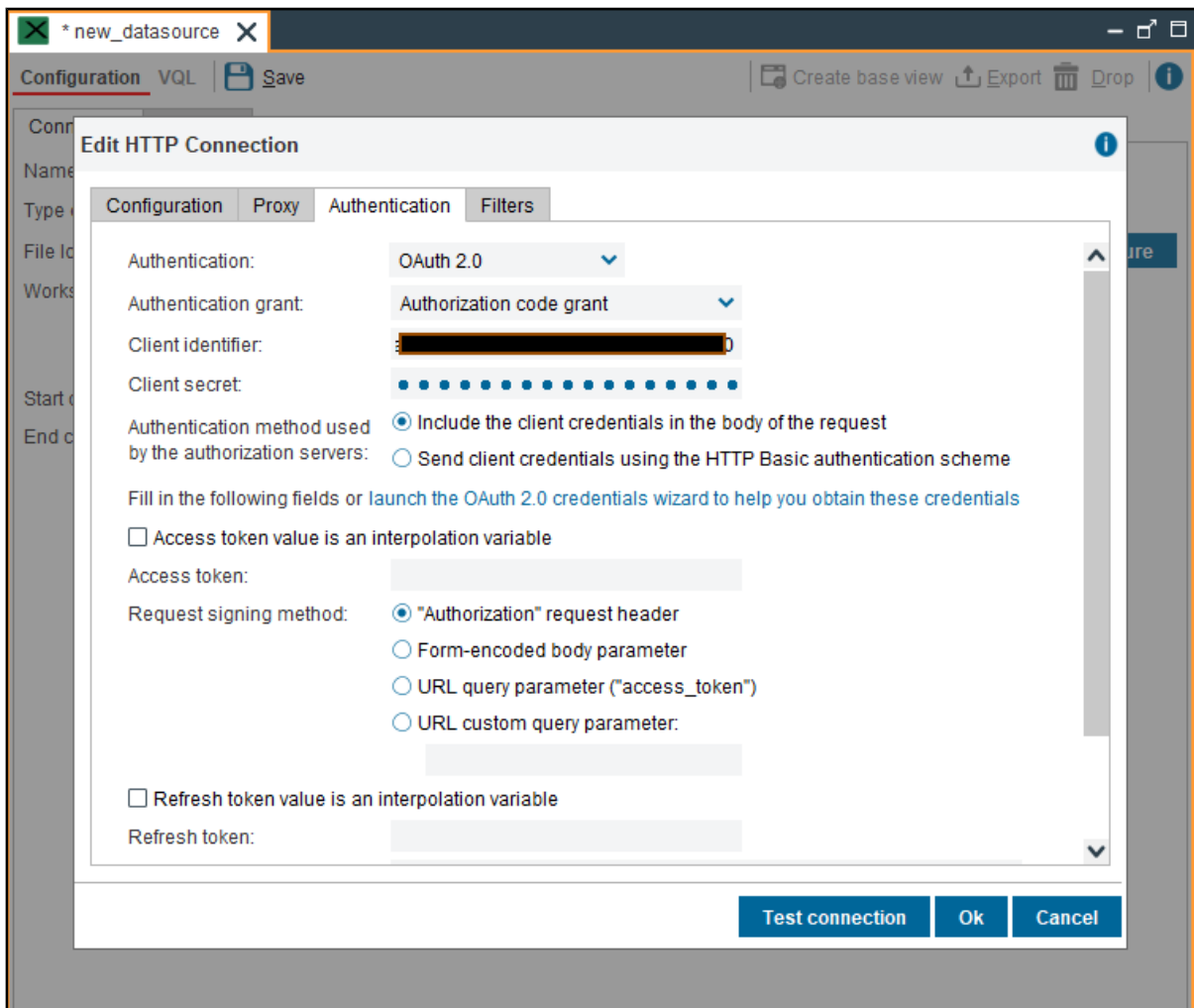
- From the Virtual DataPort Administration Tool or Design Studio, create an Excel Data Source.
- In the data source configuration, select the **File Location** as **HTTP Client** and select **Configure**.
- In the **Edit HTTP Configuration** dialog box set the HTTP Method to GET and provide the URL with the following format:

```
https://<tenant_name>.sharepoint.com/sites/site/_api/Web/GetFileByServerRelativePath(decodedurl='<location of an Excel file in the sharepoint>')/$value
```

Note: For some scenarios, the **site** parameter of the URL may need to be modified according to the company's Sharepoint structure. Review this URL with your Sharepoint administrator to make sure you use the correct one for your specific installation.



