

```

import easygui as eg
import sys, re
from os import path

def isnum(txt):
    if not txt: return True
    try:
        float(txt)
        return True
    except ValueError:
        return False

def isdate(txt):
    # Date format MM[/]DD[/][YY]YY
    dateformat = r'^[0123]?\\d[/]?[0123]?\\d[/]?[\\d]{2,4}$'
    if not txt: return True
    return re.match(dateformat,txt)!=None

def get_filename_gui(read=True):
    #Open SDC File
    filetypes = [ "*.txt", "Fixed Width"] if read else [ "*.tsv",
    "Tab Separated"]
    msg="Open SDC Data File" if read else "Choose Save Location"
    egfunc = eg.fileopenbox if read else eg.filesavebox
    filename = egfunc(msg=msg,
                      title="SDC Data File",
                      filetypes=filetypes)
    if filename == None:
        sys.exit(0)
    elif read and not path.exists(filename):
        sys.exit(0)
    return filename

def get_firstline_gui(file,filehead):
    tmplist = []
    for i in range(15):
        tmp = file.readline()
        tmplist.append("Line %02d:      %s"%(i,tmp))
        filehead.append(tmp)
    file.seek(0)
    line = eg.choicebox(msg="Select the first line of
data",choices=tmplist)
    if line:

```

```

        try:
            return int(re.search(r'Line (\d+):',line).group(1))
        except:
            print "Integer cast error: ",sys.exc_info()[0]
            return -1
    return -1

def get_header(filehead):
    for i in range(len(filehead)-1,-1,-1):
        if len(filehead[i]) <= 1:
            del filehead[i]
    #print filehead
    line1 = filehead[0]
    headers = re.split(r'[ ]{2,}','',line1)
    loc = 0
    colstart = []
    for head in headers:
        thisline = line1.find(head,loc)
        if thisline < 0:
            print "Can't find '%s' in '%s'"%(head,line1[loc:])
            return colstart,[]
        colstart.append(max(thisline-1,0))
        loc = thisline+len(head)
    for line in filehead[1:]:
        for i in range(len(colstart)):
            try:
                head_txt = line[colstart[i]:colstart[i+1]]
                headers[i] = headers[i].strip() + " "
+head_txt.strip()
            except IndexError:
                break
    return colstart,[x.strip() for x in headers if x.strip()]

def get_vars_gui(headers_raw):
    num = len(headers_raw)
    headers_short = ["var%d"%i for i in range(num)]
    headers_ret = []
    if num < 16:
        return eg.multenterbox(msg='Enter variable names.',
title='Variable Names',
                                fields=headers_raw, values=headers_short)
    for i in range(0,num,10):
        iplus1 = min(i+10,num)

```

```

        print i,iplus1
        tmp = eg.multenterbox(msg='Enter variable names. (%1d of
%1d)'%(i/10+1,num/10+1),
                                title='Variable Names',
                                fields=headers_raw[i:iplus1],
values=headers_short[i:iplus1])
        if not tmp:
            headers_ret += headers_short[i:iplus1]
        else:
            headers_ret += tmp
    return headers_ret

def clean_numbers(matchobj):
    return re.sub(r'["',,',''',matchobj.group(0))

def get_row(colnums,rowstring,delim='',cint=[],cdate=[],clen=[]):
    retlist = []
    num = len(colnums)-1
    for i in range(num):
        val = rowstring[colnums[i]:colnums[i+1]].strip()
        retlist.append(re.sub(r'^[\d,.]+$ ',clean_numbers,val))
        if cint and len(cint) >= num and cint[i]:
            cint[i] &= isnum(retlist[-1])
        elif cdate and len(cdate) >= num and cdate[i]:
            cdate[i] &= isdate(retlist[-1])
        if clen and len(clen) >= num and clen[i] < len(retlist[-1]):
            clen[i] = len(retlist[-1])
        if i == 1:
            print "longest: ",retlist[-1]
    return retlist if not delim else delim.join(retlist)

def
make_sas_import(var_names,var_desc,clean_filename,cint,cdate,clen):
    sas_content = "DATA SDC_IMPORT;\n        %let _EFIERR_ = 0;\n
infile "
    sas_content += "'%s' delimiter='09'x MISSOVER DSD lrecl=32767
firstobs=2;\n"%clean_filename
    for i in range(len(var_names)):
        if cint[i]:
            sas_content += "        informat %s best32.;\n"%var_names[i]
        elif cdate[i]:
            sas_content += "        informat %s
mmddyy10.;\n"%var_names[i]

```

```

        else:
            sas_content += "        informat %s
%d.;\n"%(var_names[i], clen[i])
        for i in range(len(var_names)):
            if cint[i]:
                sas_content += "        format %s best32.;\n"%var_names[i]
            elif cdate[i]:
                sas_content += "        format %s mmddyy10.;\n"%var_names[i]
            else:
                sas_content += "        format %s
%d.;\n"%(var_names[i], clen[i])
        sas_content += "        input\n"
        for i in range(len(var_names)):
            if cint[i] or cdate[i]:
                sas_content += "        %s\n"%var_names[i]
            else:
                sas_content += "        %s $\n"%var_names[i]
        sas_content += "        ;\n        if _ERROR_ then call
symputex('_EFIERR_',1);\nrun;"
        print sas_content
        return sas_content

def run_main():
    delim="\t"
    infilename = get_filename_gui()

    try:
        infile = open(infilename, 'r')
    except:
        print "File open error: ", sys.exc_info()[0]
        sys.exit(1)

    filehead=[]
    firstline = get_firstline_gui(infile, filehead)

    if firstline < 0:
        print "Firstline was %d"%firstline
        sys.exit(1)

    if firstline > 0:
        (splits, headers_full)=get_header(filehead[:firstline-1])
        #print headers_full
        var_names = get_vars_gui(headers_full)

```

```

#print var_names

outfilename = get_filename_gui(False)
try:
    outfile = open(outfilename, 'w')
    outfile.write(delim.join(var_names)+"\n")
except:
    eg.msgbox("Problem writing to file. Sorry :-(", title="Oh
Noes!")
    infile.close()
    print sys.exc_info()[0]
    sys.exit(1)

#clear out headers. Throw em away.
for i in range(firstline):
    infile.readline()

col_is_int = [True for i in range(len(var_names))]
col_is_date = [True for i in range(len(var_names))]
col_max_len = [0 for i in range(len(var_names))]

for line in infile:
    row =
get_row(splits, line, delim=delim, cint=col_is_int, cdate=col_is_date, cle
n=col_max_len)
    outfile.write(row+"\n")

for i in range(len(col_is_date)):
    col_is_date[i] &= not col_is_int[i]

if eg.ynbox("Do you want to create a SAS import file?"):
    sascontent =
make_sas_import(var_names, headers_full, outfile,

col_is_int, col_is_date, col_max_len)
    sasfilename = get_filename_gui(False)
    sasfile = open(sasfilename, 'w')
    sasfile.write(sascontent)
    sasfile.close()

infile.close()
outfile.close()
sys.exit(0)

```

```
if __name__ == "__main__":  
    run_main()
```