A project report on

DASHBOARD ON TASK ACTIVITY USING TASK SCHEDULER (DELOITTE USI)

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology

in

Electronics and Communication

by

SAYANTAN BAL (19BEC1333)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering

Vellore Institute of Technology, Chennai

Vandalur-Kelambakkam Road

Chennai – 600127, India

May 2023

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DECLARATION

I hereby declare that the thesis entitled **"Dashboard on Task Activity using Task Scheduler (Deloitte USI)"** submitted by me, for the award of the degree **Bachelor of Technology in Electronics and Communication Engineering** is a record of bonafide work carried out by me as a PAT Internship.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Company: Deloitte USI

Date: 26/05/23

Name: Sayantan Bal

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering

CERTIFICATE

This is to certify that the project report titled **"Dashboard on Task Activity using Task Scheduler (Deloitte USI)"** submitted by **Sayantan Bal (19BEC1333)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** is a bonafide work conducted under my supervision. The project report fulfils the requirements as per the regulations of the University and meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Supervisor	**Head of the Department**
Signature:	Signature:
Name:	Name:
Date:	Date:

Examiner

Signature:

Name:

Date:

ABSTRACT

Task Scheduler is an inbuilt application in Microsoft Windows OS that manages and schedules tasks in an automated way that gives us full flexibility. It can be used to schedule scripts or programs at pre-defined times.

The Task Scheduler application provides various datapoints such as Triggers, Actions, Conditions, History etc.

Triggers are a set of criteria that must be met for the task to execute. For example, a particular time every day.

Actions allow us to create, import or delete a task. It refers to the actual running of the task.

Conditions allow us to set various boundaries regarding the execution of a task. For instance, we can use it to decide whether the task runs when the system is locked, when the user is logged in and so on.

History gives us the past run information of all the tasks inside Task Scheduler. It gives us the date, time and event information of every single run instance.

Settings enable us to specify additional behaviors of the task such as allowing the task to be run on demand, force stop if it fails multiple times.

ACKNOWLEDGEMENT

I wish to express my sincere thanks and deep sense of gratitude to my lead, Mr. Seera Pravakar for his constant guidance, encouragement, and moral support for the POC.

I am also grateful to my manager, Mr. Nalliboyina Balaji for giving me this opportunity at Deloitte and for guiding me in successfully completing the POC.

I would also like to thank my HOD, Dr. Mohanaprasad K, my Dean, Dr. Susan Elias and all other faculty for their support and knowledge throughout the capstone project.

I am indebted to my parents for always backing me and giving me this opportunity.

CONTENTS

LIST OF FIGURES

UDEMY CERTIFICATIONS

1. AGILE FUNDAMENTALS: INCLUDING SCRUM AND KANBAN
   The BA Guide | Jeremy Aschenbrenner, Vivek Khattri

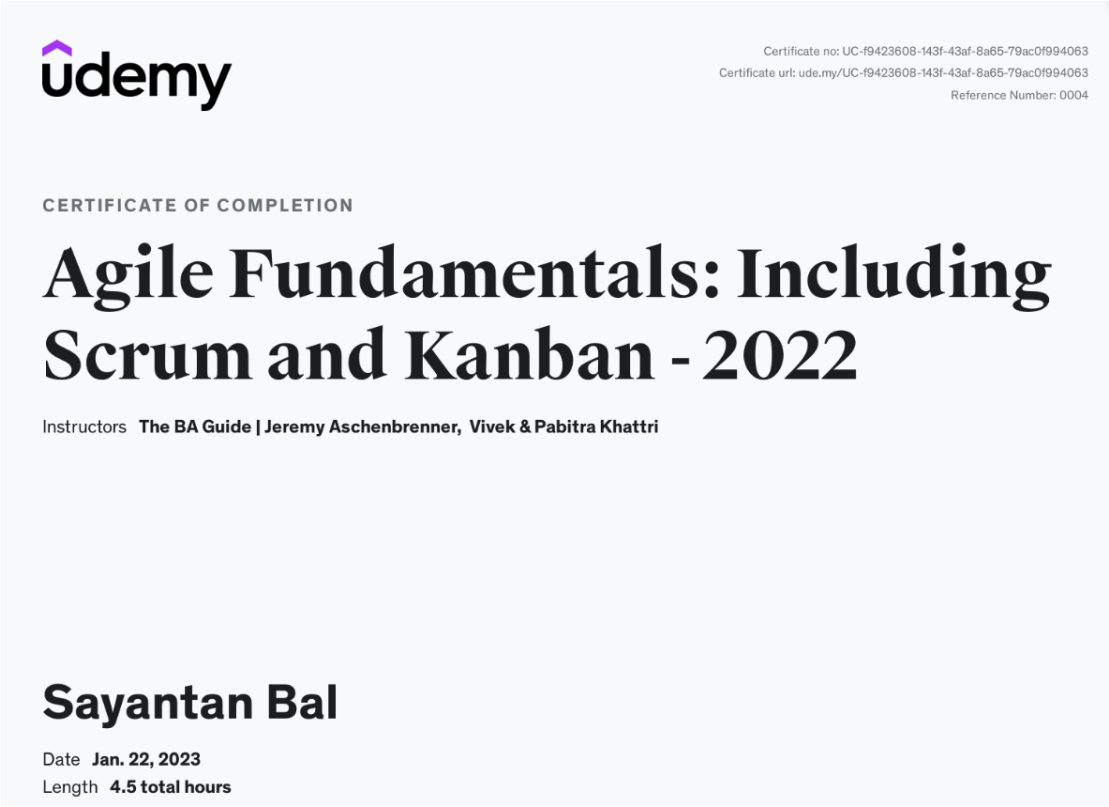
Learn how to get into the Agile mindset and achieve success in helping organizations evolve

Command a strong focus on delivering customer value and exceeding expectations

Master popular Agile frameworks Scrum, Kanban, and Scrumban, enabling you to land and excel in any Agile position

Explore fun, interactive, and highly effective lessons from best-selling instructors

Assist with your organization's current, or future, transformation to Agile



udemy

Certificate no: UC-f9423608-143f-43af-8a65-79ac0f994063
Certificate url: ude.my/UC-f9423608-143f-43af-8a65-79ac0f994063
Reference Number: 0004

CERTIFICATE OF COMPLETION

Agile Fundamentals: Including Scrum and Kanban - 2022

Instructors **The BA Guide | Jeremy Aschenbrenner, Vivek & Pabitra Khattri**

Sayantan Bal

Date **Jan. 22, 2023**
Length **4.5 total hours**

2. AZURE DEVOPS FUNDAMENTALS FOR BEGINNERS
   Brian Culp

Create an Azure DevOps organization

Align Azure DevOps work items using Agile, Scrum, or Basic work processes

Integrate an Azure DevOps code repository with GitHub

Fork and clone code using multiple tools

Understand the basic vocabulary of DevOps: what it is and why it matters

CI/CD: Understand how Pipelines facilitate Continuous Implementation and Continuous Deployment

Commit code changes and track Pull Requests

Push a code Repo from the command line of an Integrated Development Environment (IDE)


udemy

Certificate no: UC-459ba23c-58a6-4794-a92a-546741b7659f
Certificate url: ude.my/UC-459ba23c-58a6-4794-a92a-546741b7659f
Reference Number: 0004

CERTIFICATE OF COMPLETION

Azure DevOps Fundamentals for Beginners

Instructors Brian Culp

Sayantan Bal

Date Jan. 24, 2023
Length 3.5 total hours

3. MICROSOFT POWER BI DESKTOP FOR BUSINESS INTELLIGENCE
   Maven Analytics, Chris Dutton

Build professional-quality business intelligence reports from the ground up

Blend and transform raw data into beautiful interactive dashboards

Design and implement the same B.I. tools used by professional analysts and data scientists

Showcase your skills with two full-scale course projects (with step-by-step solutions)

Understand the business intelligence workflow from end-to-end

Learn from a best-selling instructor and professional BI developer


udemy

Certificate no: UC-0b957088-9656-4108-b99d-380136661d36
Certificate url: ude.my/UC-0b957088-9656-4108-b99d-380136661d36
Reference Number: 0004

CERTIFICATE OF COMPLETION

Microsoft Power BI Desktop for Business Intelligence

Instructors Maven Analytics, Chris Dutton

Sayantan Bal

Date May 18, 2023
Length 11 total hours

Fig 1.0 – Udemy Certificates

4. AZ-900 MICROSOFT AZURE FUNDAMENTALS IN A WEEKEND
   Ranga Karanam

Learn the Basic Concepts of Azure and Cloud Computing

Describe the benefits and considerations of using cloud services

Describe the differences between Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS)

Describe the differences between Public, Private and Hybrid cloud models

Understand the core Azure architectural components

Describe core products available in Azure

Describe solutions available on Azure

Understand Azure management tools

Understand securing network connectivity in Azure

Describe core Azure Identity services

Describe security tools and features of AzureDescribe Azure governance methodologies

Understand privacy, compliance, and data protection standards in Azure

Understand monitoring and reporting options in Azure

Understand Azure subscriptions

INTRODUCTION

Task Scheduler is a job scheduler in Microsoft Windows that launches computer programs or scripts at pre-defined times or after specified time intervals. It lets you automate tasks on Windows 10.

By tracking the status and properties of the scheduled jobs, a dashboard can be created in the PowerBI software which allows us to easily monitor the day-to-day running of all tasks which are scheduled in the system at one glance.

Such a dashboard will make it much more convenient to view exactly which tasks were completed successfully, which tasks are currently running and more importantly, which tasks failed.

The tasks which failed can also be investigated further, to try and figure out the exact reason behind the failure.

All jobs can be filtered using their run times, authors, and names. This allows us to view exactly the jobs that we need to monitor and accordingly, deal with them.

The details of all the jobs can be extracted using a number of methods such as Python, PowerShell, JavaScript etc. So, it is important to explore using all these methods and then find which one is most efficient and time saving.

Microsoft Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence. It is extremely intuitive and can be used for visualizations and filtering.

The dashboard is created using PowerBI software.

Such a dashboard will remove the need to open Task Scheduler and manually click on each task to find out all its properties. One glance at the dashboard instead gives us all the information.

Another important facet of the POC is to fetch the details of the past run results of each task. This allows us to analyze historical results and see how they have run in the past.

Thus, such a POC will ensure that all tasks on a system are constantly monitored, and it is easy to identify if something goes wrong.

SOFTWARES USED

PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule.

Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.

Python consistently ranks as one of the most popular programming languages.

POWERSHELL

PowerShell is a task automation and configuration management program from Microsoft, consisting of a command-line shell and the associated scripting language. Initially a Windows component only, known as Windows PowerShell, it was made open-source and cross-platform on August 18, 2016, with the introduction of PowerShell Core.[5] The former is built on the .NET Framework, the latter on .NET (previously .NET Core).

