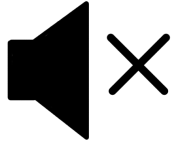


Jan 18, 2022

CDQ Custom Dashboard Utilizing IICS REST API

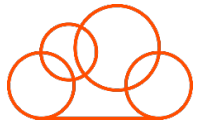
Vivek Singh, Solutions Architect

Housekeeping Tips



- Today's Webinar is scheduled for **1 hour**
- The session will include a webcast and then your questions will be answered live at the end of the presentation
- All dial-in participants will be muted to enable the speakers to present without interruption
- Questions can be submitted to "All Panelists" via the **Q&A option** and we will respond at the end of the presentation
- The webinar is **being recorded** and will be available on our **INFASupport YouTube channel** and **Success Portal** - where you can download the **slide deck** for the presentation. The link to the recording will be emailed as well.
- Please take time to complete the **post-webinar survey** and provide your feedback and suggestions for upcoming topics.

Feature Rich Success Portal



Bootstrap trial and
POC Customers



Enriched Customer
Onboarding
experience



Product Learning
Paths and Weekly
Expert Sessions



Informatica
Concierge



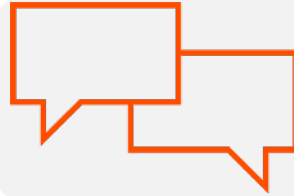
Tailored training and
content
recommendations

More Information



Success Portal

<https://success.informatica.com>



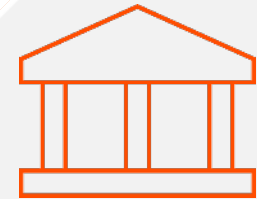
Communities & Support

<https://network.informatica.com>



Documentation

<https://docs.informatica.com>



University

<https://www.informatica.com/in/services-and-training/informatica-university.html>

Safe Harbor

The information being provided today is for informational purposes only. The development, release, and timing of any Informatica product or functionality described today remain at the sole discretion of Informatica and should not be relied upon in making a purchasing decision.

Statements made today are based on currently available information, which is subject to change. Such statements should not be relied upon as a representation, warranty or commitment to deliver specific products or functionality in the future.

CDQ Custom Dashboard Utilizing IICS REST API

Vivek Singh
Solutions Architect

Sachin Jain
Principal Customer Success Technologist

Agenda

- Overview
- Requirement
- Design
- Solution Scope and Technical Details
- Demo
- Q&A

Overview

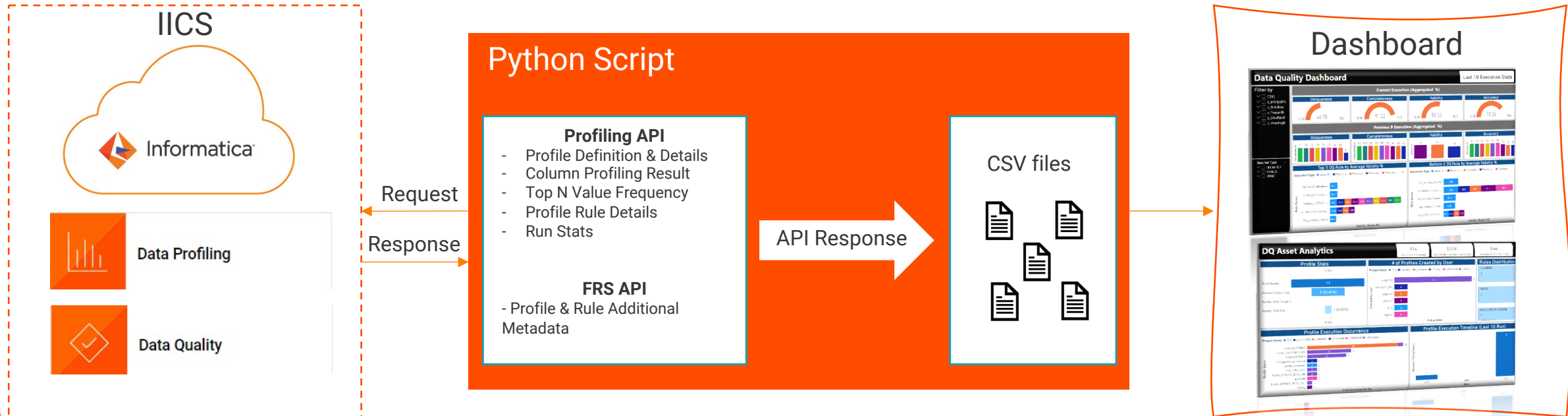
- The Cloud Data Quality Reporting Dashboard Template is designed to provide a framework to capture reporting metrics for data quality issues by extracting the Profile Details from the CDQ Profile Warehouse. Furthermore, it demonstrates how to visualize the profiling data in a business intelligence tool.
- Template includes a Python Script, and a sample Power BI Report and Dashboards illustrating DQ metrics. In addition, it includes sample data files on which provided Power BI Report is build, to allow you to view the reports without having the system up and running and generating metrics.
- Sample report is build using Power BI but any reporting tool can be utilized to design similar DQ Dashboard.

