FINDING CODE WITH KEY WORDS IN UNIX FOLDERS

James Willis, BA, Base & Advanced SAS® Certified
Senior Data Analyst
Research for Aging Populations, Optum

ABSTRACT

When code is stored in UNIX, the folder structure is similar to a Windows structure; however, UNIX does not have the “Search” feature that Windows has. A programmer has to know where the code is located or use the UNIX “grep command” to search through files in a folder. Using the "Find_a_variable" utility, words in UNIX files can be found, listed and sent to a txt file. SAS® code can then read the .txt file, extract pertinent information, write the extract to a spreadsheet and email the spreadsheet. This utility will solve a question that was asked about “grep” commands on the SAS COMMUNITIES website. The utility also defines a macro that uses the SAS “x” statements to issue UNIX commands that change the directory, delete text files if they exist, insert “grep” values inside the files, and send the results to a text file. SAS code will then scan the text file for keywords, separate the file parts into variables, and create a SAS dataset. The SAS dataset can then be printed to a SAS Enterprise Guide window or exported to a spreadsheet and emailed. Thus, this macro provides an easy way to find words in UNIX files, and by using SAS as an intermediate step, export them to a spreadsheet as desired.

INTRODUCTION

The task assigned was to update SAS code that runs on a UNIX platform, designed and written by one programmer, and updated by three additional programmers, none of whom were available to answer questions. The task required knowing where macros were called from, where parm values were written to, and where macro values were used. The UNIX ‘grep’,’egrep’, and ‘fgrep’ commands list files that contain a pattern and write readable results. For example: grep -i ‘jwillis’ */my/folder/path/*code/*.sas. The results are visible in the window where the command was submitted; however, the result is not useful when there are a multitude of .sas files containing ‘jwillis’.

A way to create a file that contained the ‘grep’ results was needed. The next step was to direct the results away from the submission window into a text file. That way, a retrievable list of code containing the searched word would be available. The solution was to add a right-angle bracket, “>”, as a redirection operator. grep -i ‘jwillis’ */my/folder/path/*code/*.sas > */my/folder/path/*code/grep_results.txt. This allows the user to open the text file at any time and determine which SAS files contained the searched word.

Next, a way was needed to look for values in code without having to leave an Enterprise Guide (E.G.) session. Using E.G. avoids constantly typing out the UNIX ‘grep’ command. UNIX commands can be submitted by a SAS job such that that job can read and apply logic to text files and can send emails with files attached. Knowing these SAS capabilities, a utility was created to look for values in any file in any UNIX location.

METHODS

The first step is to create a macro that will perform the ‘grep’ process from a SAS program. This is a macro named test2 with hardcoded locations and values.

```
%macro test2;
x " cd /my/folder/path/code " ;
x " rm /my/folder/path/code/grep_results.txt";
x " grep -i ‘jwillis’ /my/folder/path/code/*.sas > /my/folder/path/code/grep_results.txt ";
%mend test2;
%macro test2;
```

The second step is to make macro variables that can be changed to suit the user’s needs. This step allows hardcoding the values in the macro that do not change and making changes to the values in the macro that vary.

```
%global basepath name suff loc;
%let basepath = /My/master/working/location;          /*** Where are you working from? ***/
%let name     = %nrstr(%"jwillis%");                   /*** What value are you looking for? ***/
%let suff     = sas;                                    /*** Which file suffix do you look in? ***/
%let loc      = code;                                 /*** Which folder do you want to look in? ***/
%let src      = /The/base/location/to/lookin/&loc.;
```

This is shows how the macro named test2 is rewritten to use macro variable values.

```
%macro test2;
x " cd &src. " ;                                          /*** go to the location to search in ***/
x " rm &basepath./var_name_&loc._list.txt";                  /***delete the .txt file if it exists. ***/
x " grep -i &name. *.&suff. > &basepath./var_name_&loc._list.txt";
%mend test2;
%test2;
```

The third step is to use SAS to read the text file created by the macro test2.

```
/*** read in the text file just created and select out the pertinent information ***/
proc sql; drop table work.varnametxt; quit;
 data work.varnametxt;
    infile "&basepath./var_name_&loc._list.txt"
    dlm='09'x dsd
    truncover firstobs=1 end=eof;
    FORMAT FULL_NAME $200.;
    Input  FULL_NAME $200.;
/**** add any additional special coding here ****/
    if index(FULL_NAME,".&suff") > 0 then output;
run;

/*** This prints the varnametxt sas file to the SAS session results window ***/
  title "looking for: &name.   ";
  title2 "in folder:  &src.    ";
proc print data=work.varnametxt(obs=max);
run;
  title "  ";
```

The final step is to use SAS to convert the SAS file into an Excel file and email it to a larger audience. This step is also helpful for sending the results to multiple persons, or to run multiple searches in succession when there is not enough time to analyze the results one search at a time.

```
/*** convert the &name macro value to a value that can be used for different purposes ***/
data null;
  namit = trim(left(compress(translate(trim(left("&name.")),",""))));
  call symput('namit',namit);
run;
%put &=namit;

proc export
    data= work.varnametxt
    outfile = "&basepath./find_%SYSFUNC(compress(&namit.))_in_PROD_&loc..xlsx"
    replace
    DBMS = xlsx;
    sheet=&loc._&namit.;
run;

%LET from_email = 'my.email@address.com';
%LET to_email = 'my.email@address.com';

FILENAME outmail EMAIL
    SUBJECT=" &name. &loc. &src. &namit."
    ATTACH=("&basepath./find_%SYSFUNC(compress(&namit.))_in_PROD_&loc..xlsx"
         content_type = 'application/excel')
    TO = (&to_email.)
    ;

DATA _NULL_;
    FILE outmail;

    Put "Attached you will find the ‘find a key word report’ spreadsheet ";
    Put " ";
    Put " Looking for: &name. in folder:  &src. for &loc.  ";
    Put " ";
    Put " ";
    Put 77*"=";
    Put "This is a system generated e-mail. Email replies to &from_email.";
    ;
RUN;
```

CONCLUSION

When SAS code is stored in UNIX, a programmer learns that UNIX does not have the "Search" features that Windows has. A programmer has to know exactly what is written in the UNIX files or use the UNIX "grep command" to search through files in a folder. Using the "Find_a_variable" utility, words in UNIX files can be found, listed and sent to a txt file. SAS code can then read the .txt file, extract pertinent information, write the extract to a spreadsheet and email the spreadsheet.

REFERENCES:

Robbins, Arnold. October 2005. "UNIX IN A NUTSHELL', 4TH Edition, Sebastopol, CA: O'Reilly.

CONTACT INFORMATION:

James Willis
Phone: 763-283-3654
Email: james.willis@optum.com