Since Windows 10 build 14971, PowerShell replaced Command Prompt and became the default command shell for File Explorer.[6][7]

In PowerShell, administrative tasks are performed via cmdlets (pronounced command-lets), which are specialized .NET classes implementing a particular operation. These work by accessing data in different data stores, like the file system or Windows Registry, which are made available to PowerShell via providers. Third-party developers can add cmdlets and providers to PowerShell.[8][9] Cmdlets may be used by scripts, which may in turn be packaged into modules. Cmdlets work in tandem with the .NET API.

MICROSOFT POWER BI

Microsoft Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence. It is part of the Microsoft Power Platform. Power BI is a collection of software services, apps, and connectors that work together to turn unrelated sources of data into coherent, visually immersive, and interactive insights. Data may be input by reading directly from a database, webpage, or structured files such as spreadsheets, CSV, XML, and JSON.

Power BI provides cloud-based BI (business intelligence) services, known as "Power BI Services", along with a desktop-based interface, called "Power BI Desktop". It offers data warehouse capabilities including data preparation, data discovery, and interactive dashboards. In March 2016, Microsoft released an additional service called Power BI Embedded on its Azure cloud platform. One main differentiator of the product is the ability to load custom visualizations.

METHODOLOGY

TASK SCHEDULER

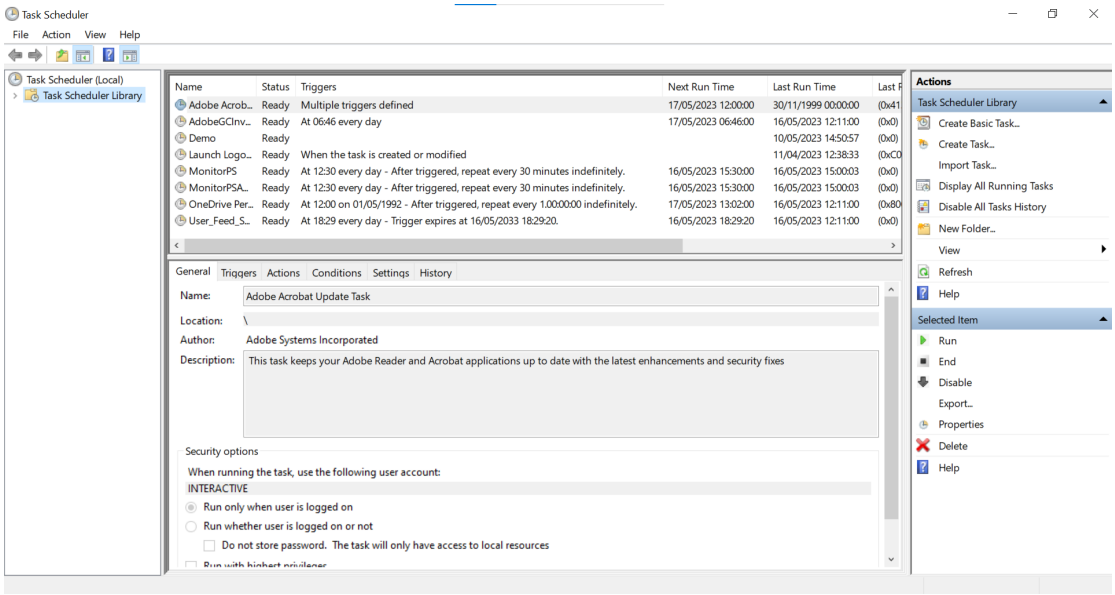


Fig 1.1 – Task Scheduler on Windows 10

USING PYTHON

CODE

A Python script is used to retrieve all information about the tasks including the Task Name, Task Path, Last Run Time, Next Run Time, and Last Run Status.

```python
import win32com.client
import csv
import json
import time

TASK_ENUM_HIDDEN = 1
TASK_STATE = {0: 'Unknown',
              1: 'Disabled',
              2: 'Queued',
              3: 'Ready',
              4: 'Running'}

scheduler = win32com.client.Dispatch('Schedule.Service')
scheduler.Connect()

fields = ['Name', ' Path', 'State',' Last Run Time', ' Next Run Time', ' Result', 'Author']

with open('TasksPy.csv', 'w') as csvfile:
            filewriter = csv.writer(csvfile)
            filewriter.writerow(fields)

a=[[]]

n=0
count=0
failCount=0
runCount=0
unknown=0

folders = [scheduler.GetFolder('\\')]
while folders:
    folder = folders.pop(0)
    folders += list(folder.GetFolders(0))
    tasks = list(folder.GetTasks(TASK_ENUM_HIDDEN))
    n += len(tasks)
    for task in folder.GetTasks(0):
        if task.LastTaskResult == 267011:
            continue
        elif task.LastTaskResult == -2147023829:
            result = 'Failed'
        elif task.LastTaskResult == 267009:
            result = 'Currently Running'
        elif task.LastTaskResult == 0:
            result = 'Completed Successfully'
        else:
            result = 'Unknown'
            settings = task.Definition.Settings

        if(task.path.startswith("\\Microsoft")):
            author = 'System Defined'
        elif (task.path.startswith("\\Adobe")):
            author = 'System Defined'
        else:
            author = 'User Defined'

        print('Name           : %s' % task.Name)
        print('Path           : %s' % task.Path)
        print('State          : %s' % TASK_STATE[task.State])
        print('Last Run Time  : %s' % task.LastRunTime)
        print('Next Run Time  : %s' % task.NextRunTime)
        print("Run Result     :",result)
        print("Author         :",author, '\n')

        if(result=='Completed Successfully'):
            count=count+1

        if(result=='Currently Running'):
            runCount=runCount+1
        if (result=='Failed'):
            failCount=failCount+1
        if (result=='Unknown'):
            unknown=unknown+1

        a=([[task.name, task.path, TASK_STATE[task.State], task.LastRunTime, task.NextRunTime, result,author]])

        with open('TasksPy.csv', 'a') as csvfile:
            filewriter = csv.writer(csvfile)
            filewriter.writerows(a)


print('Listed %d tasks' % n)
print('%d Tasks Successfully Completed' % count)
s=count/(count+runCount+failCount+unknown)
print('%d Tasks Currently Running' % runCount)
print('%d Tasks Failed' % failCount)
print("Success Rate of Tasks: ", s*100 ,'%')

csvfile = open('TasksPy.csv', 'r')
jsonfile = open('Tasks.json', 'w')

reader = csv.DictReader( csvfile, fields)
for row in reader:
    json.dump(row, jsonfile)
    jsonfile.write('\n')
```

Fig 1.2 – Python Code to Retrieve Task Scheduler Task Details

EXPLANATION OF CODE

Firstly, we import the various libraries that will be required to retrieve the tasks, these include win32com, csv etc.

We then define the various states the task can be in namely – Unknown, Disabled, Queued, Ready and Running.

We go task by task defining the run results for each of them based on the status code that we get, for instance 0 corresponds to successful, 267009 refers to running and so on.

For each task, we print the name, path, state, last run time, next run time and last run result.

We even include a formula at the end to calculate exactly how many tasks there are and percentage of tasks which were completed and failed.

We export the entire data to a CSV file. For this we use a 'with open' command and define the various fields in the CSV.

OUTPUT

```
Name           : AdobeGCInvoker-1.0
Path           : \AdobeGCInvoker-1.0
State          : Ready
Last Run Time  : 2023-05-16 12:11:00+00:00
Next Run Time  : 2023-05-17 06:46:00+00:00
Run Result     : Completed Successfully
Author         : System Defined

Name           : Demo
Path           : \Demo
State          : Ready
Last Run Time  : 2023-05-10 14:50:57+00:00
Next Run Time  : 1899-12-30 00:00:00+00:00
Run Result     : Completed Successfully
Author         : User Defined

Name           : Launch Logon Script As Interactive User
Path           : \Launch Logon Script As Interactive User
State          : Ready
Last Run Time  : 2023-04-11 12:38:33+00:00
Next Run Time  : 1899-12-30 00:00:00+00:00
Run Result     : Unknown
Author         : User Defined

Name           : MonitorPS
Path           : \MonitorPS
State          : Ready
Last Run Time  : 2023-05-16 15:00:03+00:00
Next Run Time  : 2023-05-16 15:30:00+00:00
Run Result     : Completed Successfully
Author         : User Defined
```

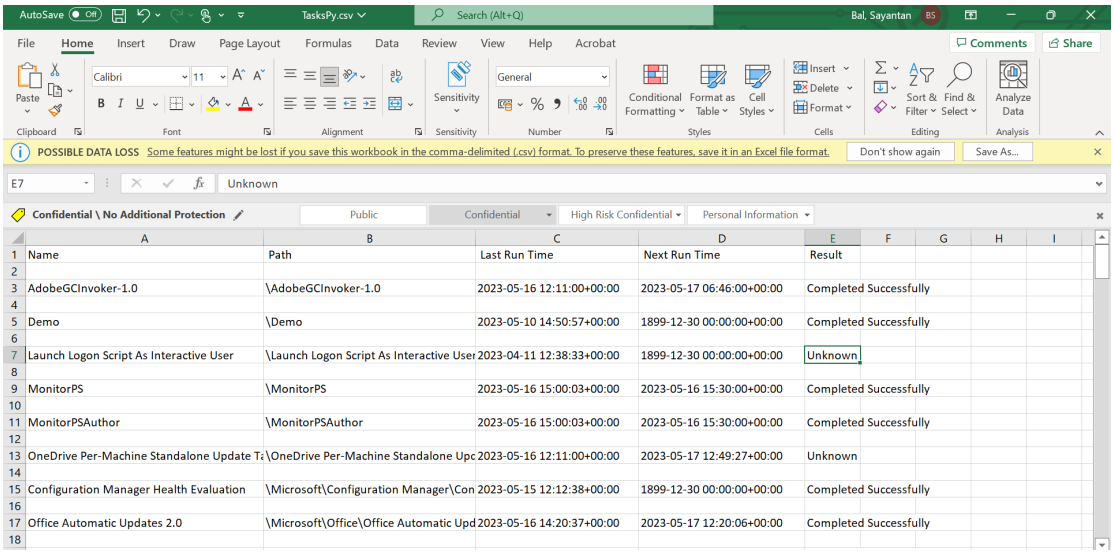
```
Name           : appuriverifierdaily
Path           : \Microsoft\Windows\ApplicationData\appuriverifierdaily
State          : Ready
Last Run Time  : 2023-05-16 14:20:39+00:00
Next Run Time  : 1899-12-30 00:00:00+00:00
Run Result     : Completed Successfully
Author         : System Defined

Name           : appuriverifierinstall
Path           : \Microsoft\Windows\ApplicationData\appuriverifierinstall
State          : Ready
Last Run Time  : 2023-05-15 12:06:36+00:00
Next Run Time  : 1899-12-30 00:00:00+00:00
Run Result     : Completed Successfully
Author         : System Defined

Name           : CleanupTemporaryState
Path           : \Microsoft\Windows\ApplicationData\CleanupTemporaryState
State          : Ready
Last Run Time  : 2023-05-15 12:12:34+00:00
Next Run Time  : 1899-12-30 00:00:00+00:00
Run Result     : Completed Successfully
Author         : System Defined

Name           : DsSvcCleanup
Path           : \Microsoft\Windows\ApplicationData\DsSvcCleanup
State          : Ready
Last Run Time  : 2023-05-15 12:12:37+00:00
Next Run Time  : 1899-12-30 00:00:00+00:00
Run Result     : Completed Successfully
Author         : System Defined
```

```
Listed 163 tasks
61 Tasks Successfully Completed
2 Tasks Currently Running
2 Tasks Failed
Success Rate of Tasks:  80.26315789473685 %
```