Requirement

- Managing the quality of data within an enterprise requires precise feedback on the data quality of any data movement. An automated Data Quality Report and Dashboard solution can provide a means to take already ongoing activities (DQ validation rules, Data Profiling) and combine them together to generate periodic data quality reports and dashboards to provide immediate metric feedback into the current and trending state of data quality.
- CDQ provides out of the box profiling solution to identify the data anomalies, apply validation rules and present the result in visual form. These visuals are dataset specific and business users might be interested in having a holistic and summarized view of Identified DQ anomalies across the organization/department.
- Building a custom DQ reporting dashboard using the Profiling data from CDQ can help users to achieve the stated objective. A summarized dashboard enables users to monitor the enterprise-wide data quality issues from a single window, resulting improved DQ monitoring and feedback process.

Design

Process Flow



Solution Scope

- Prerequisite -
 - DQ Rules are already created in CDQ and attached to the profiles
 - Profiles are successfully executed in CDP
 - A user account who can run the Rest API and have access to the CDQ/CDP assets
 - Python installed with below libraries –
 - requests, json, pandas, csv, os, sys, datetime, shutil, glob
- Covered in this solution Framework –
 - Python Script to call the API and capture response in CSV files
 - Use the generated CSV files to build Reporting dashboard
- Automating CSV file generation -
 - User can automate the Python Script execution using Cron Job (on Linux) or Task Scheduler (On windows) or can even use CDI.

Profiling REST API

Following IICS Rest APIs have been used

Login API

To log into IICS and get Session ID. The session ID expires after 30 minutes of inactivity and used as header parameter for all subsequent REST API calls.

[https://dm-**{POD_region}**.informaticacloud.com/ma/api/v2/user/login](https://dm-{POD_region}.informaticacloud.com/ma/api/v2/user/login)