- In the Authentication section, select **OAuth 2.0** for **Authentication**.
- Paste the **Application (client) ID** from the newly created application to the **Client Identifier** and the **Value of the Client Secret** to the **Client Secret**.



- Then select the **Launch the OAuth 2.0 credentials wizard**. The OAuth 2.0 credentials wizard will pop up. See “Get the OAuth tokens using the OAuth credentials wizard” below.
- As for the **Token endpoint URL** use **https://login.microsoftonline.com/<tenant_id>/oauth2/v2.0/token** where **<tenant_id>** matches the tenant_id copied in one of the previous steps.
- Provide the Authorization server URL (again, replace **<tenant_id>** with the appropriate value).

https://login.microsoftonline.com/<tenant_id>/oauth2/v2.0/authorize

- Select the **Redirect URI**, provide the redirect URI that you used to register an application in the first step. Make sure that it matches.
- For the **Scopes** option provide the values

- `https://<tenant_name>.sharepoint.com/AllSites.Read`
- `offline_access`

Remember to replace `<tenant_name>` with your actual tenant name. The value `offline_access` is required in order to obtain the refresh token.

- After providing all the URL's click on the option **Generate Authorization URL**.
- When the Authorization URL is generated, click on the **Open URL**.
- Once the URL is opened on your desired browser, provide the permission, and once the permission is given the redirect URL with code would be generated. In this example, the format of the URL will be:

```
https://<hostname>:9443/oauth/2.0/redirectURL?
code=<>&state=<>&session_state=<>#
```

- Copy the full URL and paste the same on the **Paste the authorized response URL** and select **Obtain the OAuth 2.0 credentials**.
- Select **Ok**.

OAuth 2.0 Credentials Wizard

Follow these steps to obtain the OAuth 2.0 credentials.

- Enter the authentication details:

Token endpoint URL:	<input type="text" value="https://[redacted]/oauth2v2.0/token"/>
Authorization server URL:	<input type="text" value="https://[redacted]/oauth2v2.0/authorize"/>
Redirect URI:	<input checked="" type="radio"/> <input type="text" value="https://localhost:9443/oauth/2.0/redirectURL.jsp"/> <input type="radio"/> <input type="text" value="https://"/>
Scopes:	<input type="button" value="+"/> <input type="button" value="x"/> <input type="text" value="[redacted].sharepoint.com/AllSites.Read"/> <input type="button" value="x"/> <input type="text" value="offline_access"/>
Set the "state" request parameter:	<input checked="" type="checkbox"/>
- Generate the authorization URL

