WhereScape Source Enablement Pack - Azure Data Lake Storage Gen2

This is a guide for installing Source Enablement Packs for WhereScape RED 8.6.1.x

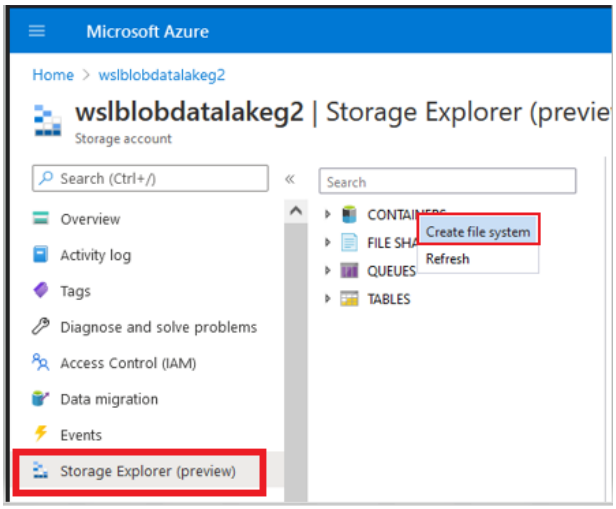
Prerequisites

- Python 3.8 or higher
  - Download python installer from https://www.python.org/downloads/
  - Select "Add Python 3.8 to PATH" from installation Window
- PIP Manager
  - From Command Prompt (Run As Administrator) run below command

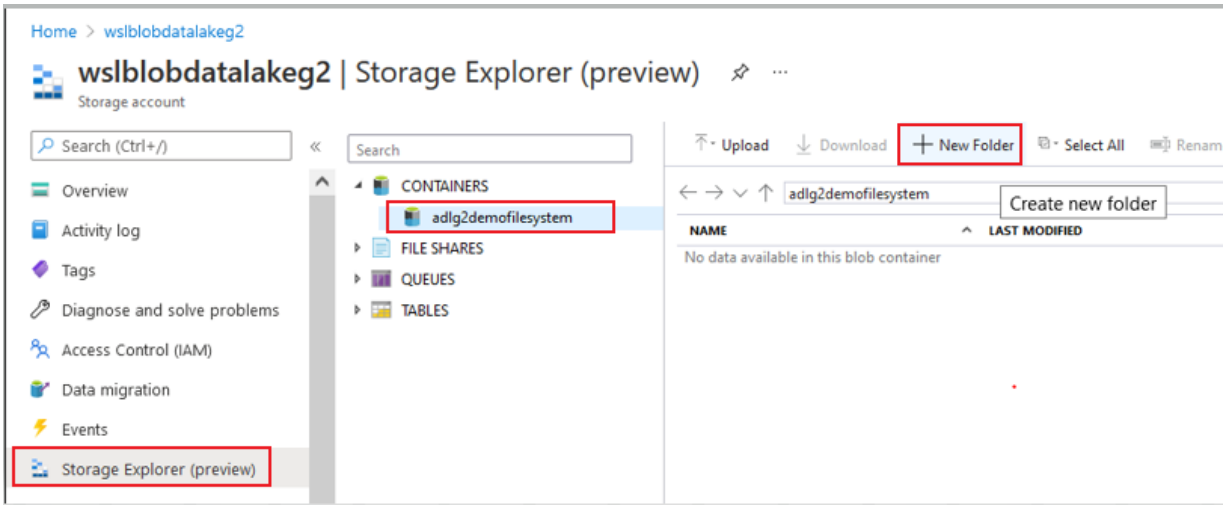
PIP Manager Install

```
python -m pip install --upgrade pip
```

- Azure Data Lake Storage Gen2
  - Azure Data Lake Storage Gen2 Account Name
  - Azure Data Lake Storage Gen2 Access Key
  - Azure Data Lake Storage Gen2 SAS Token
  - Azure Data Lake Storage Gen2 File System Name (Created in Storage Explorer Preview) . Eg:



  - Azure Data Lake Storage Gen2 Directory Name (Created in Storage Explorer Preview). Eg:



  - Install Python package - pip install azure-storage-file-datalake

- Net Framework 4.8 or higher
- Windows Powershell version 5 or higher

- Run these commands in "Windows PowerShell":