Fig 1.3 – Output of Python Script

CSV

Next, all the Task Details are exported to a CSV File for easy readability and accessibility



	A	B	C	D	E
1	Name	Path	Last Run Time	Next Run Time	Result
2					
3	AdobeGCInvoker-1.0	\AdobeGCInvoker-1.0	2023-05-16 12:11:00+00:00	2023-05-17 06:46:00+00:00	Completed Successfully
4					
5	Demo	\Demo	2023-05-10 14:50:57+00:00	1899-12-30 00:00:00+00:00	Completed Successfully
6					
7	Launch Logon Script As Interactive User	\Launch Logon Script As Interactive User	2023-04-11 12:38:33+00:00	1899-12-30 00:00:00+00:00	Unknown
8					
9	MonitorPS	\MonitorPS	2023-05-16 15:00:03+00:00	2023-05-16 15:30:00+00:00	Completed Successfully
10					
11	MonitorPSAuthor	\MonitorPSAuthor	2023-05-16 15:00:03+00:00	2023-05-16 15:30:00+00:00	Completed Successfully
12					
13	OneDrive Per-Machine Standalone Update Ta	\OneDrive Per-Machine Standalone Upc	2023-05-16 12:11:00+00:00	2023-05-17 12:49:27+00:00	Unknown
14					
15	Configuration Manager Health Evaluation	\Microsoft\Configuration Manager\Con	2023-05-15 12:12:38+00:00	1899-12-30 00:00:00+00:00	Completed Successfully
16					
17	Office Automatic Updates 2.0	\Microsoft\Office\Office Automatic Upd	2023-05-16 14:20:37+00:00	2023-05-17 12:20:06+00:00	Completed Successfully
18					

Fig 1.4 – CSV File

USING POWERSHELL

PowerShell is a much more powerful tool than Python and it can be used to directly retrieve the details of the tasks without any need to download Python separately, so it was preferred.

CODE 1

First, a PowerShell Script which retrieves the Task Name, Task Path, Last Run Time, Next Run Time, and Last Run Result Code was implemented.

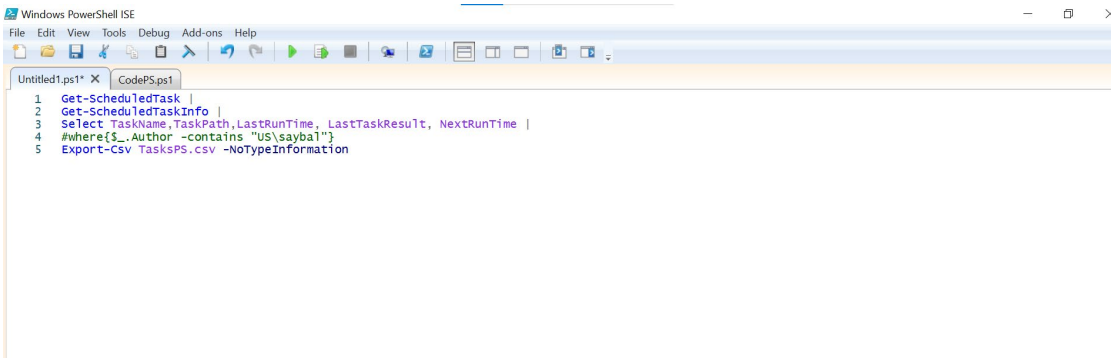
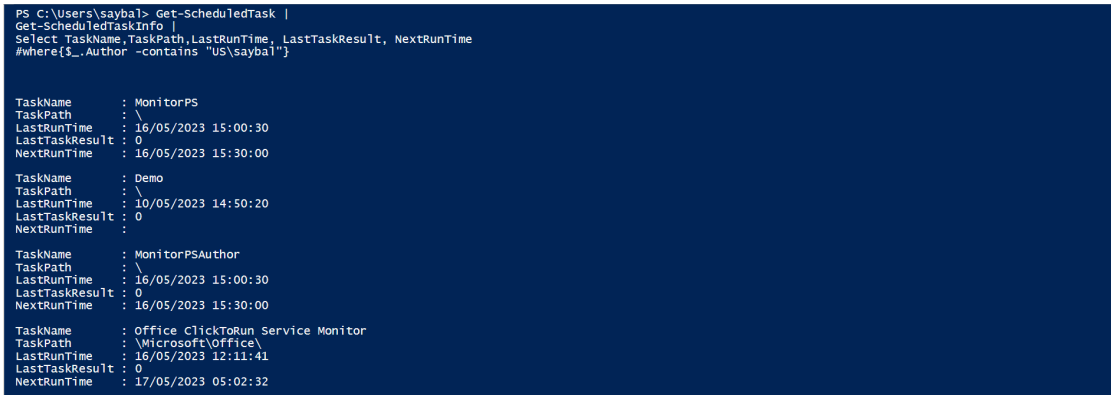


Fig 1.5 – PowerShell Script to Retrieve Task Details

It successfully fetches the details of the tasks.



```
PS C:\Users\saybal> Get-ScheduledTask |
Get-ScheduledTaskInfo |
Select TaskName,TaskPath,LastRunTime, LastTaskResult, NextRunTime
#where{$_.Author -contains "US\saybal"}


TaskName       : MonitorPS
TaskPath       : \
LastRunTime    : 16/05/2023 15:00:30
LastTaskResult : 0
NextRunTime    : 16/05/2023 15:30:00

TaskName       : Demo
TaskPath       : \
LastRunTime    : 10/05/2023 14:50:20
LastTaskResult : 0
NextRunTime    :

TaskName       : MonitorPSAuthor
TaskPath       : \
LastRunTime    : 16/05/2023 15:00:30
LastTaskResult : 0
NextRunTime    : 16/05/2023 15:30:00

TaskName       : Office ClickToRun Service Monitor
TaskPath       : \Microsoft\Office\
LastRunTime    : 16/05/2023 12:11:41
LastTaskResult : 0
NextRunTime    : 17/05/2023 05:02:32
```

```
TaskName       : Office Feature Updates Logon
TaskPath       : \Microsoft\Office\
LastRunTime    : 15/05/2023 13:10:40
LastTaskResult : 0
NextRunTime    :

TaskName       : .NET Framework NGEN v4.0.30319 64 Critical
TaskPath       : \Microsoft\Windows\.NET Framework\
LastRunTime    : 24/04/2023 11:58:28
LastTaskResult : 0
NextRunTime    :

TaskName       : OfficeTelemetryAgentFallBack2016
TaskPath       : \Microsoft\Office\
LastRunTime    : 16/05/2023 12:38:08
LastTaskResult : 0
NextRunTime    :

TaskName       : .NET Framework NGEN v4.0.30319 Critical
TaskPath       : \Microsoft\Windows\.NET Framework\
LastRunTime    : 24/04/2023 11:58:28
LastTaskResult : 0
NextRunTime    : |

TaskName       : Microsoft Compatibility Appraiser
TaskPath       : \Microsoft\Windows\Application Experience\
LastRunTime    : 16/05/2023 12:11:41
LastTaskResult : 2147483658
NextRunTime    : 17/05/2023 04:44:14
```

Fig 1.6 – PowerShell Output

The information is exported to a CSV File for easy readability and accessibility.

	A	B	C	D	E	F	G	H
1	TaskName	TaskPath	LastRunTime	LastTaskResult	NextRunTime			
2	OneDrive Per-Machine Standalone Update Task	\	16/05/2023 12:11	2147806724	17/05/2023 12:36			
3	Demo	\	10/05/2023 14:50	0				
4	Configuration Manager Health Evaluation	\Microsoft\Configuration Manager\	15/05/2023 12:12	0				
5	MonitorPSAuthor	\	16/05/2023 15:00	267009	16/05/2023 15:30			
6	MonitorPS	\	16/05/2023 15:00	267009	16/05/2023 15:30			
7	AdobeGCInvoker-1.0	\	16/05/2023 12:11	0	17/05/2023 06:46			
8	Adobe Acrobat Update Task	\	30/11/1999 00:00	267011	17/05/2023 12:00			
9	Launch Logon Script As Interactive User	\	11/04/2023 12:38	3221225786				
10	User_Feed_Synchronization-{0A110474-3D1F-4BA3-AD4F-6EAB7E7178AF}	\	16/05/2023 12:11	0	16/05/2023 18:29			
11	Office Serviceability Manager	\Microsoft\Office\	30/11/1999 00:00	267011	16/05/2023 18:25			
12	Office ClickToRun Service Monitor	\Microsoft\Office\	16/05/2023 12:11	0	17/05/2023 07:17			
13	Office Automatic Updates 2.0	\Microsoft\Office\	16/05/2023 14:20	0	17/05/2023 13:23			
14	Office Performance Monitor	\Microsoft\Office\	30/11/1999 00:00	267011				
15	.NET Framework NGEN v4.0.30319 64 Critical	\Microsoft\Windows\.NET Framework\	24/04/2023 11:58	0				
16	OfficeTelemetryAgentLogOn2016	\Microsoft\Office\	15/05/2023 11:59	0				
17	Office Feature Updates Logon	\Microsoft\Office\	15/05/2023 13:10	0				
18	.NET Framework NGEN v4.0.30319 64	\Microsoft\Windows\.NET Framework\	15/05/2023 12:12	0				

Confidential \ No Additional Protection | Public | Confidential | High Risk Confidential | Personal Information

	A	B	C	D	E	F	G	H
30	StartupAppTask	\Microsoft\Windows\Application Experienc	15/05/2023 12:12	0				
31	DsSvcCleanup	\Microsoft\Windows\ApplicationData\	15/05/2023 12:12	0				
32	Proxy	\Microsoft\Windows\Autochk\	16/05/2023 14:20	0				
33	Pre-staged app cleanup	\Microsoft\Windows\AppxDeploymentClie	03/01/2023 01:49	0				
34	CleanupTemporaryState	\Microsoft\Windows\ApplicationData\	15/05/2023 12:12	0				
35	BitLocker Encrypt All Drives	\Microsoft\Windows\BitLocker\	30/11/1999 00:00	267011				
36	BitLocker MDM policy Refresh	\Microsoft\Windows\BitLocker\	30/11/1999 00:00	267011				
37	appuriverifierdaily	\Microsoft\Windows\ApplicationData\	16/05/2023 14:20	0				
38	Backup	\Microsoft\Windows\AppListBackup\	15/05/2023 12:12	0				
39	UninstallDeviceTask	\Microsoft\Windows\Bluetooth\	30/11/1999 00:00	267011				
40	BgTaskRegistrationMaintenanceTask	\Microsoft\Windows\BrokerInfrastructure\	15/05/2023 12:12	268435456				
41	PcaPatchDbTask	\Microsoft\Windows\Application Experienc	16/05/2023 12:11	0	17/05/2023 04:51			
42	UserTask	\Microsoft\Windows\CertificateServicesClie	16/05/2023 14:20	0				
43	ProactiveScan	\Microsoft\Windows\Chkdsk\	15/05/2023 12:12	0				
44	UserTask-Roam	\Microsoft\Windows\CertificateServicesClie	15/05/2023 12:55	0				
45	SyspartRepair	\Microsoft\Windows\Chkdsk\	30/11/1999 00:00	267011				
46	CreateObjectTask	\Microsoft\Windows\CloudExperienceHost	04/01/2023 17:43	0				

Fig 1.7 – PowerShell to CSV

CODE 2

Another PowerShell Script is used to fetch the details of the authors separately. This helps us to filter the tasks that we create ourselves and monitor them separately.

