

Take Notepad++ to the next level with the Python Script Plugin

John Bielinski



Disclaimer

All opinions expressed in this presentation are my own and do not necessarily reflect those of Galderma or its employees.

I have no connection with any of the software or apps demonstrated here.

All scripts and programs are intended for demonstration purposes only and cannot be guaranteed to work correctly in other settings.

Introduction

Why Notepad++ ? Why Python?

Creating and running a script

Some demo scripts

Very brief overview - much more detail in paper

Why Notepad++? Why Python?



free, open-source text editor
easy to install and use
great for editing SAS files
loads of features + lots of
plugins



hugely popular
intuitive and easy to learn
vast built-in library of functions

Python Script for Notepad++

Install as Plugin - very quick and easy

Plugins>Plugins Admin>Available>Python Script>Install

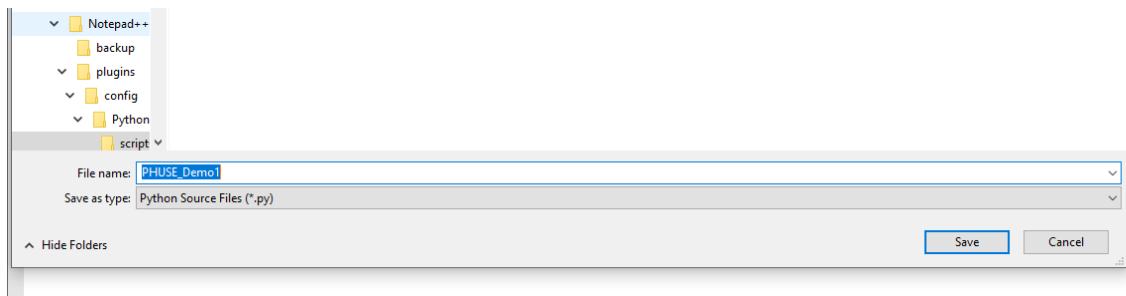
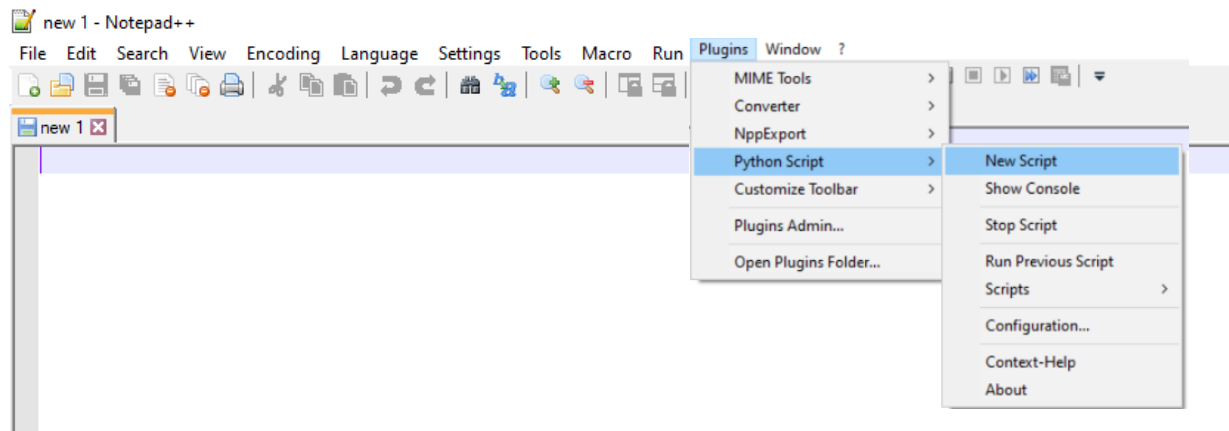
<http://npppythonscript.sourceforge.net/>

