Overview

This Python script automates the process of **Tickets Pending** Daily Task.

Functionality

1. Reads ticket data from an OTRS generated Excel source file.
2. Uses a predefined Excel template to generate a new output excel file with the transformed data.
3. Detects and tags high-priority tickets by scanning ticket subjects for specific keywords.
4. Sorts tickets by priority and creation date.
5. Categorizes Tickets into Sheets and automatically assigns tickets to different sheets based on the “Queue” field.

V2 Script Update (Queue.txt file)

The **Queues.txt** file is used to categorize tickets into different sheets in the output Excel file. It defines mapping rules in the following format:

<Keyword>;<SheetName>

- The script scans the "**Queue**" column in each row in **generated excel file**.
- If the text in that column **contains** a keyword from the **first column of Queues.txt**, the entire row is copied to the sheet named in the second column.
- If no match is found, the row is excluded, left untagged and row won’t be copied to any other excel sheet.

Example of Queues.txt content:

```
FI;FI-CO
FI-CO;FI-CO
BPC
MM;MM_PP_QM
MM - WM - QM;MM_PP_QM
PO: Integrazioni
PP - PPDS;MM_PP_QM
SD;SD_CS
SAP BASIS;System
SUPP-SISTEMISTICO;System
ABAP;System
```

Explanation:

- If a ticket's queue contains FI or FI-CO, the row will go to the " FI-CO" sheet.
- If it contains SD, it goes to the " SD_CS" sheet.
- If no sheet name is specified (e.g., BPC), the script may skip the copy of the row
- This allows you to control how tickets are grouped in the final Excel file.

Output

New excel file “**Tickets Pending <current-date>.xlsx**“ that contains:

- Sheet “**All**” with all tickets formatted and sorted by **Priority** and **Date Created**.
- Categorized sheets (**SD_CS, MM_PP_QM, FI-CO, System**) with relevant tickets based on **Queue** column.

Method 1 – Python is installed directly to Windows (Error free)

Prerequisites

1. **Install Python for Windows**
   Download and install Python 3 from the official website: https://www.python.org/downloads/.
   During installation, make sure to check **“Add Python to PATH”**.
2. **Install Required Packages**
   Open **Command Prompt** and run the following command:

   ```
   pip install pandas openpyxl
   ```
3. **Prepare Script and Files**
   Ensure the following files are in the same folder as the script:
   - Latest OTRS Excel file (input)
   - Template.xlsx
   - Queues.txt
   - Customers.txt

Execution Instructions

1. Open **Command Prompt**, navigate to the location where the script is placed with the following command: **cd “PATH_TO_LOCATION”** - for example if script is placed to Desktop/ Tickets Pending Automation, the command to change to path should be:

   ```
   cd “Users/<username>/Desktop/Tickets Pending Automation”
   ```
2. Make sure that you download latest excel file from OTRS and place it to the same folder as the script.
3. Execute script with the following command:

python tp_automation_v2.py <generated_OTRS_file.xlsx>

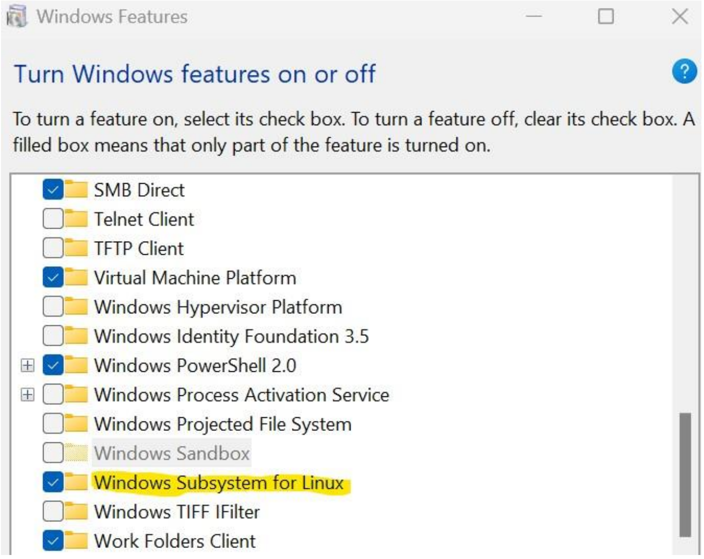
4. After executing script, a new file named “**Tickets Pending <current_date>.xlsx**” will be created.

Note: If you see any empty values in Customer or Queue column, alter the file Customers.txt or Queues.txt as there you can add all the customers and queues.

Method 2 – WSL (Contains errors | Not recommended)

WSL prerequisites (Old – Not recommended)

1. Make sure that **Windows Subsystem for Linux (WSL)** is enabled.



Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

☑ SMB Direct
□ Telnet Client
□ TFTP Client
☑ Virtual Machine Platform
□ Windows Hypervisor Platform
□ Windows Identity Foundation 3.5
⊞ ☑ Windows PowerShell 2.0
⊞ □ Windows Process Activation Service
□ Windows Projected File System
□ Windows Sandbox
☑ Windows Subsystem for Linux
□ Windows TIFF IFilter
☑ Work Folders Client

2. Files **Template.xlsx**, **Queues.txt** and **Customers.txt** must be in the same folder as the script.

WSL Installation (Old – Not recommended)

1. Ensure that **Windows Subsystem for Linux (WSL)** is enabled.
2. Install **Ubuntu** from **Microsoft Store**
3. In the **Ubuntu** execute the following commands:
    a) **sudo apt update -y**
    b) **sudo apt install python3-pandas python3-openpyxl -y**

WSL Execution Instructions (Old – Not recommended)

5. Launch Ubuntu instance.
6. Navigate to the location where the script is placed with the following command: **cd “/mnt/c/PATH_TO_LOCATION”** - for example if script is placed to Desktop/ Tickets

Pending Automation, the path should be: cd "/mnt/c/Users/<username>/Desktop/Tickets Pending Automation"

7. Make sure that you download latest excel file from OTRS and place it to the same folder as the script.
8. Execute script with the following command: **python3 tp_automation.py <generated_OTRS_file.xlsx>**

   Note: If you face any issues executing the script, try to execute as root user
   (sudo python3 tp_automation.py <generated_OTRS_file.xlsx>)
9. After executing script, a new file named "**Tickets Pending <current_date>.xlsx**" will be created. The file contains errors (working to fix it), select yes to let excel repair it and when it opens, save it again to have a non-error file.

   Note: If you see any empty values in Customer or Queue column, alter the file Customers.txt or Queues.txt as there you can add all the customers and queues.

Notes

Please let me know if you need further information and feel free to enhance the script and fix any issues that may arise.