Install Azure Storage Package

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
Install-Module Az.Storage
```

Note: Use a 64-bit powershell terminal

Enablement Pack Setup Scripts

The Enablement Pack Install process is entirely driven by scripts. The below table outlines these scripts, their purpose and if "Run as Administrator" is required.

#	Enablement Pack Setup Scripts	Script Purpose	Run as Admin	Intended Application
1	install_Source_Enablement_Pack.ps1	Install Python scripts and UI Config Files for browsing files from Amazon S3, Azure Data Lake Gen2,Google Drive	Yes	New and Existing installations

Powershell script above provides some help at the command line, this can be output by passing the "-help" parameter to the script.

* *Note that on some systems executing Windows Powershell scripts is disabled by default, see troubleshooting for workarounds*

Source Enablement Pack Installation

Run Windows Powershell as Administrator

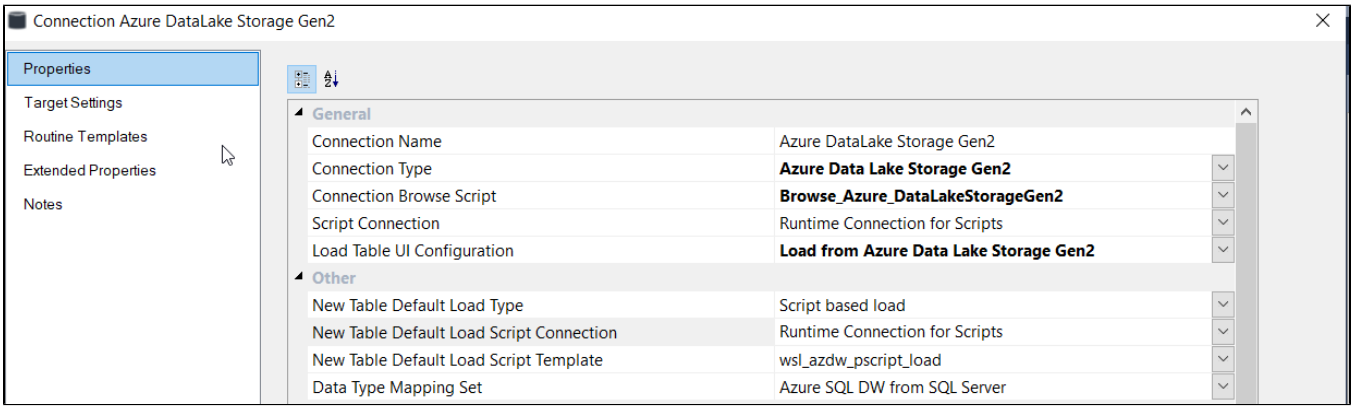
Install Source Connectivity Packs