Fully-integrated within Notepad++

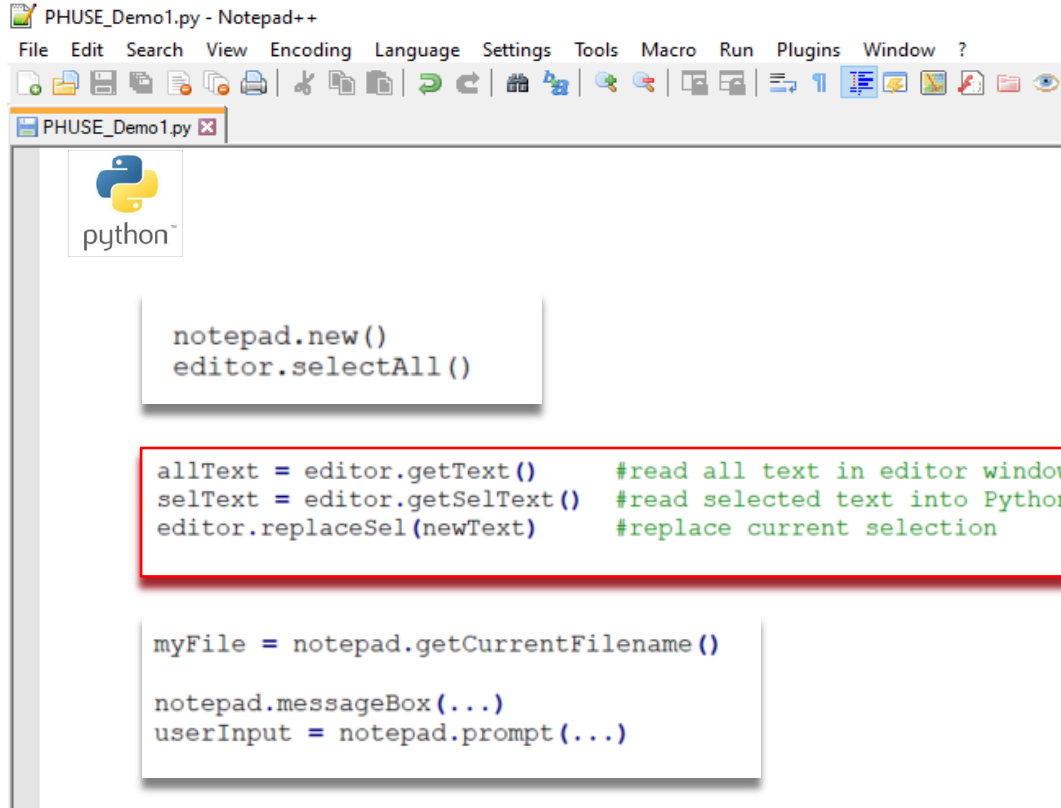
Complete control of all aspects of Notepad++