The script uses the Get-ScheduledTask cmdlet to retrieve information about the tasks, and the Get-ScheduledTaskInfo cmdlet to retrieve additional information about the last run of each task. The script loops through each scheduled task, retrieves its information,

Untitled1.ps1* | CodePSAuthor.ps1

```
Get-ScheduledTask |
#Where-Object {$_.Author -contains 'US\saybal'} |
Select TaskName, Author |
Export-Csv TasksPSAuthor.csv -NoTypeInformation
```

Fig 1.8 – PowerShell Script to fetch Name and Author

The output is successful

```
PS C:\Users\saybal> Get-ScheduledTask |
#Where-Object {$_.Author -contains 'US\saybal'} |
Select TaskName, Author

TaskName                                                    Author
--------                                                    ------
Adobe Acrobat Update Task                                   Adobe Systems Incorporated
AdobeGCInvoker-1.0                                          Adobe Systems Incorporated
Demo                                                        US\saybal
Launch Logon Script As Interactive User
MonitorPS                                                   US\saybal
MonitorPSAuthor                                             US\saybal
OneDrive Per-Machine Standalone Update Task                 Microsoft Corporation
User_Feed_Synchronization-{0A110474-3D1F-4BA3-AD4F-6EAB7E7178AF} US\saybal
Configuration Manager Health Evaluation                     Microsoft
Office Automatic Updates 2.0                                Microsoft Office
Office ClickToRun Service Monitor                           Microsoft Office
Office Feature Updates                                      Microsoft Office
Office Feature Updates Logon                                Microsoft Office
Office Performance Monitor
Office Serviceability Manager                               Microsoft Office
```

```
Background Synchronization                          Microsoft Corporation
Logon Synchronization                               Microsoft Corporation
Device Install Group Policy                         Microsoft Corporation
Device Install Reboot Required                      Microsoft Corporation
Sysprep Generalize Drivers                          Microsoft Corporation
AnalyzeSystem                                       Microsoft Corporation
EduPrintProv
PrinterCleanupTask
VerifyWinRE                                         Microsoft Corporation
RegIdleBackup                                       Microsoft Corporation
StartComponentCleanup
BackgroundUploadTask
NetworkStateChangeTask
Account Cleanup
CreateObjectTask                                    Microsoft Corporation
FamilySafetyMonitor                                 Microsoft Corporation
FamilySafetyRefreshTask                             Microsoft Corporation
IndexerAutomaticMaintenance                         Microsoft Corporation
ThemesSyncedImageDownload                           Microsoft Corporation
SvcRestartTaskLogon                                 Microsoft Corporation
SpaceAgentTask                                      Microsoft Corporation
SpaceManagerTask                                    $(@%SystemRoot%\system32\spaceman.exe,-2)
MaintenanceTasks                                    $(@%SystemRoot%\system32\Windows.StateRepositoryClien...
Storage Tiers Management Initialization             Microsoft Corporation
Storage Tiers Optimization                          Microsoft Corporation
EnableLicenseAcquisition                            Microsoft Corporation
```

Fig 1.9 – PowerShell Output

A1 | TaskName

Confidential \ No Additional Protection | Public | Confidential | High Risk Confidential | Personal Information

	A	B	C	D	E	F	G
1	TaskName	Author					
2	Adobe Acrobat Update Task	Adobe Systems Incorporated					
3	AdobeGCInvoker-1.0	Adobe Systems Incorporated					
4	Demo	US\saybal					
5	Launch Logon Script As Interactive User						
6	MonitorPS	US\saybal					
7	MonitorPSAuthor	US\saybal					
8	OneDrive Per-Machine Standalone Update Task	Microsoft Corporation					
9	User_Feed_Synchronization-{0A110474-3D1F-4BA3-AD4F-6EAB7E7178AF}	US\saybal					
10	Configuration Manager Health Evaluation	Microsoft					
11	Office Automatic Updates 2.0	Microsoft Office					
12	Office ClickToRun Service Monitor	Microsoft Office					
13	Office Feature Updates	Microsoft Office					
14	Office Feature Updates Logon	Microsoft Office					
15	Office Performance Monitor						
16	Office Serviceability Manager	Microsoft Office					
17	OfficeTelemetryAgentFallBack2016						
18	OfficeTelemetryAgentLogOn2016						

Confidential \ No Additional Protection | Public | Confidential | High Risk Confidential | Personal Information

	A	B	C	D	E	F	G
33	CleanupTemporaryState	Microsoft Corporation					
34	DsSvcCleanup	Microsoft Corporation					
35	Backup	$(@%SystemRoot%\system32\AppListBackupLauncher.dll,-600)					
36	Pre-staged app cleanup						
37	Proxy	Microsoft Corporation					
38	BitLocker Encrypt All Drives						
39	BitLocker MDM policy Refresh						
40	UninstallDeviceTask	Microsoft					
41	BgTaskRegistrationMaintenanceTask	Microsoft Corporation					
42	UserTask	Microsoft Corporation					
43	UserTask-Roam	Microsoft Corporation					
44	ProactiveScan	Microsoft Corporation					
45	SyspartRepair						
46	LicenseImdsIntegration	$(@%SystemRoot%\system32\fclip.exe,-100)					
47	CreateObjectTask						
48	Consolidator	Microsoft Corporation					
49	UsbCeip	Microsoft Corporation					
50	Data Integrity Check And Scan	Microsoft Corporation					

Fig 2.0 – CSV File

SCHEDULING TASKS

Once we can retrieve the details of the Tasks including the Author, it is necessary to schedule a task that will run these PowerShell Codes periodically.

This will ensure that there is no need to manually run the code every time we want to monitor our tasks.

For this to work smoothly, we need to create exe files for both scripts which will ensure that the codes are able to be run with the help of just a double click.

For this, we import a to-csv module in PowerShell which automatically converts the .PS file to an executable.

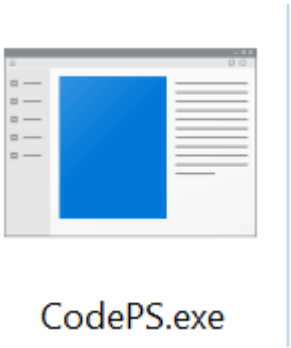
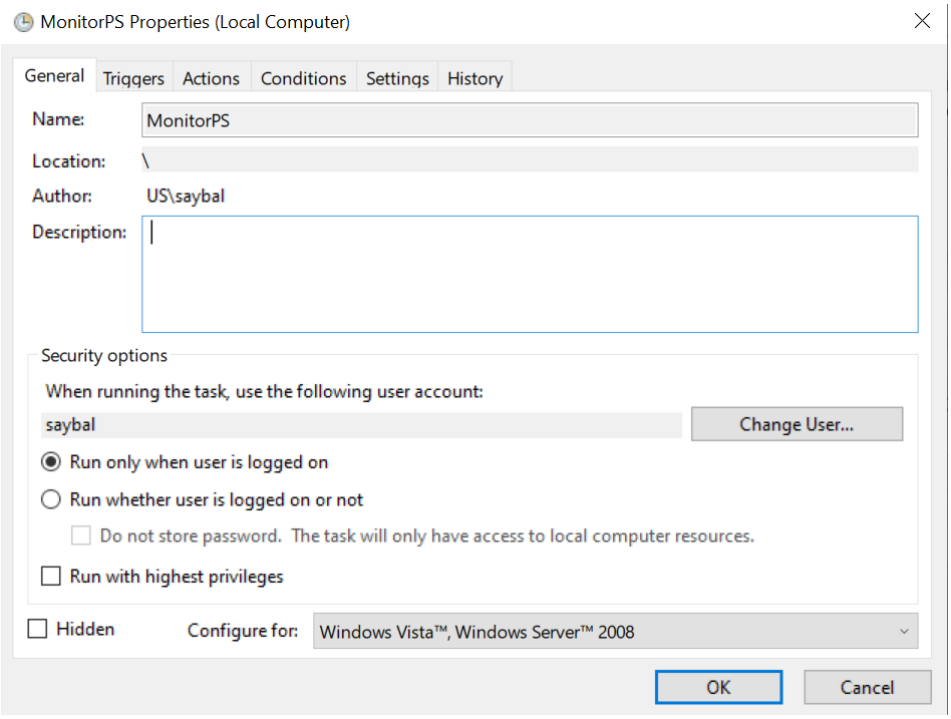


Fig 2.1 – Exe File

Both scripts which fetch the task details, and the authors are scheduled on Task Scheduler to run every 30 minutes. The report is ensured to be never out of date.



