Jan Cordtz

Jcordtz@Microsoft.com



Purview loader for Databricks, Oracle and MSSQL sources

Table of Contents

[Overview 2](#_Toc208313352)

[Initial setup 2](#_Toc208313353)

[Requirements 2](#_Toc208313354)

[PURVIEW\_NAME 2](#_Toc208313355)

[TENANT\_ID 2](#_Toc208313356)

[CLIENT\_ID and CLIENT\_SECRET 2](#_Toc208313357)

[Scripts 3](#_Toc208313358)

[Establish environment 4](#_Toc208313359)

[Each time you open a new Ubuntu terminal 5](#_Toc208313360)

[Load purview with a new “system.” 6](#_Toc208313361)

# Overview

This document describes a small command-line based utilities provided to European Medicine Agency (EMA) for maintaining Purview through the Purview API.

***The code is provided as “as-is” code and is Ubuntu Linux based.***

On a Windows machine access to Ubuntu can be established using WSL.

The code is based on the Purview cli interface provided through GitHub:  
[GitHub - tayganr/purviewcli: Microsoft Purview CLI](https://github.com/tayganr/purviewcli)

# Limitations

For each source – information from the “server” to the “tables” are loaded. No columns in added.

The hierarchy structure and the connection to the collection shown in the Purview Portal does not as of now.

# Initial setup

This document uses a shell prompt in Ubuntu for screen dumps.

## Requirements

It is a requirement that access to Purview using an Azure client has been established.

Hence the scripts supplied requires the following variable to be set:

* PURVIEW\_NAME
* TENANT\_ID
* CLIENT\_ID
* CLIENT\_SECRET

The value for these variables must be entered into the file setup\_purview.sh

### PURVIEW\_NAME

This is the name of the purview account you want to use.

### TENANT\_ID

The TENANT\_ID can be found in the Azure portal navigating to **Microsoft Entra ID** and on the Overview tab you will find this value.

### CLIENT\_ID and CLIENT\_SECRET

To obtain CLIENT\_ID and CLIENT\_SECRET you will need to follow these steps:

* Create an Azure Active Directory Application in Microsoft Entra ID:
  + In Azure portal navigate to **Microsoft EntraID**.
  + Click on App registrations and then **New registration**.
  + Once the application is created, note down the Application (client) ID. This is your CLIENT\_ID.
* Generate a Client Secret:
  + In the application settings, go to **Certificates & secrets**.
  + Under the **Client secrets** section, click **new client secret**.
  + Generated client secret is your CLIENT\_SECRET – remember to copy this as instructed.

## 

This Client-app must then be granted access to the domains in Purview.

* In the **Purview Portal** go to Data Map and then select the default domain
* In the overview section select the tab “Role assignments”
* Assign your Registered Client to the following roles in purview (clicking the “person with the plus” icon):
  + Collection Admin
  + Data Source Admin
  + Data curators
  + Data Readers

Detailed instructions can be found here   
<https://learn.microsoft.com/en-us/purview/data-gov-api-rest-data-plane#set-up-authentication-using-service-principal>

## Scripts

This document assume that are using Ubuntu using WSL in windows.

If you are using Ubuntu “clean” please change directory names accordingly.

Unzip the files provided to a directory of your preference – this document uses the name for a directory located in the root of c-drive called **purview**.

Start the ubuntu distribution using the Windows Teminal.

Change directory to /mnt/c/purview.

In this directory you will find files used by this setup. The 4 important files are:

**setup\_purview.sh** – this script must be run once, and it sets up the variable described above and write them into the init\_purview.sh script.  
  
The value for the 4 variables must be entered into the file   
named parameters,txt (see below).  
  
The variables are named <<VARIABLE\_NAME>> and this must then be changed accordingly, including removing the <<>>.

**init\_purview.sh** – this file is generated by the setup\_purview.sh scrips and must be run every time a new Ubuntu terminal is started using the command   
. ./init\_purview.sh

**load\_purview.sh** – this is the main scrip used to load Purview. It required the init\_purview.sh script to have been run as described. It uses a dialogue to get the values needed for the load.

**tables.csv** – to load Purview the script read a file like tables.csv (provided as an example). It is important that the structure of this file is a comma separated file with the 3 values table\_schema, table\_name, table\_type.

**Before using the commands described in the following please issue the following command from the command prompt:**

* **chmod +x setup\_purview.sh**
* **chmod +x load\_purview.sh**

# Establish environment

Before running the scripts, you need to change the **parameters.txt** file and then run the script setup\_purview.sh

In the **parameters.txt** file you must specify the values for PURVIEW\_NAME, TENANT\_ID, CLIENT\_ID and CLIENT\_SECRET.

The format of the init\_parameters.txt file is as follows.

A red line with black text

AI-generated content may be incorrect.

After these changes has been made the setup script must be run using the command

**./setup\_purview.sh**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Note**

If you would like this script also to install the required python files for the setup to run – instead of doing this manually – you can run this script using this command

**./setup\_purview.sh setup**

**A screenshot of a computer program

AI-generated content may be incorrect.**

# Each time you open a new Ubuntu terminal

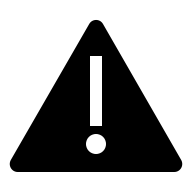
Every time the Ubuntu shell is opened the command **. ./init\_purview.sh** must be issued.

**Please note that both punctuations must be entered!!!**

This command initializes the parameters to be used for connecting to the Purview account.

A white background with black text

AI-generated content may be incorrect.



This command must run every time the cloud shell is activated.

# Load purview with a new “system.”

To create a new placeholder for a “system” within the Purview instance you then run

the command **./load\_purview.sh**

This command has a dialogue. First the connection is made to the Purview account.

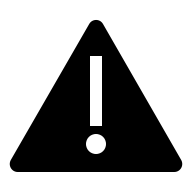


Please note that script provides default values in [] and that the default for Yes/No questions is **N**o.

Parameters that the dialogue asks for

* Name of parent collection – default is the root collection.
* Name of system – whatever name you would like to use. You cannot use “spaces” in the name, use “\_” instead.
* You can choose type of system being MS Sql, Oracle or Databricks.
* The name of the file the contains the “tables” the need to be load. This name can be “anything” if that file has the right structure.

If you are satisfied with the information enter Y to continue.

 Any administration/deletion after this script has been run must be done through the Purview Portal.

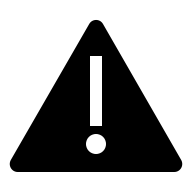
**NOTE:** In case you get an error, you must likely need to run the initialization command**. ./init\_purview.sh**

# Cleanup procedure

Every time the load\_purview script is run, and a source is created, the process registers its progress in a file called <SOURCE\_NAME>\_<PID>\_log.sh, where <SOURCE\_NAME> is the name of the source being created, <PID> is the process\_id for the Linux process – provided for uniqueness, so that the same source can be loaded multiple times, for instance with different tables.

The structure of these files in so that the contains the corresponding “delete” statements using pv of what has been created.

So, if you want to delete everything from a run simply execute the file using the command ./<SOURCE\_NAME>\_<PID>\_log.sh.

Alternatively, you can use the commands shown in the file to selective delete unwanted items.  
  
 If you run the complete file the Source itself is also deleted, meaning that this will affect any other runs that’s has been made towards this source.