Run scripts with click of mouse

Creating a script



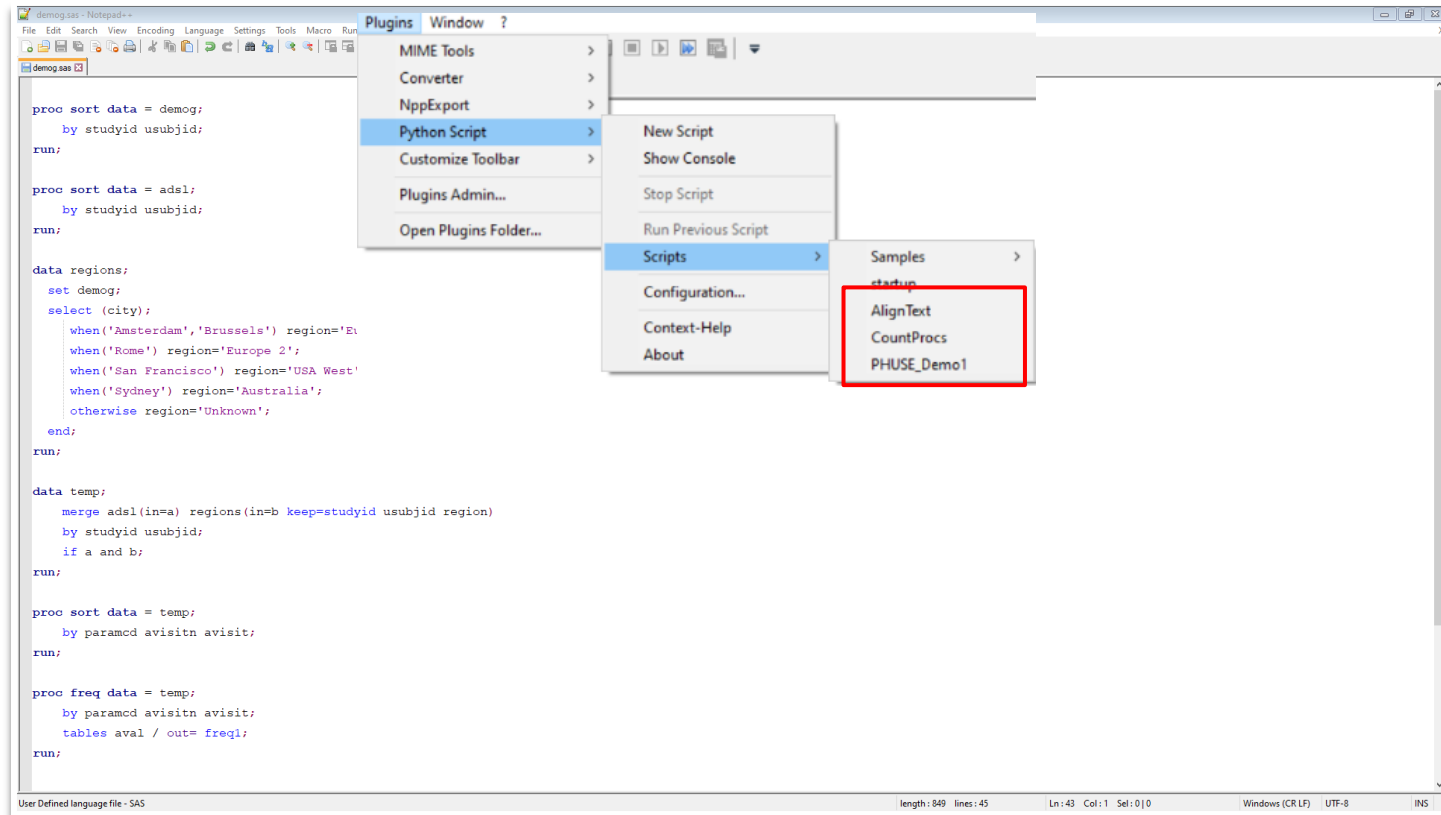
Creating a script



notepad object
editor object

- run menu commands
- get information --> Python variables

Running a script



Example 1: Script to align text

The image shows a Notepad++ window with two SAS scripts. The left window, titled 'demog.sas - Notepad++', contains the following code:

```
proc sort data = demog;
  by studyid usubjid;
run;

proc sort data = adsl;
  by studyid usubjid;
run;

data regions;
  set demog;
  select (city);
    when('Amsterdam','Brussels') region='Europe 1';
    when('Rome') region='Europe 2';
    when('San Francisco') region='USA West';
    when('Sydney') region='Australia';
    otherwise region='Unknown';
  end;
run;

data temp;
  merge adsl(in=a) regions(in=b keep=studvid usubjid region)
```

An 'AlignText' dialog box is open, showing the text 'region' in the input field. The dialog has a title bar 'AlignText' and a close button 'X'. The main text area contains 'region'. Below the text area are 'OK' and 'Cancel' buttons. The text 'Enter the text to align on (CASE-SENSITIVE)' is visible above the input field.

The right window, titled '*demog.sas - Notepad++', shows the same code after alignment:

```
set demog;
select (city);
  when('Amsterdam','Brussels') region='Europe 1';
  when('Rome') region='Europe 2';
  when('San Francisco') region='USA West';
  when('Sydney') region='Australia';
  otherwise region='Unknown';
end;
run;

data temp;
  merge adsl(in=a) regions(in=b keep=studvid usubjid rec
```

The 'Plugins' menu is open, showing options like 'MIME Tools', 'Converter', 'NppExport', 'Python Script', 'Customize Toolbar', 'Plugins Admin...', and 'Open Plugins Folder...'. The 'Scripts' submenu is also open, showing options like 'New Script', 'Show Console', 'Stop Script', 'Run Previous Script', 'Scripts', 'Configuration...', 'Context-Help', and 'About'. The 'Scripts' submenu is further open, showing options like 'Samples', 'startup', 'AlignText', 'CountProcs', and 'PHUSE_Demo1'.