Profiling API

- Get list of profiles
 - [https://**{POD}**-dqprofile.dm-**{POD_region}**.informaticacloud.com/profiling-service/api/v1/profile](https://{POD}-dqprofile.dm-{POD_region}.informaticacloud.com/profiling-service/api/v1/profile)
- Get details of each profile – like Profilable fields, Rules tagged, Sampling Option, Filter enable flag, Created by, date etc.
 - [https://**{POD}**-dqprofile.dm-**{POD_region}**.informaticacloud.com/profiling-service/api/v1/profile/**{id}**](https://{POD}-dqprofile.dm-{POD_region}.informaticacloud.com/profiling-service/api/v1/profile/{id})
- Get Column Profiling Details By Column Id – Column Profile Result as seen on Profile Result page in IICS – Data Profiling
 - [https://**{POD}**-dqprofile.dm-**{POD_region}**.informaticacloud.com/metric-store/api/v1/odata/Profiles\(''**{profileId}**''\)/Columns\(''**{columnId}**''\)](https://{POD}-dqprofile.dm-{POD_region}.informaticacloud.com/metric-store/api/v1/odata/Profiles(''{profileId}'')/Columns(''{columnId}''))
- Get Top N value frequencies by column id
 - [https://**{POD}**-dqprofile.dm-**{POD_region}**.informaticacloud.com/metric-store/api/v1/odata/Profiles\(''**{profileId}**''\)/Columns\(''**{columnId}**''\)/ValueFrequencies](https://{POD}-dqprofile.dm-{POD_region}.informaticacloud.com/metric-store/api/v1/odata/Profiles(''{profileId}'')/Columns(''{columnId}'')/ValueFrequencies)
- Lists all the profile runs for a profile ID – like Job execution status, Start/End time, Time Taken, Memory Consumed etc.
 - [https://**{POD}**-dqprofile.dm-**{POD_region}**.informaticacloud.com/profiling-service/api/v1/runDetail?profileId=**{profileId}**](https://{POD}-dqprofile.dm-{POD_region}.informaticacloud.com/profiling-service/api/v1/runDetail?profileId={profileId})

FRS API

- Get Additional Metadata Details – like profile and rule project, folder name, rule dimension name etc.
 - [https://**{POD}**.dm-**{POD_region}**.informaticacloud.com/frs/v1/Documents\(''**{in_frs_id}**''\)?\\$expand=userInfo](https://{POD}.dm-{POD_region}.informaticacloud.com/frs/v1/Documents(''{in_frs_id}'')?$expand=userInfo)

Python Script

Below Steps are performed by Python script

1. If specified folder does not exist, then create it else archive the CSV files if they are already present
2. Call CDQ Profiling Rest API in below sequence and capture the parsed response in respective CSV files

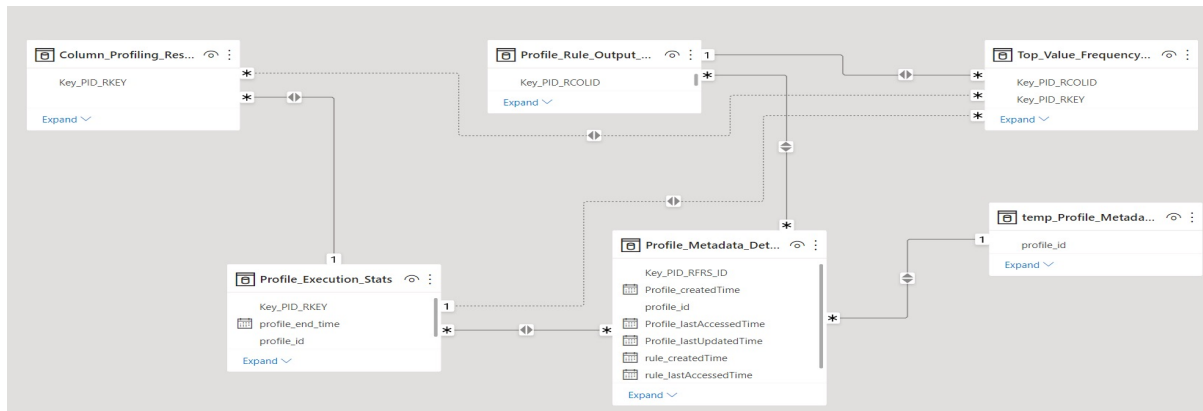
Rest API (Refer previous slide)	Output CSV file
IICS Login API	N/A
Get list of profiles API	N/A
Get details of each profile	Profile_Metadata_Details.csv Profile_Rule_Output_Fields.csv
Get Column Profiling Details By Column Id	Column_Profiling_Result.csv
Get Top N value frequencies by column id	Top_Value_Frequency_Data.csv
Lists all the profile runs for a profile ID	Profile_Execution_Stats.csv
Get Additional Metadata Details	Profile_Metadata_Details.csv