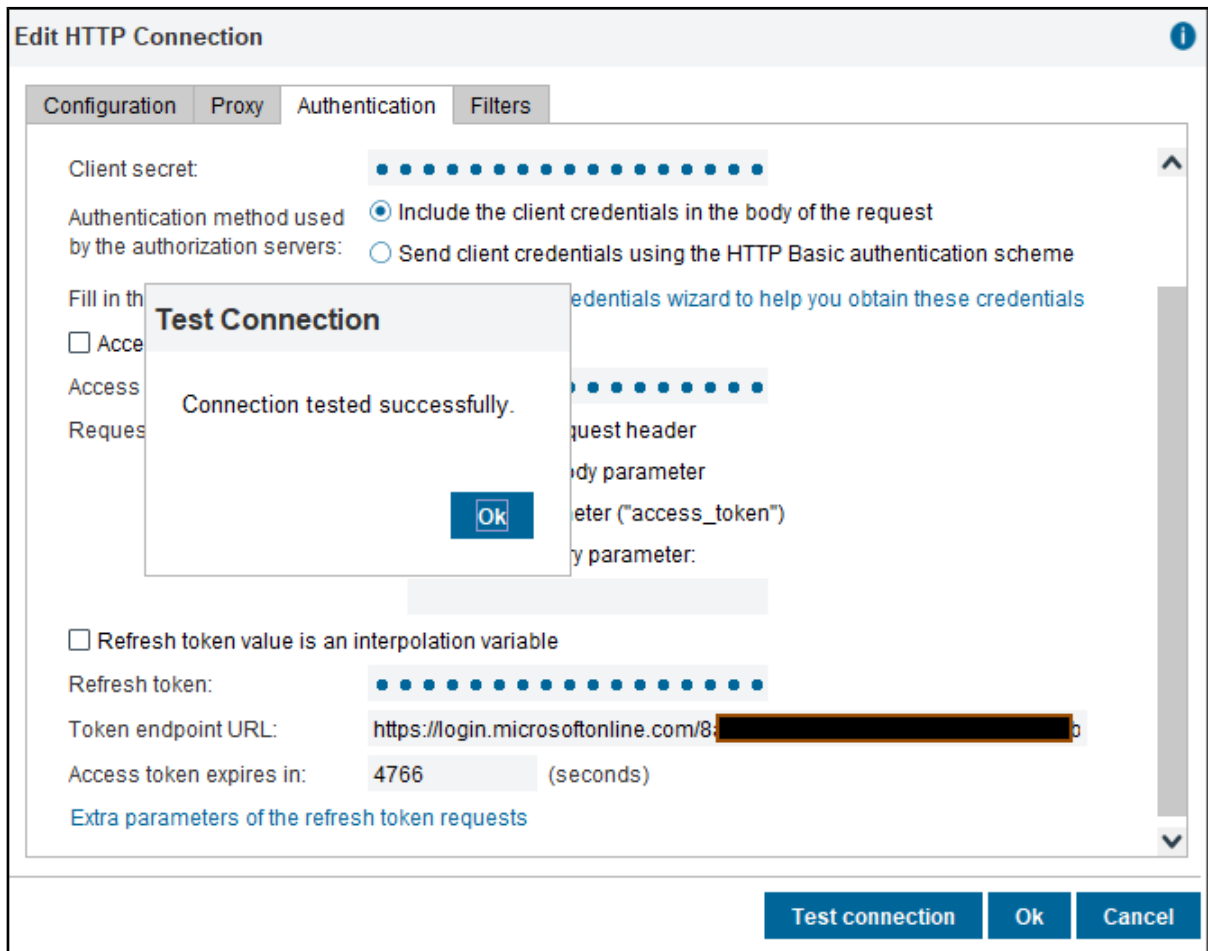
Authorization URL:

Open URL: [https://login.microsoftonline.com/\[redacted\]/oauth2v2.0/authorize?client_id=\[redacted\]&redirect_uri=https://localhost:9443/oauth/2.0/redirectURL.jsp&response_type=code&scope=\[redacted\].sharepoint.com/AllSites.Read%20offline_access](https://login.microsoftonline.com/[redacted]/oauth2v2.0/authorize?client_id=[redacted]&redirect_uri=https://localhost:9443/oauth/2.0/redirectURL.jsp&response_type=code&scope=[redacted].sharepoint.com/AllSites.Read%20offline_access)
- Paste the authorization response URL:
- Obtain the OAuth 2.0 credentials

The OAuth 2.0 credentials have been obtained.
Click Ok to store them.

[Click to see the extra properties returned by the OAuth token server](#)

Once the OAuth 2.0 Credentials are obtained , test the connection to check if the connection can be established.



Now you will be able to create a new base view from this Excel data source.