AlignText.py

def alignText_main():

```
oldString = editor.getSelText()
```

```
splitchar = notepad.prompt('Enter the text to align on (CASE-SENSITIVE)', 'AlignText', ' ')
```

```
if splitchar==None or len(splitchar)<=0 or splitchar not in oldString:
    return None
```

```
oldList = oldString.split('\n')
splitpos = max([line.find(splitchar) for line in oldList])
```

```
splitList = [line.rstrip().partition(splitchar) for line in oldList]
newList = []
for line in splitList:
    newLine = '{0:{width}}{1}{2}'.format(line[0],line[1],line[2],width=splitpos)
    newList.append(newLine)
```

```
newString = '\n'.join(newList)
```

```
editor.replaceSel(newString)
```

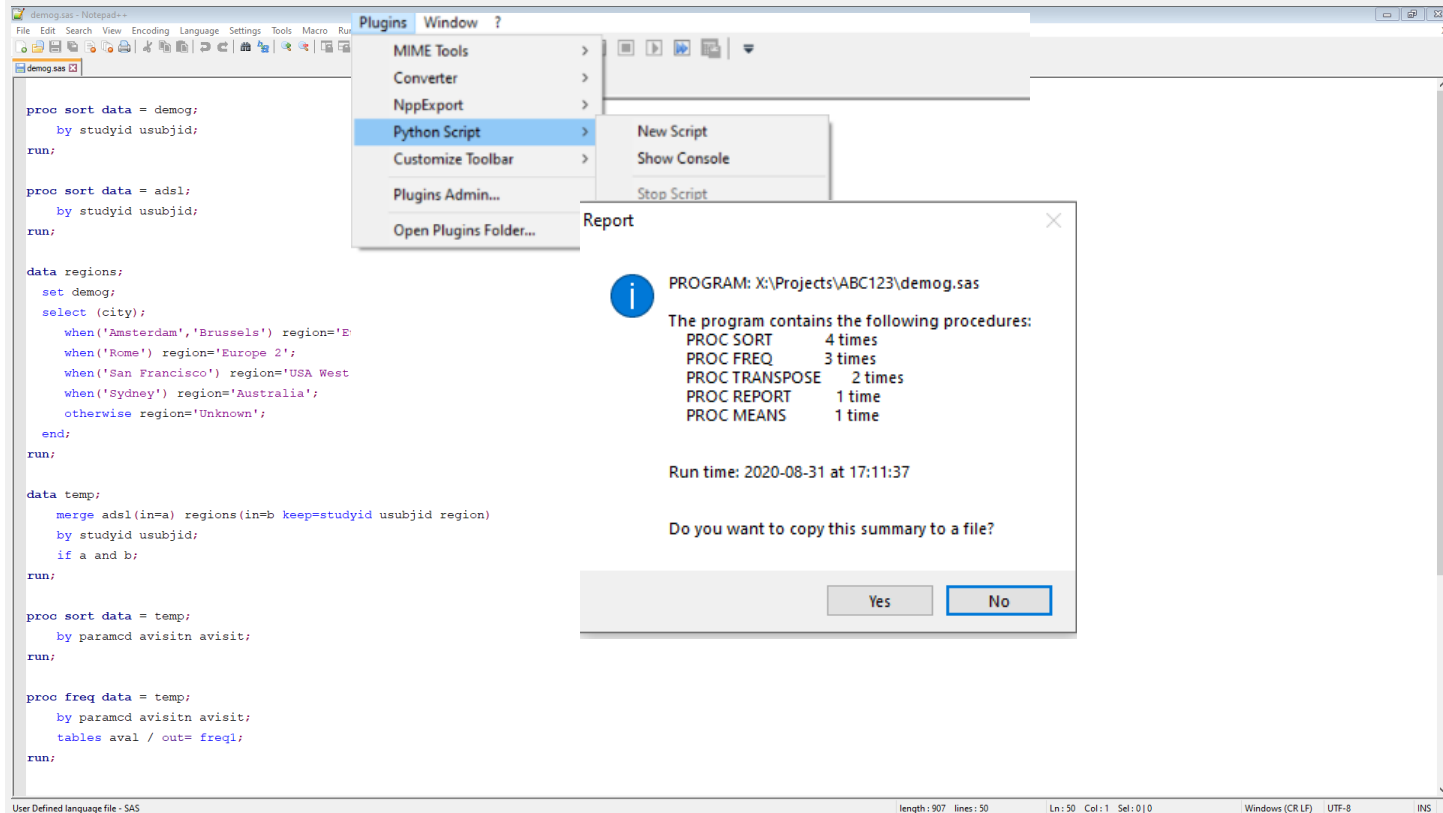
alignText_main()

```
data regions;
set demog;
select (city);
  when('Amsterdam','Brussels') region='Europe 1';
  when('Rome') region='Europe 2';
  when('San Francisco') region='USA West';
  when('Sydney') region='Australia';
  otherwise region='Unknown';
end;
run;
```



```
data regions;
set demog;
select (city);
  when('Amsterdam','Brussels') region='Europe 1';
  when('Rome') region='Europe 2';
  when('San Francisco') region='USA West';
  when('Sydney') region='Australia';
  otherwise region='Unknown';
end;
run;
```