3. Delete the temporary files generated during the process

Reporting Dashboard

A sample reporting dashboard has been created on top of the CSV files generated

1. In this solution Power BI has been used to create the Reporting dashboard but any reporting tool which is capable of reading CSV files can be utilized
2. In Power BI below Model has been created to enable across report filter capability



• Here -

- Solid line represents Active relationship
- Dotted line represents Inactive relationship
- temp_Profile_Metadata_Details_distinct_profiles is a derived dataset referencing Profile_Metadata_Details

3. The Report contains 2 dashboards –
 - DQ Dashboard - Shows DQ anomalies like Uniqueness, Completeness, Validity, Accuracy check results
 - DQ Asset Analytics - Shows stats on DQ assets like # of profiles with distribution against rules tagged, sampling/filter applied, execution stats, rule count distribution
4. Users can also drill down the summarized graphs to view the underlying data in tabular format

How can I use this solution?

Once the prerequisites mentioned in Solution Scope slide are fulfilled, perform below steps -

1. Download the package (link is in Reference slide)

2. Update following lines in the python script

```
POD_region = "<YOUR IICS POD REGION NAME>"
```

```
username = "<YOUR IICS USERNAME>"
```

```
password = "<YOUR IICS PASSWORD>"
```

```
csv_file_path = "<COMPLETE PATH WHERE CSV FILES WILL BE GENERATED>"
```

Optionally you can also update value of history_rec_count variable. Default value is 10 which means report can show trend for last 10 executions

3. Execute Python Script. On successful execution you will see below 5 files in the "csv_file_path" location

- *Profile_Metadata_Details.csv, Profile_Rule_Output_Fields.csv, Column_Profiling_Result.csv, Top_Value_Frequency_Data.csv, Profile_Execution_Stats.csv*

4. Open report in Power BI Desktop and Update the Power BI datasets to point to the correct file location

5. Refresh Data for each datasets in the Report

Once you have tested the solution, You can automate the process of python script execution (refer next slide) and data refresh in the report (like using gateway in Power BI)

Python Script Execution Automation - Sample

- User can automate the Python script execution to get updated data files at a regular interval. Few Sample Ways -

Using Cron job in Linux Machine

Sample configuration in Linux Cron schedule file to run the python script every 1 hour

```
*/1 * * * /usr/bin/python3 /home/user/IICS_CDQ_Dashboard_Dataset.py >> /home/user/python_script_log.log 2>&1
```

Using Task Scheduler in Windows machine

The screenshot shows the Windows Task Scheduler interface. A task named 'Python_Script_Execution' is highlighted in blue. The task is set to run 'At 4:16 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.' The 'Next Run Time' is '11/21/2021 5:16:05 PM' and the 'Last Run Time' is '11/21/2021 4:16:05 PM'. The 'Last Run Result' is 'The operation completed successfully. (0x0)'. Below the task list, the 'General' tab is selected, showing the 'Action' as 'Start a program' with the 'Details' as 'C:\Users\vivesingh\Desktop\IICS_CDQ_Dashboard_Dataset.py'.

Using CDI

The screenshot shows the CDI Command Task configuration. The task is named 'Command Task 1' and is assigned using the 'Vivek_Local_Agent'. The 'Runtime Environment' is set to 'Content', the 'Script File Name' is 'IICS_CDQ_Dashboard_Dataset.py', and the 'Work Directory' is 'C:\Users\vivesingh\Desktop'. The 'Assigned Using' dropdown is set to 'Vivek_Local_Agent'.

Reference

- Getting Started with Cloud Data Profiling REST API
 - <https://docs.informatica.com/data-governance-and-quality-cloud/cloud-data-profiling/h2l/1547-getting-started-with-cloud-data-profiling-rest-api/getting-started-with-cloud-data-profiling-rest-api.html>
- CDQ/CDP custom reporting dashboard framework artifacts
 - https://github.com/vks9907/CDQ_Custom_Dashboard

Demo



Q & A



Thanks