Trigger	Details	Status
Daily	At 12:30 every day - After triggered, repeat every 30 minutes indef...	Enabled

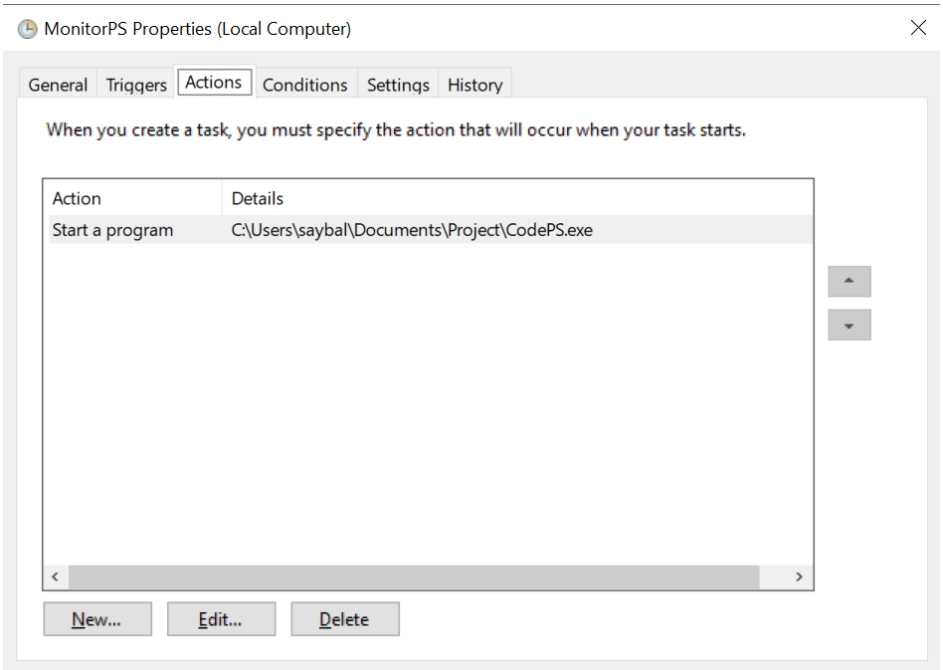
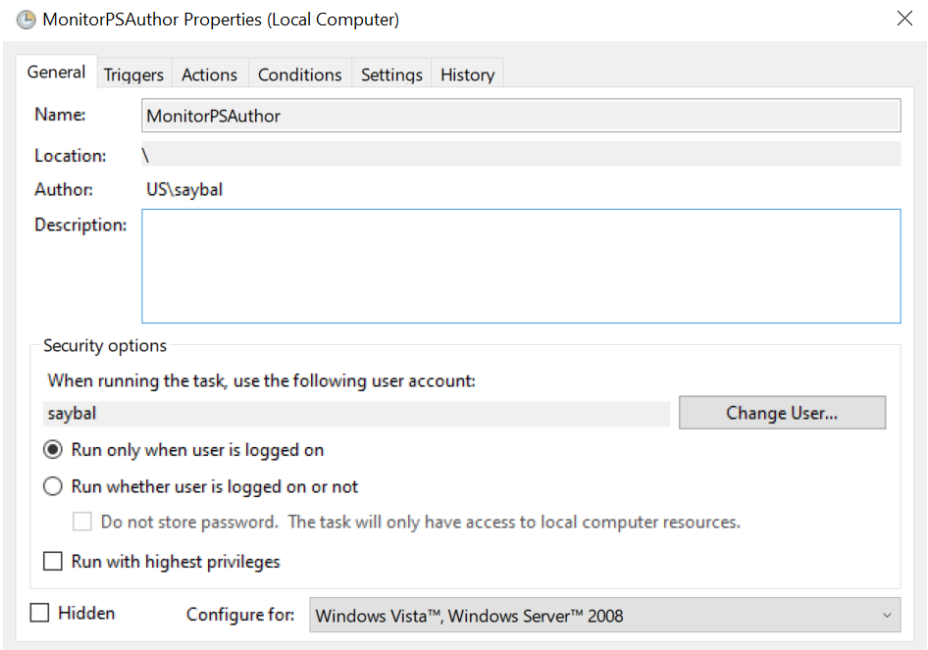


Fig 2.3 – Code 1 scheduled to run every 30 minutes



Trigger	Details	Status
Daily	At 12:30 every day - After triggered, repeat every 30 minutes indef...	Enabled

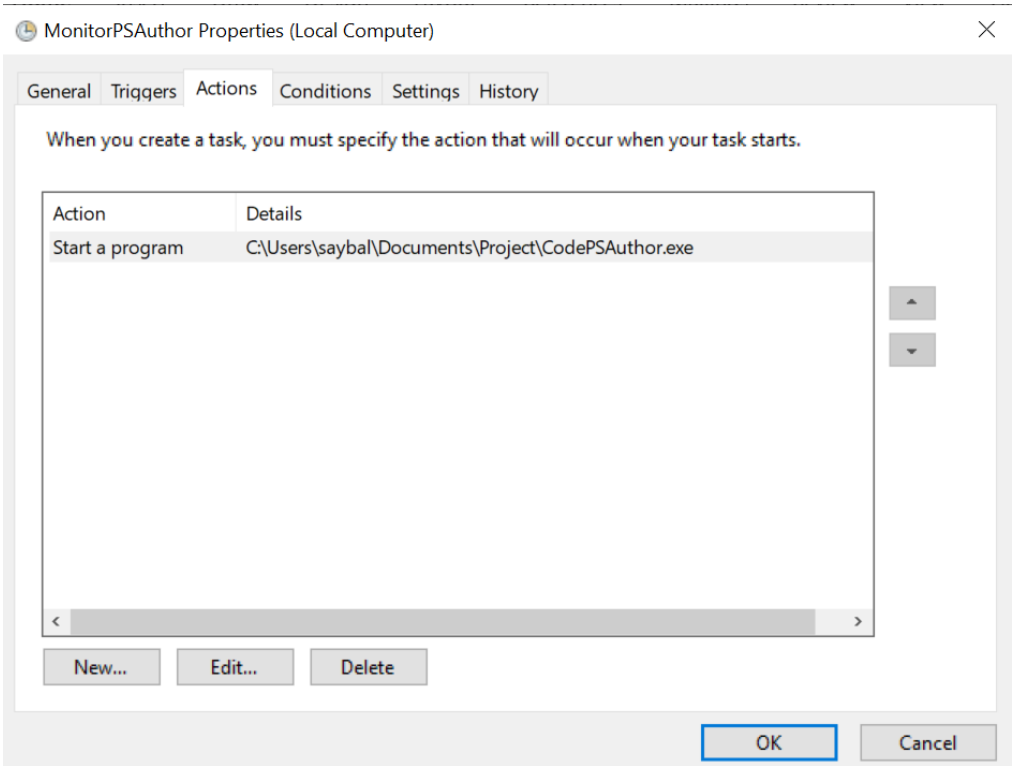


Fig 2.4 – Scheduling Code 2

We now have two different CSV files each of which update every 30 minutes from the time they were scheduled. The next step is to go into PowerBI software and create a dashboard there.

First, however we need to create a connection between the two files so that the system can recognize that the tasks in both the CSV Files are the same. Such a relationship can also be configured using PowerBI.

Then we can proceed with the creation of the various tabs based on whether the task was completed successfully, currently running or failed.

GETTING INTO POWERBI

IMPORTING DATA

Data from both the CSV Files is imported into PowerBI using the Get Data option.

TasksPS.csv

File Origin: 1252: Western European (Windows)
Delimiter: Comma
Data Type Detection: Based on first 200 rows

TaskName	TaskPath	LastRunTime	LastTaskResult	NextRunTime
Adobe Acrobat Update Task	\	30/11/1999 00:00:30	267011	17/05/2023 12:00:3
AdobeGCInvoker-1.0	\	16/05/2023 12:11:41	0	17/05/2023 06:46:1
User_Feed_Synchronization-{0A110474-3D1F-4BA3-AD...	\	16/05/2023 12:11:41	0	16/05/2023 18:29:5
MonitorPSAuthor	\	16/05/2023 15:30:00	267009	16/05/2023 16:00:3
Configuration Manager Health Evaluation	\Microsoft\Configuration Manager\	15/05/2023 12:12:42	0	nu
MonitorPS	\	16/05/2023 15:30:00	267009	16/05/2023 16:00:3
Demo	\	10/05/2023 14:50:20	0	nu
Launch Logon Script As Interactive User	\	11/04/2023 12:38:08	3221225786	nu
Office Performance Monitor	\Microsoft\Office\	30/11/1999 00:00:30	267011	nu
Office ClickToRun Service Monitor	\Microsoft\Office\	16/05/2023 12:11:41	0	17/05/2023 04:19:4
OneDrive Per-Machine Standalone Update Task	\	16/05/2023 12:11:41	2147806724	17/05/2023 13:27:5
.NET Framework NGEN v4.0.30319 64	\Microsoft\Windows\.NET Framework\	15/05/2023 12:12:42	0	nu
OfficeTelemetryAgentLogOn2016	\Microsoft\Office\	15/05/2023 11:59:29	0	nu
.NET Framework NGEN v4.0.30319	\Microsoft\Windows\.NET Framework\	15/05/2023 12:12:42	0	nu
Office Feature Updates	\Microsoft\Office\	15/05/2023 12:02:32	2147946720	16/05/2023 20:15:4
.NET Framework NGEN v4.0.30319 64 Critical	\Microsoft\Windows\.NET Framework\	24/04/2023 11:58:28	0	nu
Office Automatic Updates 2.0	\Microsoft\Office\	16/05/2023 14:20:50	0	17/05/2023 04:57:2
Office Feature Updates Logon	\Microsoft\Office\	15/05/2023 13:10:40	0	nu
Office Serviceability Manager	\Microsoft\Office\	30/11/1999 00:00:30	267011	16/05/2023 18:25:5
OfficeTelemetryAgentFallBack2016	\Microsoft\Office\	16/05/2023 12:38:08	0	nu

TasksPSAuthor.csv

File Origin: 1252: Western European (Windows)
Delimiter: Comma
Data Type Detection: Based on first 200 rows

Column1	Column2
TaskName	Author
Adobe Acrobat Update Task	Adobe Systems Incorporated
AdobeGCInvoker-1.0	Adobe Systems Incorporated
Demo	US\saybal
Launch Logon Script As Interactive User	
MonitorPS	US\saybal
MonitorPSAuthor	US\saybal
OneDrive Per-Machine Standalone Update Task	Microsoft Corporation
User_Feed_Synchronization-{0A110474-3D1F-4BA3-AD...	US\saybal
Configuration Manager Health Evaluation	Microsoft
Office Automatic Updates 2.0	Microsoft Office
Office ClickToRun Service Monitor	Microsoft Office
Office Feature Updates	Microsoft Office
Office Feature Updates Logon	Microsoft Office
Office Performance Monitor	
Office Serviceability Manager	Microsoft Office
OfficeTelemetryAgentFallBack2016	
OfficeTelemetryAgentLogOn2016	
.NET Framework NGEN v4.0.30319	
.NET Framework NGEN v4.0.30319 64	

Fig 2.5 – Importing Data into PowerBI

All blank rows are defaulted to NULL values on PowerBI.

Tasks which have not been scheduled are assigned an arbitrary Next Run Time of 1899-12-30 00:00:30.

Tasks which have not yet run are assigned an arbitrary Last Run Time of 1999-12-30 00:00:30.

DEFINING RUN RESULTS

When we use PowerShell, there are no run results which are directly visible. We get status codes each of which correspond to a different run result.