Summary Edit Options VQL Execution panel Used by Associations Publish Export Drop

View type: Base

Schema:

Field Name	Field Type	Description
invoice_id	int	
date_invoice	text	
order_id	int	
date_placed	text	

Owner: admin Last modifier: admin

Creation: 27.12.2021 18:49:14 Last modification: 27.12.2021 18:49:14

Swap status: default Cache status: off

Folder: /2 - base views/authorization_code_grant

Execute Query Results

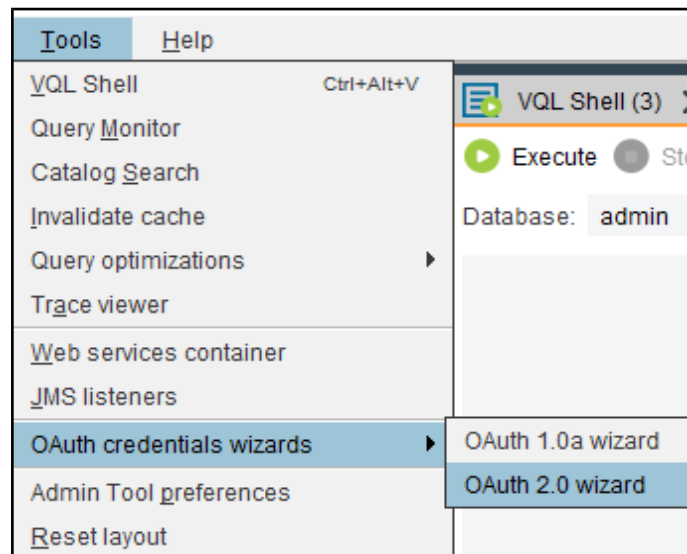
Results Execution Trace Query: SELECT * FROM ds_sharepoint LIMIT 150 CONTEXT ('i18n'='de_euro', 'cache_v

Total rows received: 150 (shown 150)

invoice_id	date_invoi...	order_id	date_placed	date_deliv...	date_closed	first_name	last_name	email	addre
1	Fri Feb 06 ...	21742	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Julia	Chavez	jchavezbx...	9122 E
2	Fri Feb 06 ...	28912	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Harry	Boyd	hboydgh@r...	6 Melb
3	Fri Feb 06 ...	64042	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Adam	Stephens	astephens...	23 Div
4	Fri Feb 06 ...	6307	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	William	Lynch	wlynchnj@...	70536
5	Fri Feb 06 ...	33322	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Ruth	Bailey	rbaileylj@b...	96 Ind
6	Fri Feb 06 ...	42232	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Jennifer	King	jking2a@w...	39305
7	Fri Feb 06 ...	40207	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Donna	Torres	dtorresot@...	360 Tc
8	Fri Feb 06 ...	66997	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Betty	Watson	bwatson7n...	4 Nortl
9	Fri Feb 06 ...	5392	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Brenda	Palmer	bpalmerip...	1537 S
10	Fri Feb 06 ...	11167	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Denise	Sullivan	dsullivanh1...	0005 \
11	Fri Feb 06 ...	17092	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Kimberly	Burns	kburnsfp@...	8 Tenr
12	Fri Feb 06 ...	13012	Sun Feb 01...	Wed Feb 0...	Fri Feb 06 ...	Fred	Webb	fwebbhb@j...	89131

3.1 GET THE OAUTH TOKENS USING THE OAUTH CREDENTIALS WIZARD

If you need to save the OAuth tokens to use for example in the JSON-based parts of the Denodo templates for SharePoint, you can generate them using the OAuth credentials wizard following the steps described below.



OAuth 2.0 Credentials Wizard ✕

Follow these steps to obtain the OAuth 2.0 credentials.

- Enter the authentication details:
 - Authentication grant: Authorization code grant
 - Client identifier:
 - Client secret:
 - Authentication method used by the authorization servers:
 - Include the client credentials in the body of the request
 - Send client credentials using the HTTP Basic authentication scheme
 - Token endpoint URL: https://login.microsoftonline.com/
 - Authorization server URL: https://login.microsoftonline.com/
 - Redirect URI:
 - http://localhost:9090/oauth/2.0/redirectURL.jsp
 - https://
 - Scopes:
 - +
 - 🗑️ AllSites.Read
 - Set the "state" request parameter:
- Generate the authorization URL
 - Authorization URL:
- Paste the authorization response URL:
- Obtain the OAuth 2.0 credentials

Close

Where:

- Client identifier: the client id from the newly created Azure application
- Client secret: the value of the secret generated in the previous step for the newly created application.
- Token endpoint URL:

`https://login.microsoftonline.com/<tenant>/oauth2/token.`

- Authorization server URL :
- `https://login.microsoftonline.com/common/oauth2/authorize?resource=https://<tenant_name>.sharepoint.com/.`
- Redirect URI: the default one.
- Scopes: add `AllSites.Read` and `offline_access`.