```
<Script1 Location > Powershell -ExecutionPolicy Bypass -File .\install_Source_Enablement_Pack.ps1
```

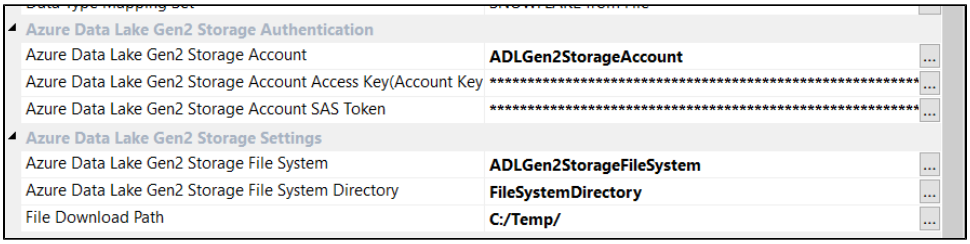
If prompted enter source enablement pack as **'Azure'**

Azure Data Lake Storage Gen2 Connection Setup

1. Login to RED
2. Check in **Host Script** Browse_Azure_DataLakeStorageGen2 in objects list.
3. Check **UI Configurations** in Menu, Tools UI Configurations Maintain UI Configurations
4. Create new connection in RED
5. Select properties as shown in below screenshot

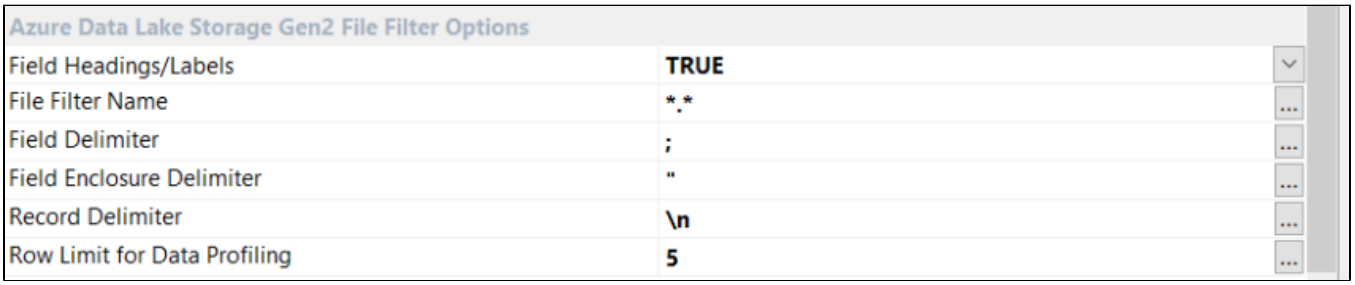


- Property Section **Azure Data Lake Gen2 Storage Authentication**
    - Azure Data Lake Gen2 Storage Account : Azure Data Lake Gen2 Storage Account Name,
      The token used to read storage account name in the scripts is *$WSL_SRCCFG_azureStorageAccountName$*
    - Azure Data Lake Gen2 Storage Account Access Key(Account Key) : Azure Data Lake Gen2 Storage Account Access Key also called as Account Key,
      The token used to read access key is environment variable: WSL_SRCCFG_azureStorageAccountAccessKey
    - Azure Data Lake Gen2 Storage Account SAS Token: Azure Data Lake Gen2 Storage Account Shared Access Signature (SAS) Token,
      The token used to read environment variable: *WSL_SRCCFG_azureSASToken*
- Property Section **Azure Data Lake Gen2 Storage Settings**
    - Azure Data Lake Gen2 Storage File System : Azure Data Lake Gen2 Storage File System name,
      The token used to read storage file system name in the scripts is *$WSL_SRCCFG_azureStorageFileSystem$*
    - Azure Data Lake Gen2 Storage File System Directory: Azure Data Lake Gen2 Storage Directory name where blob exists,
      The token used to read directory name in the browse script is *$WSL_SRCCFG_azureStorageFileSystemDirectory$*
    - File Download Path: Local directory where file needs to be downloaded for data profiling from the sourceAzure Data Lake Gen2 Storage.For Example Eg: C:\\Source\\Subfolder\\ or C:/Source/Subfolder/
      The token used to read path name in the browse script is *$WSL_SRCCFG_fileDownloadPath$*



- Property Section **Azure Data Lake Gen2 Storage File Filter Options**
    - Field Headings/Labels : Indicates whether the first line of source file contains a heading/label for each field, which is not regarded as data so it should not be loaded.
      The token used to reader field header boolean value in the script is *$WSL_SRCCFG_azureDataLakeGen2FirstLineHeader$*
    - File Filter Name: Indicates source file name.Provide Azure Blob filename pattern.
      The file list filters with file extensions,file name patterns.
        - *.*
        - *.<File Extension>
        - <File Name>.<File Extension>
        - <File Name Start>*

      The Token used to read File Filter Name in the scripts is *$WSL_SRCCFG_azureDataLakeGen2FileFilterName$*



    - Field Delimiter : This is a character that separates the fields within each record of the source file.The field delimiter identifies end of each field.For Example, comma (,),pipe(|).
      The token used to reader field delimiter in the script is *$WSL_SRCCFG_azureDataLakeGen2FieldDelimiter$*
    - Field Enclosure Delimiter: This is a character that delimits BOTH start and end of field value i.e. encapsulates value. A double quote is common enclosure delimiter.
      The token used to reader field enclosure delimiter in the script is *$WSL_SRCCFG_azureDataLakeGen2FieldEnclosureDelimiter$*

- Record Delimiter : This is to identify how each line/record in source file is ended/terminated/delineated.Default is '\n'
  The token used to read record delimiter value in the script is *$WSL_SRCCFG_azureDataLakeGen2RecordDelimiter$*

- Row Limit for Data Profiling : Number of records to scan for Data Profiling.Data profiling is used to get the column names and data types from the source file.By default 100 records will be scanned.
  The token used to read record delimiter value in the script is *$WSL_SRCCFG_azureDataLakeGen2RowLimit$*

Troubleshooting and Tips

Run As Administrator

Press the Windows Key on your keyboard and start typing cmd.exe, when the cmd.exe icon shows up in the search list right click it to bring up the context menu, select "Run As Administrator"

Now you have an admin prompt navigate to to the folder where you have unpacked your WhereScape Source Enablement Pack to using the 'cd' command:

```
C:\Windows\system32> cd <full path to the unpacked folder>
```

Run Powershell (.ps1) scripts from the administrator prompt by typing the Powershell run script command, for example:

```
C:\temp\EnablementPack>Powershell -ExecutionPolicy Bypass -File .\install_Source_Enablement_Pack.ps1
```

Notes: In the event you can not bypass the Powershell execution policy due to group policies you can instead try "-ExecutionPolicy RemoteSigned" which should allow unsigned local scripts.

Windows Powershell Script Execution

On some systems Windows Powershell script execution is disabled by default. There are a number of workarounds for this which can be found by searching the term "Powershell Execution Policy".

Here is the most common workaround which WhereScape suggests, which does not permanently change the execution rights:

Start a Windows CMD prompt as Administrator, change directory to your script directory and run the WhereScape Powershell scripts with this command:

- cmd:>Powershell -ExecutionPolicy Bypass -File .\<script_file_name.ps1>

Restarting failed scripts

Some of the setup scripts will track each step and output the step number when there is a failure. To restart from the failed step (or to skip the step) provide the parameter "-startAtStep <step number>" to the script.

Example:

Powershell -ExecutionPolicy Bypass -File .\<script_file_name.ps1> -startAtStep 123

Tip: to avoid having to provide all the parameters again you can copy the full command line with parameters from the first "INFO" message from the beginning of the console output.

If a valid RED installation can not be found

If you have Red 8.6.1.x or higher installed but the script (install_Source_Enablement_Pack.ps1) fails to find it on you system then you are most likely running PowerShell (x86) version which does not show installed 64 bit apps by default. Please open a 64 bit version of PowerShell instead and re-run the script.