Example 2: Count the SAS procs in a program



The screenshot shows the SAS Studio interface. On the left, a Notepad window displays a SAS program. The program includes several PROC statements: PROC SORT, PROC FREQ, PROC TRANSPOSE, PROC REPORT, and PROC MEANS. A 'Report' dialog box is open in the center, displaying the results of a PROC REPORT statement. The dialog box shows the program name, the list of procedures and their frequencies, the run time, and a prompt to copy the summary to a file.

```
proc sort data = demog;
  by studyid usubjid;
run;

proc sort data = adsl;
  by studyid usubjid;
run;

data regions;
  set demog;
  select (city);
    when('Amsterdam','Brussels') region='E';
    when('Rome') region='Europe 2';
    when('San Francisco') region='USA West';
    when('Sydney') region='Australia';
    otherwise region='Unknown';
  end;
run;

data temp;
  merge adsl(in=a) regions(in=b keep=studyid usubjid region)
  by studyid usubjid;
  if a and b;
run;

proc sort data = temp;
  by paramcd avisitn avisit;
run;

proc freq data = temp;
  by paramcd avisitn avisit;
  tables aval / out= freq1;
run;
```

Report

PROGRAM: X:\Projects\ABC123\demog.sas

The program contains the following procedures:

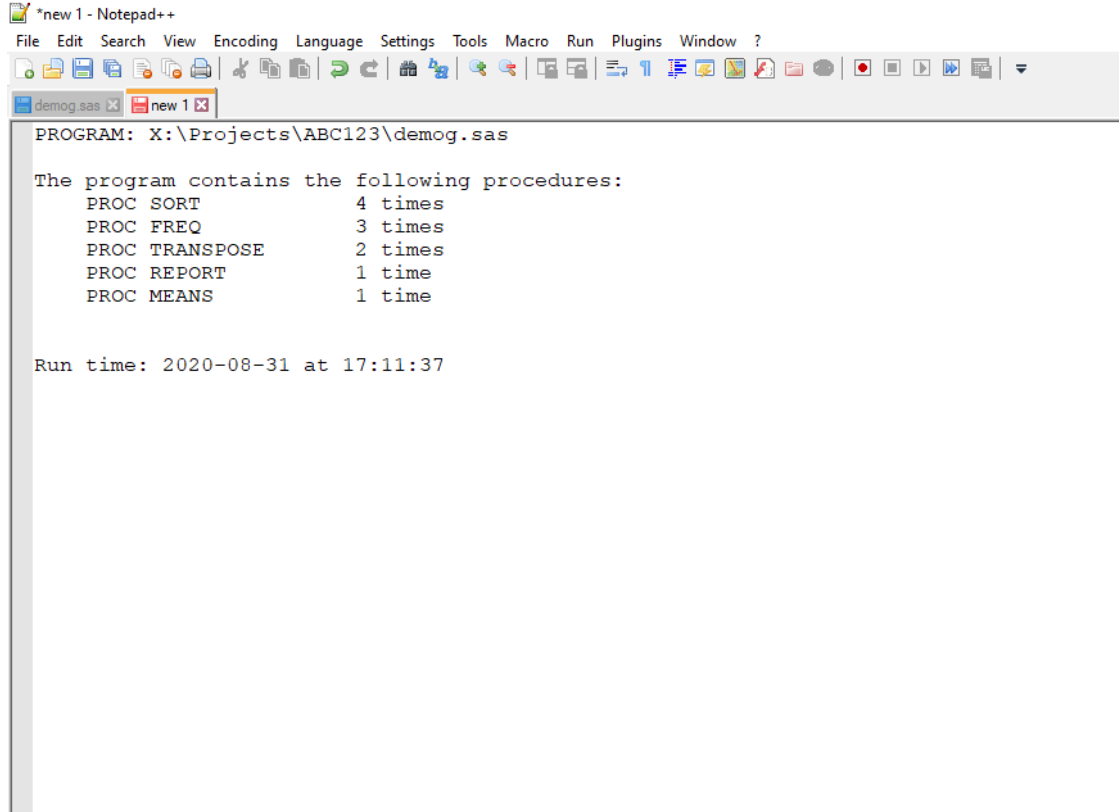
PROC SORT	4 times
PROC FREQ	3 times
PROC TRANSPOSE	2 times
PROC REPORT	1 time
PROC MEANS	1 time

