Spendesk-NetSuite Python Automation Guidance

# Contents:

* [Installing](#_Installing:)
* [Running](#_Running:)
* [Updating](#_Updating:)
* [Expected Output](#_Expected_Output:)

The code and reference documents call all be found on GitHub at the following link: <https://github.com/jamescake2552/Spendesk_Automation>

# Installing:

1. Install Microsoft Visual Studio (VS) Code (from Microsoft store)
2. Install Python version 3.13 or higher (from Microsoft store)
3. Install Python extension in VS code

A screenshot of a computer menu

AI-generated content may be incorrect.

1. Open the code file in visual studio code.
2. Open the terminal in visual studio code (see picture below) and 1 by 1 paste the following instructions in, then press enter:
   1. pip install pandas
   2. pip install openpyxl

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Running:

1. **Open formatting.py in VS Code.**
2. **Run the script**Press the Run button in the top-right of VS Code. This will open a terminal window at the bottom of the screen.

A screenshot of a computer program

AI-generated content may be incorrect.

1. **The following options will appear, select option 1:**

* 1: Full process (Clean CSV → Enrich → Generate Summary)
* 2: Regenerate Summary only (from existing Excel)

1. **For option 1, input:**

* The full path to the raw Spendesk CSV file
* The desired path for the output Excel file (e.g. C:\Users\You\Documents\output.xlsx)
* The path to the reference Excel file containing staff and account mappings
* You can find these path locations the following way:

A screenshot of a computer

AI-generated content may be incorrect.

1. **Manually update expenses over £250**

Any expenses claiming over £250 will have the location left blank. Go into the spreadsheet and manually update these cells

1. **Rerun the script and select option 2.**

This will update the summary page with the newly entered locations from expense claims over £250

1. **For option 2, input:**

* The full path to the existing saved file
* The path to the reference Excel file containing staff and account mappings

1. **Make sure no Excel files are open during the run, especially the output or reference file.**
2. **Let it run — you'll see progress messages in the terminal. Processing typically takes under a minute.**

If you get an error, there is copilot in visual studio code, or paste the entirety of the code and the entirety of the error into ChatGPT, ask it why you are getting this error and how to correct it, 9/10 it will be able to solve this.

**Common errors:**



**Cause**: The referenced or output file is open in Excel.

**Fix**: Close the file and re-run the script.



**Cause**: The Spendesk CSV format changed or columns were renamed.

**Fix**: The Data sheet will still be produced. Check that it includes:

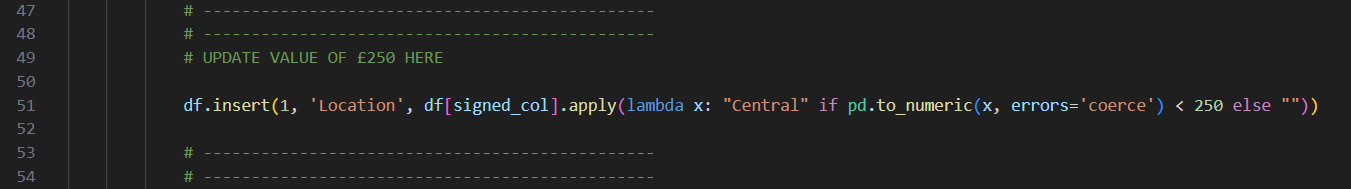
* Expense Account
* Net Amount
* Tax Amount
* Signed Total Amount

Update the script if column names differ. Refer to the update section on how to do this.

# Updating:

1. **Change the £250 Threshold**

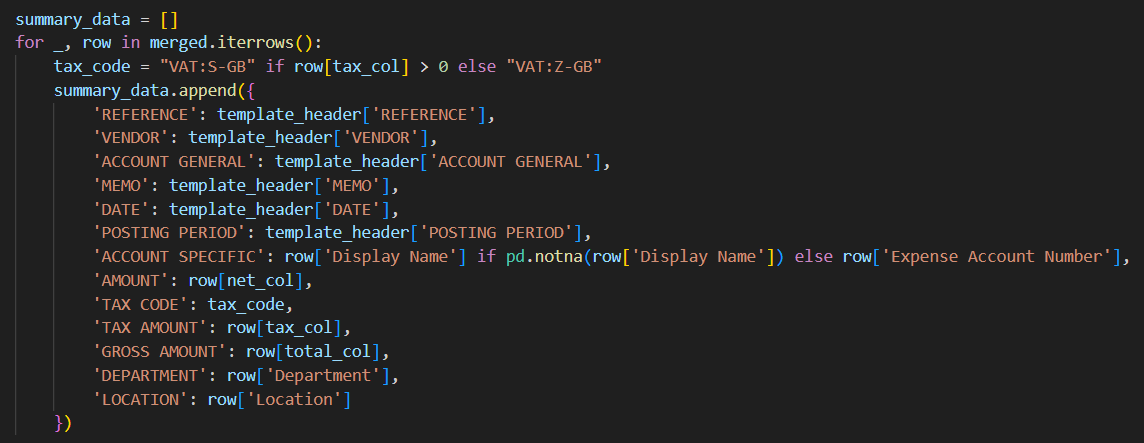
* Currently any amount under £250 is allocated to central location.
* This is defined on line 51 of ‘formatting.py’. If you would like to update this, change the value of 250 to your desired value.

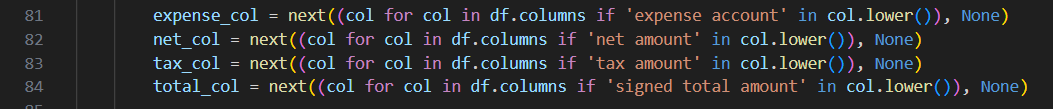


**2. Update the Reference File**

* File includes two sheets:
  + **Employee**: maps Spendesk names → NetSuite departments
  + **Account**: maps expense account numbers → display names
* Make sure all employees and accounts are included and spelled correctly, check the name used on Spendesk and ensure this matches what is written in the Spendesk Name column.
* The reference document can be found in the GitHub link at the front of this document

**3. Add or change fields in the summary sheet**

* If the name of a column in the spendesk CSV file changes, update these fields with the new name
* If you would like to update which columns are put into the summary spreadsheet, copy and past one of these lines, and the name with the name of your new column to include in the summary spreadsheet. Then add a row with the new summary column name to this section below. Run the code, if it doesn’t work, consult ChatGPT.



# Expected Output:

The script produces an Excel file with **two sheets**:

**Sheet 1: Data**

* Contains all expense entries
* Adds:
  + **Department** (matched from reference file)
  + **Location** (set to “Central” if under £250)
* Any entries over £250 will have a blank location (to be manually filled)

Example of Expected Data Sheet:

A screenshot of a computer

AI-generated content may be incorrect.

**Sheet 2: Summary**

* Summarised view grouped by:
  + Expense Account
  + Department
  + Location
* Includes:
  + **Net, Tax, Gross Amount**
  + **Tax Code** (based on whether tax amount is > 0)
  + **Posting period** auto generated from today’s date

Example of Expected Summary Sheet:

A screenshot of a computer

AI-generated content may be incorrect.