0 – Completed Successfully

2670009 – Currently Running

267011 – Yet to Run

2147xxxxxx – Failed

All Tasks which do not fall into these categories are assigned a run result of undefined.

The Run Result column is manually added in PowerBI using a DAX Code.

```
= Table.AddColumn(#"Renamed Columns", "Result", each if [Last Task Result] = 0
    then "Completed Successfully"
else if [Last Task Result] = 267009 then "Currently Running"
else if [Last Task Result] = 267011 then "Yet to Run"
else if [Last Task Result] > 2147000000 and [Last Task Result] < 2148000000
    then "Failed"
else "Unknown")
```

Fig 2.6 – DAX Code to define run results

The columns are also renamed to more accurately describe what they contain.

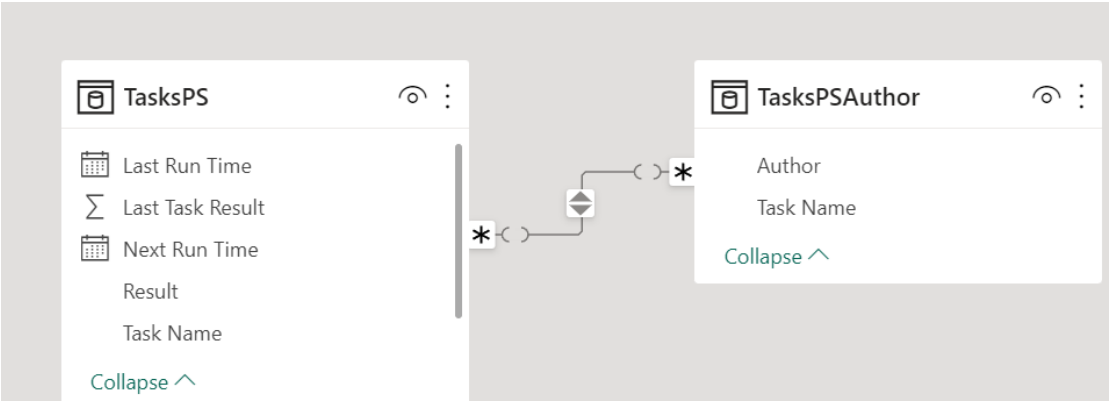
CREATING A RELATIONSHIP

To create a relationship between the 2 CSV Files, we need to find a common denominator between them. In this case, the names of the tasks are common in both the files.

So, we can define a many-to-many relationship between them by linking them using the Task Name.

A many-to-many relationship is a type of cardinality that refers to the relationship between two entities, say, A and B, where A may contain a parent instance for which there are many children in B and vice versa.

The relationship goes both ways as shown in the figure below.



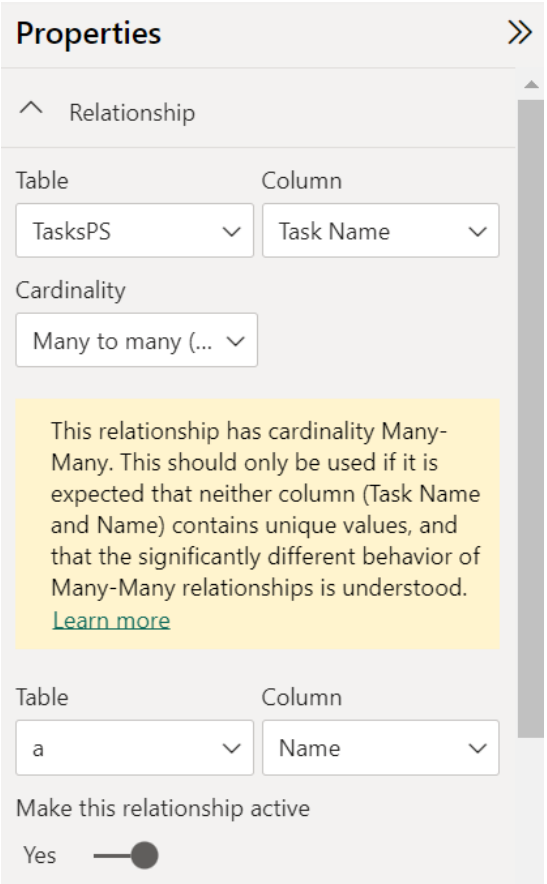


Fig 2.7 – Relationship between 2 Files

This relationship has cardinality Many-Many. This should only be used if it is expected that neither column (Task Name and Name) contains unique values.

DASHBOARD

Now, we move on to creating the Dashboard to monitor all the tasks.

We are going to create separate tabs for tasks that have been successfully completed, failed, and currently running.

Another tab is used for unscheduled tasks while a fifth tab is used to display user defined tasks. User defined refers to those which have been defined by us and not running automatically in the system. Often it is these tasks which need to be monitored the most closely hence the use of a separate tab.

The dashboard is created in a tabular format.

To create the individual views, we make use of the filtering system. For completed successfully, we only include those tasks which have a run result of 0, for running, we use the code 267009 and so on.

For the unscheduled jobs, we use filtering of the Next Run Time column and those for which the column is blank is used for this table.

The user defined tasks are filtered from the Author column. Those tasks for which the author is 'us\saybal,' which is the name of the system on which we are performing this program, are said to be user defined. All other pertains to system defined.

Each table has five columns – Task Name, Path, Last Run Time, Next Run Time, Result

COMPLETED SUCCESSFULLY

TASKS COMPLETED SUCCESSFULLY

Task Name	Result	Last Run Time	Next Run Time	Author
.NET Framework NGEN v4.0.30319	Completed Successfully	15/05/2023 12:12:42		
.NET Framework NGEN v4.0.30319 64	Completed Successfully	15/05/2023 12:12:42		
.NET Framework NGEN v4.0.30319 64 Critical	Completed Successfully	24/04/2023 11:58:28		
.NET Framework NGEN v4.0.30319 Critical	Completed Successfully	24/04/2023 11:58:28		
Adobe Acrobat Update Task	Completed Successfully	16/05/2023 17:12:42	18/05/2023 12:00:30	Adobe Systems Incorporated
AdobeGCInvoker-1.0	Completed Successfully	17/05/2023 11:56:26	18/05/2023 06:46:16	Adobe Systems Incorporated
AnalyzeSystem	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
appuriverifierdaily	Completed Successfully	16/05/2023 14:20:50		Microsoft Corporation
appuriverifierinstall	Completed Successfully	15/05/2023 12:06:36		Microsoft Corporation
Automatic-Device-Join	Completed Successfully	17/05/2023 11:55:25		
BackgroundUploadTask	Completed Successfully	15/05/2023 12:12:42		
Backup	Completed Successfully	15/05/2023 12:12:42		$(@%SystemRoot%\system32\AppListBackupLauncher.dll,-600)
Calibration Loader	Completed Successfully	17/05/2023 11:54:24		Microsoft Corporation
CDSSync	Completed Successfully	17/05/2023 11:55:25		
CleanupTemporaryState	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
Configuration Manager Health Evaluation	Completed Successfully	15/05/2023 12:12:42		Microsoft
Consolidator	Completed Successfully	17/05/2023 12:00:30	17/05/2023 18:00:30	Microsoft Corporation
CreateObjectTask	Completed Successfully	04/01/2023 17:43:13		
CreateObjectTask	Completed Successfully	04/01/2023 17:43:13		Microsoft Corporation
CreateObjectTask	Completed Successfully	17/05/2023 11:55:25		
CreateObjectTask	Completed Successfully	17/05/2023 11:55:25		Microsoft Corporation

TASKS COMPLETED SUCCESSFULLY

Task Name	Result	Last Run Time	Next Run Time	Author
DmClient	Completed Successfully	15/05/2023 12:12:42		Microsoft Windows Feedback
DmClientOnScenarioDownload	Completed Successfully	16/05/2023 17:15:45		Microsoft Windows Feedback
DsSvcCleanup	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
DXGIAdapterCache	Completed Successfully	17/05/2023 11:53:23		
EnableLicenseAcquisition	Completed Successfully	17/05/2023 11:53:23		Microsoft Corporation
ExploitGuard MDM policy Refresh	Completed Successfully	17/05/2023 11:55:25		Microsoft Corporation
File History (maintenance mode)	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
IndexerAutomaticMaintenance	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
Installation	Completed Successfully	16/05/2023 17:15:45		Microsoft Corporation
LicenseAcquisition	Completed Successfully	17/05/2023 11:56:26	18/05/2023 07:30:00	Microsoft Corporation
LocalUserSyncDataAvailable	Completed Successfully	05/01/2023 14:59:29		
Logon	Completed Successfully	16/05/2023 13:30:00		
LPRemove	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
MaintenanceTasks	Completed Successfully	15/05/2023 12:12:42		$(@%SystemRoot%\system32\Windows.StateRepositoryClient.dll,-600)
Microsoft-Windows-DiskDiagnosticDataCollector	Completed Successfully	16/02/2023 11:41:11		Microsoft Corporation
MsCtfMonitor	Completed Successfully	17/05/2023 11:54:24		
Office Automatic Updates 2.0	Completed Successfully	17/05/2023 11:59:29	18/05/2023 10:48:18	Microsoft Office
Office ClickToRun Service Monitor	Completed Successfully	17/05/2023 11:56:26	18/05/2023 06:07:37	Microsoft Office
Office Feature Updates	Completed Successfully	16/05/2023 17:19:49	17/05/2023 16:02:32	Microsoft Office

Fig 2.8 – Completed Tasks

FAILED

TASKS FAILED

Task Name	Result	Last Run Time	Next Run Time	Author
Microsoft Compatibility Appraiser	Failed	17/05/2023 11:56:26	18/05/2023 04:36:06	$(@%SystemRoot%\system32\compattelrunner.exe,-501)
OneDrive Per-Machine Standalone Update Task	Failed	16/05/2023 12:11:41	18/05/2023 15:00:30	Microsoft Corporation
PrinterCleanupTask	Failed	15/05/2023 13:00:30	14/06/2023 12:00:30	
ProcessMemoryDiagnosticEvents	Failed	13/01/2023 16:41:11		Microsoft Corporation
RunFullMemoryDiagnostic	Failed	13/01/2023 16:41:11		Microsoft Corporation
SilentCleanup	Failed	15/05/2023 12:32:02		Microsoft Corporation
ThemesSyncedImageDownload	Failed	15/05/2023 12:12:42		Microsoft Corporation
WinSAT	Failed	12/01/2023 11:41:11		Microsoft

Fig 2.9 – Tasks Failed

RUNNING

TASKS CURRENTLY RUNNING

Task Name	Result	Last Run Time	Next Run Time	Author
CacheTask	Currently Running	17/05/2023 11:54:24		Microsoft
MonitorPS	Currently Running	17/05/2023 12:30:00	17/05/2023 13:00:30	US\saybal
MonitorPSAuthor	Currently Running	17/05/2023 12:30:00	17/05/2023 13:00:30	US\saybal
SystemSoundsService	Currently Running	17/05/2023 11:54:24		