After providing all the values click on the option `Generate Authorization URL` and, when the Authorization URL is generated, click on the `open URL` link.

Once the URL is opened on your desired browser, provide the permission, and once the permission is given a URL with code would be generated. The format of the URL will be: `https://<tenant_name>.sharepoint.com/?code=<>&state=<>&session_state=<>#`

Paste the same on the `Paste the authorized response URL` and select `Obtain the OAuth 2.0 credentials`.

Finally, click on `Copy the credentials to the clipboard` to save the OAuth tokens.

4 APPENDIX I: AUTHENTICATING WITH NTLM

NOTE: This section only applies to SharePoint Server.

First of all, you need to download the **Denodo OData 2 Custom Wrapper** and import it as explained “Using the OData 2 Custom Wrapper” and then follow these steps.

To get the OData URL from the SharePoint Server the following steps are required:

- Open a browser and navigate to the following URL:

```
http://<tenant>.sharepoint.com/_vti_bin/ListData.svc
```

Replace <tenant> with your actual tenant name. This will be the **Service Endpoint**.

- Select the data feed you want to connect with. In the connection parameters this will be the **Entity Collection**.
- Finally it is necessary to determine the OData version (**Service Version**). You can do it by checking the following URL:

```
https://<tenant>.sharepoint.com/_api/$metadata
```


The following wrapper parameters need to be configured as follow to define datasource:

Service Endpoint	https://<tenant_name>.sharepoint.com/_vti_bin/ListData.svc Replace <tenant_name> with your actual tenant name
Service Format	XML-Atom
Service Version	Obtained in the previous step
User	Your NTLM user
Password	Your NTLM password
Use the NTLM Authentication	true
NTLM Domain	Your NTLM domain

Fill in Proxy fields if required.

This is how it will look like in the OData 2 Custom Wrapper:

Input parameters of the data source

Click to refresh the input parameters of the data source 

Service Endpoint *:

Service Format *:

Service Version:

Pass-through session credentials

User:

Password:

Proxy Host:

Proxy Port:

Proxy User:

Proxy Password:

Use NTLM Authentication

NTLM Domain:

Timeout:

Save the changes and select the Create base view option. This will display the Edit Wrapper Parameter Values dialog. Use the data feed selected as Entity Collection to query:

Edit Wrapper Parameter values

Enter values for the following wrapper parameters:

Entity Collection *:

Expand Related Entities

Enable Pagination

5 APPENDIX II: CONNECTING TO LISTS AS RSS FEEDS

NOTE: This section only applies to SharePoint Server.

To get the RSS URL from the SharePoint Server the following steps are required:

- Navigate to the site where the List is stored.
- On the left top corner of the screen select Site actions - View All Site Content.
- Select the list that is going to be used.
- Once the list has been selected open the List Tools menu and select the List option.
- In order to access the list using the RSS feed click on the RSS feed icon of the top menu and copy the RSS URL.

Once you have the RSS URL it can be imported into VDP using an XML data source:

- Create a new XML Data source selecting New > Data Source > XML in the contextual menu.
- Select "Http Client" as "Data route" and click on the "Configure" button.
- On the "URL" field insert the RSS URL.
- Fill the "Authentication" Tab if required and click the "OK" button. By default, SharePoint Server uses NTLM authentication but other authentication modes might be configured.
- Create the Base View.
- Execute the view and get the results.

Although the recommended ways to import data from Microsoft SharePoint are through the RSS interface and OData, there are other possible ways to import information:

- Using the List XML with its List GUID.
- Using the SharePoint Web Services (WSS).
- Creating an ITPilot wrapper to perform a web extraction on the SharePoint web interface.

These are more complex options and are not recommended unless the recommended options do not contain all the information that has to be imported into Virtual DataPort.

6 REFERENCES

[Quickstart: Register an application with the Microsoft identity platform](#)

[Denodo OData2 Custom Wrapper - User Manual](#)

[Denodo Templates for SharePoint - Quick Use Guide](#)