Run time: 2020-08-31 at 17:11:37

Do you want to copy this summary to a file?

Yes No

Example 2: Count the SAS procs in a program



The screenshot shows a Notepad++ window titled '*new 1 - Notepad++'. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations and editing. The active window is 'demog.sas', and the text content is as follows:

```
PROGRAM: X:\Projects\ABC123\demog.sas

The program contains the following procedures:
  PROC SORT          4 times
  PROC FREQ          3 times
  PROC TRANSPOSE     2 times
  PROC REPORT        1 time
  PROC MEANS          1 time

Run time: 2020-08-31 at 17:11:37
```

```
countProcs.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
countProcs.py
import re
import datetime

def countProcs_main():

    progtext = editor.getText()

    progname = notepad.getCurrentFilename()

    procRegex = re.compile(r'''
        ^          # Start of file or new line
        \s*        # 0 or more whitespaces
        \b         # Word boundary
        PROC       # The word 'PROC'
        \s+        # 1 or more whitespaces
        (\w+)      # Capturing group: 1 or more word characters
    ''', re.IGNORECASE|re.VERBOSE|re.MULTILINE)

    procsFound = procRegex.findall(progtext)
    procCount = dict()
    for p in [i.strip().upper() for i in procsFound]:
        if p in procCount.keys():
            procCount[p]+=1
        else:
            procCount[p]=1

    summaryText = 'PROGRAM: ' + progname + '\n\n'
    summaryText += 'The program contains the following procedures:\n'

    for k,v in sorted(procCount.items(),key=lambda item: item[1],reverse=True):
        summaryText += '    PROC {0:12} {1:4} time{2}\n'.format(k,v,'s'*(v>1))

    summaryText += '\n\nRun time: {}'.format(datetime.datetime.now().strftime("%Y-%m-%d at %H:%M:%S"))

    msgBoxText = summaryText + ('\n'*3 + 'Do you want to copy this summary to a file?')

    userInput = notepad.messageBox(msgBoxText,'Report'
        ,MESSAGEBOXFLAGS.ICONINFORMATION|MESSAGEBOXFLAGS.YESNO|MESSAGEBOXFLAGS.DEFBUTTON2)

    if userInput == MESSAGEBOXFLAGS.RESULTYES:
        notepad.new()
        editor.addText(summaryText)

countProcs_main()
```

Developing scripts - some tips

- Is there already a plug-in? A Python module?

e.g. 'Code Alignment', 'TextFX'

- Experiment...in safe setting
any command can be automated
(e.g. open file, replace text, save file...)



- Don't automate everything !!
keep some degree of user interaction

Conclusion

- Very powerful set of tools for automating tasks
- Easy to set up and integrate into workflow
- Just a brief overview - much more detail in paper