Fig 3.0 – Tasks Running

UNSCHEDHULED

UNSCHEDULED TASKS

Task Name	Result	Last Run Time	Next Run Time	Author
.NET Framework NGEN v4.0.30319	Completed Successfully	15/05/2023 12:12:42		
.NET Framework NGEN v4.0.30319 64	Completed Successfully	15/05/2023 12:12:42		
.NET Framework NGEN v4.0.30319 64 Critical	Completed Successfully	24/04/2023 11:58:28		
.NET Framework NGEN v4.0.30319 Critical	Completed Successfully	24/04/2023 11:58:28		
Account Cleanup	Yet to Run	30/11/1999 00:00:30		
AD RMS Rights Policy Template Management (Manual)	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
AnalyzeSystem	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
appuriverifierdaily	Completed Successfully	16/05/2023 14:20:50		Microsoft Corporation
appuriverifierinstall	Completed Successfully	15/05/2023 12:06:36		Microsoft Corporation
Automatic-Device-Join	Completed Successfully	17/05/2023 11:55:25		
BackgroundUploadTask	Completed Successfully	15/05/2023 12:12:42		
Backup	Completed Successfully	15/05/2023 12:12:42		$(@%SystemRoot%\system32\AppListBackupLauncher.dll,-600)
BfeOnServiceStartTypeChange	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
BgTaskRegistrationMaintenanceTask	Unknown	15/05/2023 12:12:42		Microsoft Corporation
BitLocker Encrypt All Drives	Yet to Run	30/11/1999 00:00:30		
BitLocker MDM policy Refresh	Yet to Run	30/11/1999 00:00:30		
CacheTask	Currently Running	17/05/2023 11:54:24		Microsoft
Calibration Loader	Completed Successfully	17/05/2023 11:54:24		Microsoft Corporation
CDSSync	Completed Successfully	17/05/2023 11:55:25		
Cellular	Yet to Run	30/11/1999 00:00:30		
CleanupTemporaryState	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation

UNSCHEDULED TASKS

Task Name	Result	Last Run Time	Next Run Time	Author
EDP Auth Task	Yet to Run	30/11/1999 00:00:30		
EDP Inaccessible Credentials Task	Yet to Run	30/11/1999 00:00:30		
EduPrintProv	Yet to Run	30/11/1999 00:00:30		
EnableLicenseAcquisition	Completed Successfully	17/05/2023 11:53:23		Microsoft Corporation
ExploitGuard MDM policy Refresh	Completed Successfully	17/05/2023 11:55:25		Microsoft Corporation
FamilySafetyMonitor	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
FamilySafetyRefreshTask	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
File History (maintenance mode)	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
ForceSynchronizeTime	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
GatherNetworkInfo	Yet to Run	30/11/1999 00:00:30		Microsoft
HybridDriveCachePrepopulate	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
HybridDriveCacheRebalance	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
IndexerAutomaticMaintenance	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
Installation	Completed Successfully	16/05/2023 17:15:45		Microsoft Corporation
Interactive	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
Launch Logon Script As Interactive User	Unknown	11/04/2023 12:38:08		
LocalUserSyncDataAvailable	Completed Successfully	05/01/2023 14:59:29		
Logon	Completed Successfully	16/05/2023 13:30:00		
Logon Synchronization	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation
LPRemove	Completed Successfully	15/05/2023 12:12:42		Microsoft Corporation
MaintenanceTasks	Completed Successfully	15/05/2023 12:12:42		$(@%SystemRoot%\system32\Windows.StateRepositoryClient.dll,-600)
MapsToastTask	Yet to Run	30/11/1999 00:00:30		Microsoft Corporation

Fig 3.1 – Unscheduled Tasks

USER DEFINED

USER DEFINED

Task Name	Result	Last Run Time	Next Run Time	Author
Demo	Completed Successfully	10/05/2023 14:50:20		US\saybal
MonitorPS	Currently Running	17/05/2023 12:30:00	17/05/2023 13:00:30	US\saybal
MonitorPSAuthor	Currently Running	17/05/2023 12:30:00	17/05/2023 13:00:30	US\saybal
User_Feed_Synchronization-{0A110474-3D1F-4BA3-AD4F-6EAB7E7178AF}	Completed Successfully	17/05/2023 11:56:26	17/05/2023 19:02:32	US\saybal

Fig 3.2 – User Defined Tasks

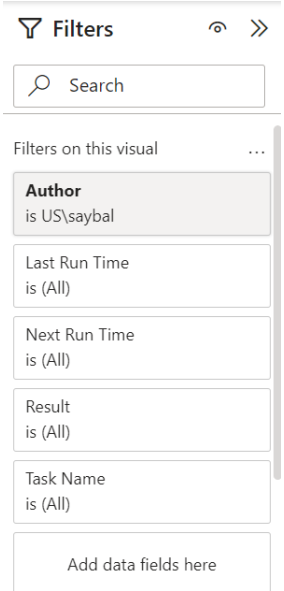


Fig 3.3 – Example of Filtering

LAST REFRESHED

While the Dashboard periodically updates based on the latest CSV File data every 30 minutes, it is also important to include a Last Updated field inside the view to make it even more clear.

A Last Refreshed Table is added at the bottom of the table and every time the report is refreshed, it updates with the latest time and date.

For this, we use the command DateTime.LocalNow.

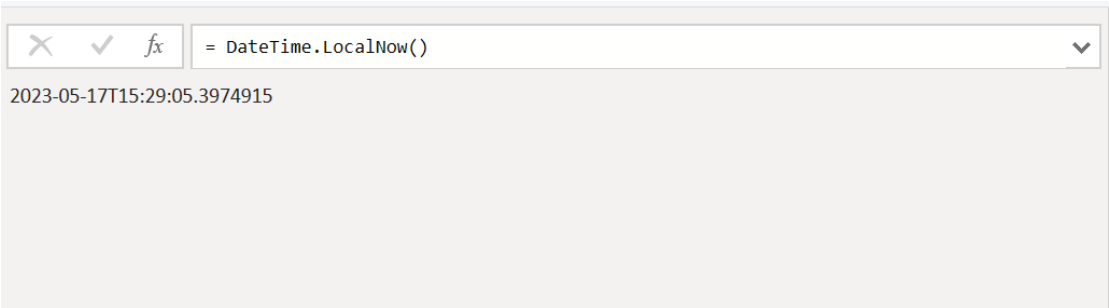


Fig 3.4 – DAX Command

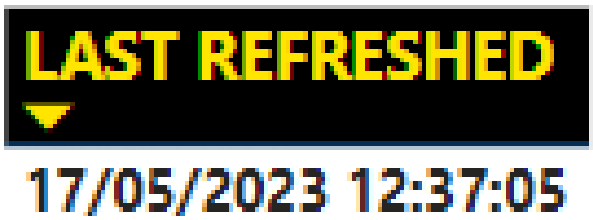


Fig 3.5 – Last Refreshed

TASK HISTORY

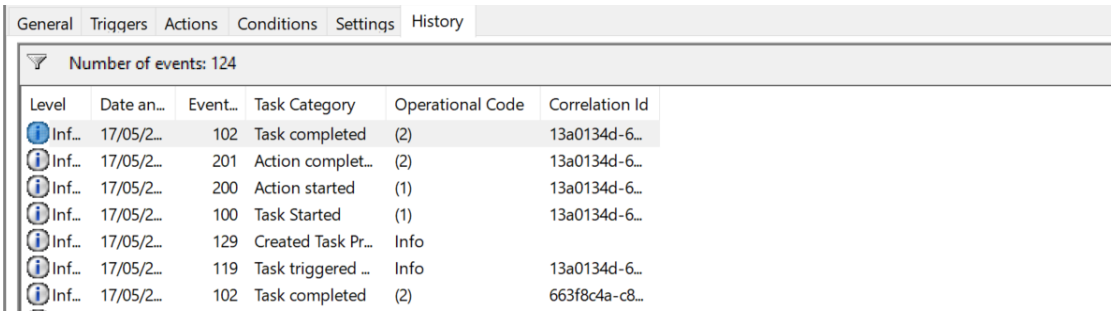
Now that we have the dashboard to monitor the status of all tasks along with all their properties, it is necessary to also gain access to their historical performances.

This is done so that a task can be analyzed in further detail and more information about its activity can be realized.

Historical data will give us insights into whether a task fails on a particular day, whether it takes too long to run, whether it is performing as expected etc.

So, with the running of a code, we can find out the past performance of a particular task.

Similarly, we can also use it to find out the last few activities that have been performed in the system.



Level	Date an...	Event...	Task Category	Operational Code	Correlation Id
Inf...	17/05/2...	102	Task completed	(2)	13a0134d-6...
Inf...	17/05/2...	201	Action complet...	(2)	13a0134d-6...
Inf...	17/05/2...	200	Action started	(1)	13a0134d-6...
Inf...	17/05/2...	100	Task Started	(1)	13a0134d-6...
Inf...	17/05/2...	129	Created Task Pr...	Info	
Inf...	17/05/2...	119	Task triggered ...	Info	13a0134d-6...
Inf...	17/05/2...	102	Task completed	(2)	663f8c4a-c8...

Fig 3.6 – Task History in PowerBI

CODE

For a particular Task Name, we add the name of the job in the code, and it retrieves the history for that task. We have the option of using the Stop -MaxEvents command to specify exactly how many run activities we want to see.

In this case, we are using the task name ‘MonitorPS,’ which is the task used to generate the CSV File from the executable.

The max no of events is 50.

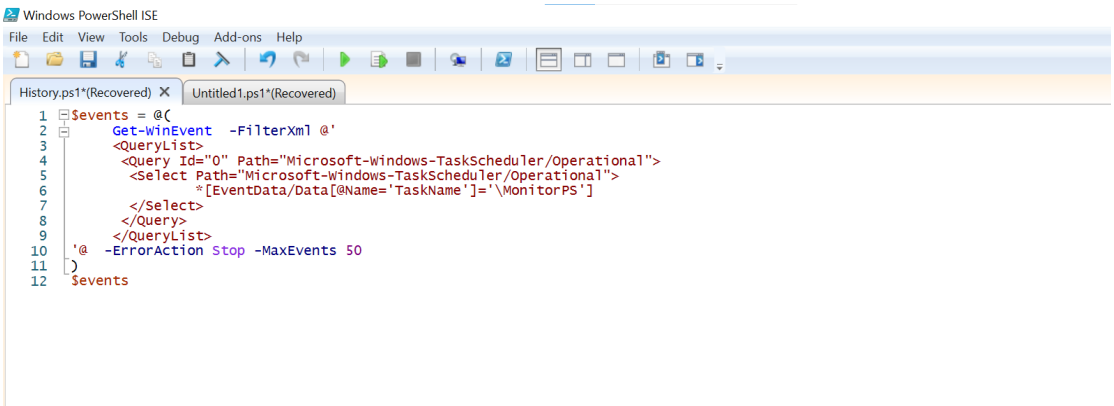
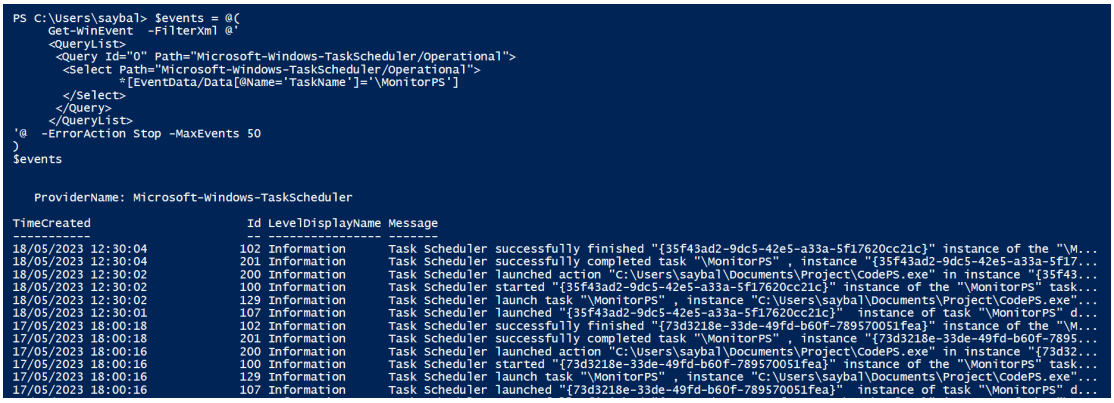


Fig 3.7 – Code to fetch history of a particular task



```
PS C:\Users\saybal> $events = @(
     Get-WinEvent  -FilterXml @'
     <QueryList>
      <Query Id="0" Path="Microsoft-Windows-TaskScheduler/Operational">
       <Select Path="Microsoft-Windows-TaskScheduler/Operational">
              *[EventData/Data[@Name='TaskName']='\MonitorPS']
       </Select>
      </Query>
     </QueryList>
'@  -ErrorAction Stop -MaxEvents 50
)
$events


   ProviderName: Microsoft-Windows-TaskScheduler

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
18/05/2023 12:30:04            102 Information      Task Scheduler successfully finished "{35f43ad2-9dc5-42e5-a33a-5f17620cc21c}" instance of the "\M...
18/05/2023 12:30:04            201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{35f43ad2-9dc5-42e5-a33a-5f17...
18/05/2023 12:30:02            200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{35f43...
18/05/2023 12:30:02            100 Information      Task Scheduler started "{35f43ad2-9dc5-42e5-a33a-5f17620cc21c}" instance of the "\MonitorPS" task...
18/05/2023 12:30:02            129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
18/05/2023 12:30:01            107 Information      Task Scheduler launched "{35f43ad2-9dc5-42e5-a33a-5f17620cc21c}"  instance of task "\MonitorPS" d...
17/05/2023 18:00:18            102 Information      Task Scheduler successfully finished "{73d3218e-33de-49fd-b60f-789570051fea}" instance of the "\M...
17/05/2023 18:00:18            201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{73d3218e-33de-49fd-b60f-7895...
17/05/2023 18:00:16            200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{73d32...
17/05/2023 18:00:16            100 Information      Task Scheduler started "{73d3218e-33de-49fd-b60f-789570051fea}" instance of the "\MonitorPS" task...
17/05/2023 18:00:16            129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
17/05/2023 18:00:16            107 Information      Task Scheduler launched "{73d3218e-33de-49fd-b60f-789570051fea}"  instance of task "\MonitorPS" d...
```

```
17/05/2023 15:30:03          102 Information      Task Scheduler successfully finished "{1525c9d8-8779-42f6-a580-ab686b34f828}" instance of the "\M...
17/05/2023 15:30:03          201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{1525c9d8-8779-42f6-a580-ab68...
17/05/2023 15:30:02          200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{1525c...
17/05/2023 15:30:02          100 Information      Task Scheduler started "{1525c9d8-8779-42f6-a580-ab686b34f828}" instance of the "\MonitorPS" task...
17/05/2023 15:30:02          129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
17/05/2023 15:30:02          107 Information      Task Scheduler launched "{1525c9d8-8779-42f6-a580-ab686b34f828}"  instance of task "\MonitorPS" d...
17/05/2023 15:00:03          102 Information      Task Scheduler successfully finished "{35b32275-d33a-45e3-8619-a8a54620808e}" instance of the "\M...
17/05/2023 15:00:03          201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{35b32275-d33a-45e3-8619-a8a5...
17/05/2023 15:00:02          200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{35b32...
17/05/2023 15:00:02          100 Information      Task Scheduler started "{35b32275-d33a-45e3-8619-a8a54620808e}" instance of the "\MonitorPS" task...
17/05/2023 15:00:02          129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
17/05/2023 15:00:02          107 Information      Task Scheduler launched "{35b32275-d33a-45e3-8619-a8a54620808e}"  instance of task "\MonitorPS" d...
17/05/2023 14:43:08          102 Information      Task Scheduler successfully finished "{e4c47bc2-9287-4616-b1a2-b45c5ecd7301}" instance of the "\M...
17/05/2023 14:43:08          201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{e4c47bc2-9287-4616-b1a2-b45c...
17/05/2023 14:43:03          200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{e4c47...
17/05/2023 14:43:03          100 Information      Task Scheduler started "{e4c47bc2-9287-4616-b1a2-b45c5ecd7301}" instance of the "\MonitorPS" task...
17/05/2023 14:43:03          129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
17/05/2023 14:43:03          107 Information      Task Scheduler launched "{e4c47bc2-9287-4616-b1a2-b45c5ecd7301}"  instance of task "\MonitorPS" d...
17/05/2023 13:00:05          102 Information      Task Scheduler successfully finished "{c190241e-fa6f-4c20-9999-43c8f8d19a55}" instance of the "\M...
17/05/2023 13:00:05          201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{c190241e-fa6f-4c20-9999-43c8...
17/05/2023 13:00:02          200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{c1902...
17/05/2023 13:00:02          100 Information      Task Scheduler started "{c190241e-fa6f-4c20-9999-43c8f8d19a55}" instance of the "\MonitorPS" task...
17/05/2023 13:00:02          129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
17/05/2023 13:00:02          107 Information      Task Scheduler launched "{c190241e-fa6f-4c20-9999-43c8f8d19a55}"  instance of task "\MonitorPS" d...
17/05/2023 12:30:05          102 Information      Task Scheduler successfully finished "{f4b24184-6efb-4852-a026-82ca6cbd6a00}" instance of the "\M...
17/05/2023 12:30:05          201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{f4b24184-6efb-4852-a026-82ca...
17/05/2023 12:30:02          200 Information      Task Scheduler launched action "C:\Users\saybal\Documents\Project\CodePS.exe" in instance "{f4b24...
17/05/2023 12:30:02          100 Information      Task Scheduler started "{f4b24184-6efb-4852-a026-82ca6cbd6a00}" instance of the "\MonitorPS" task...
17/05/2023 12:30:02          129 Information      Task Scheduler launch task "\MonitorPS" , instance "C:\Users\saybal\Documents\Project\CodePS.exe"...
17/05/2023 12:30:01          107 Information      Task Scheduler launched "{f4b24184-6efb-4852-a026-82ca6cbd6a00}"  instance of task "\MonitorPS" d...
16/05/2023 15:37:49          140 Information      User "saybal"  updated Task Scheduler task "\MonitorPS"
```

Fig 3.8 – Task History for ‘MonitorPS’

Now, if we want to fetch the history for all the tasks in the system, we can remove the name of the specific job from the code.

History.ps1*(Recovered) ✕ | Untitled1.ps1*(Recovered)

```
$events = @(
      Get-WinEvent  -FilterXml @'
      <QueryList>
       <Query Id="0" Path="Microsoft-Windows-TaskScheduler/Operational">
        <Select Path="Microsoft-Windows-TaskScheduler/Operational">
                *[EventData/Data[@Name='TaskName']]
        </Select>
       </Query>
      </QueryList>
'@  -ErrorAction Stop -MaxEvents 10
)
$events
```

```
   ProviderName: Microsoft-Windows-TaskScheduler

TimeCreated                     Id LevelDisplayName Message
-----------                     -- --------------- -------
18/05/2023 12:39:31            102 Information      Task Scheduler successfully finished "{03c78b2b-60f0-4c86-8d4e-09b9fb7114dd}" instance of the "\M...
18/05/2023 12:39:31            201 Information      Task Scheduler successfully completed task "\Microsoft\Office\OfficeTelemetryAgentFallBack2016" ,...
18/05/2023 12:39:29            200 Information      Task Scheduler launched action "C:\Program Files\Microsoft Office\root\Office16\msoia.exe" in ins...
18/05/2023 12:39:29            100 Information      Task Scheduler started "{03c78b2b-60f0-4c86-8d4e-09b9fb7114dd}" instance of the "\Microsoft\Offic...
18/05/2023 12:39:29            129 Information      Task Scheduler launch task "\Microsoft\Office\OfficeTelemetryAgentFallBack2016" , instance "C:\Pr...
18/05/2023 12:39:29            119 Information      Task Scheduler launched "{03c78b2b-60f0-4c86-8d4e-09b9fb7114dd}"  instance of task "\Microsoft\Of...
18/05/2023 12:36:00            140 Information      User "NT AUTHORITY\SYSTEM"  updated Task Scheduler task "\Microsoft\Windows\WindowsUpdate\Schedul...
18/05/2023 12:30:04            102 Information      Task Scheduler successfully finished "{35f43ad2-9dc5-42e5-a33a-5f17620cc21c}" instance of the "\M...
18/05/2023 12:30:04            201 Information      Task Scheduler successfully completed task "\MonitorPS" , instance "{35f43ad2-9dc5-42e5-a33a-5f17...
18/05/2023 12:30:04            102 Information      Task Scheduler successfully finished "{644c2c7c-4f6a-4963-8c51-da0c60487c84}" instance of the "\M...
```

Fig 3.9 – Task History for all Jobs

CONCLUSION

We have thus successfully implemented a dashboard in PowerBI which shows the status of all tasks in the system.

The CSV File continues to update itself every 30 minutes, and the dashboard is always up to date.

We have separate views for successful, failed and running tasks.

We can even view the history of each task by running the code, and it retrieves the data of as many run instances as we want.

The report can also be published online to make it easily accessible with the help of a hyperlink.

BIODATA

Name: Sayantan Bal

Phone No: +91 9051080559

Email: sayantan.bal2019@vitstudent.ac.in

Location: Kolkata