phuse·connect
connect·share·advance
March 8th–11th
Renaissance Orlando at SeaWorld Florida
US 2020
The Clinical Data Science Conference

Paper SP12

A dynamic-automated approach to reviewing RTF files

Youying Yu, Eli Lilly and Company, Indianapolis, Indiana, USA
Hrideep Antony, Syneos Health, Morrisville, North Carolina, USA

ABSTRACT

This paper introduces “reviewfiles” utility which automates the review process of multiple RTF files. Combined features of SAS® language, web scripting languages, and SAS Enterprise Guide® (SAS EG) project functionality are used to build this utility. The current version offers a verity of file review capabilities such as summarizing the differences in the RTF files based on any changes to the contents, providing a refined version of changes to the contents based on a threshold value (eg. Search for changes to p-values > .5, etc.), creating a search result on all the files in the folder based on a key-value (eg. Search for file names and summary statistics relevant to a specific preferred term “Dizziness” etc.), Extracting the first title of the RTF files and providing hyperlinks to the files for easy single-click access to the files. This utility not only makes the review of files easier but will also act as a means to control the quality of the overall deliverables. The utility can be accessed through a web interface or by using the SAS Enterprise Guide project functionality. This paper will focus mainly on the SAS EG interface.

INTRODUCTION

This utility consists of three key components as shown in flowchart figure 1 and they are:

- The HTML and SAS Enterprise Guide interface which acts as an interface between the user and communicates with the SAS macro
- The %reviewfiles SAS utility macros that read in the RTF files and processes the data.
- The XML output files, which provides the summary of findings along with the corresponding titles and file names for quick access

The functionality of this utility will be further discussed in this paper.

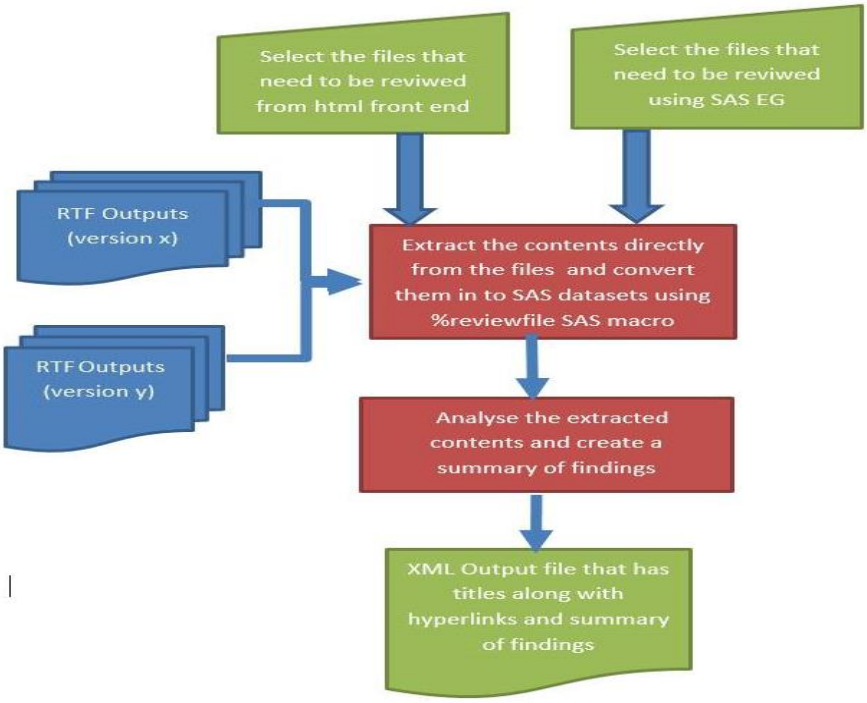

Figure1: Process flow chart

PROJECT INTERFACE

SAS Enterprise Guide is a point-and-click, menu- and wizard-driven tool that empowers users to analyze data and publish their results. As indicated in Figure 2 interface design flow chart below, either SAS EG or the Web interface is used to pass the input parameters to the utility macro. Submit key on the webpage runs the script or the “RUN” button on the SAS EG screen compiles and runs the %reviewfiles SAS macro. The %reviewfiles macro reads in the RTF files and processes the data, generating a summary of findings which are then displayed on the web interface while also allowing the XML version of the output to be saved in the location specified by the user.

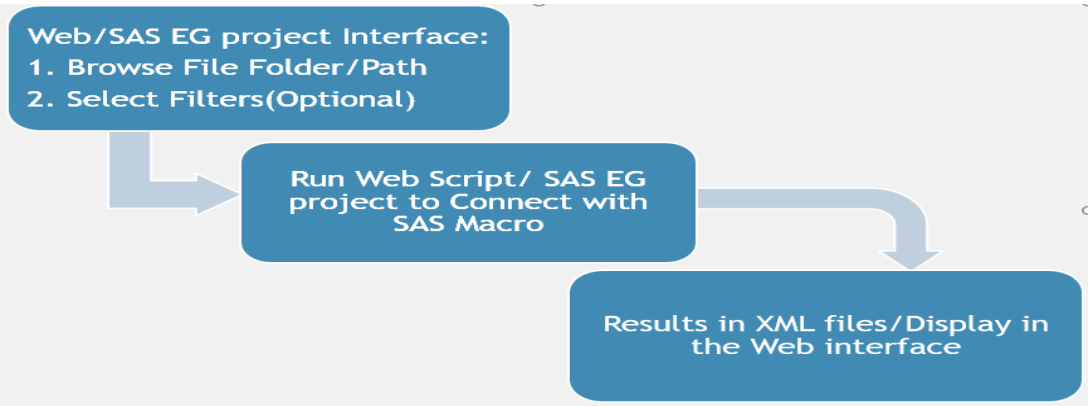

Figure 2 interface design flow chart

A screenshot of the SAS EG Project interface that allows the user to pass parameters is shown in figure 3.

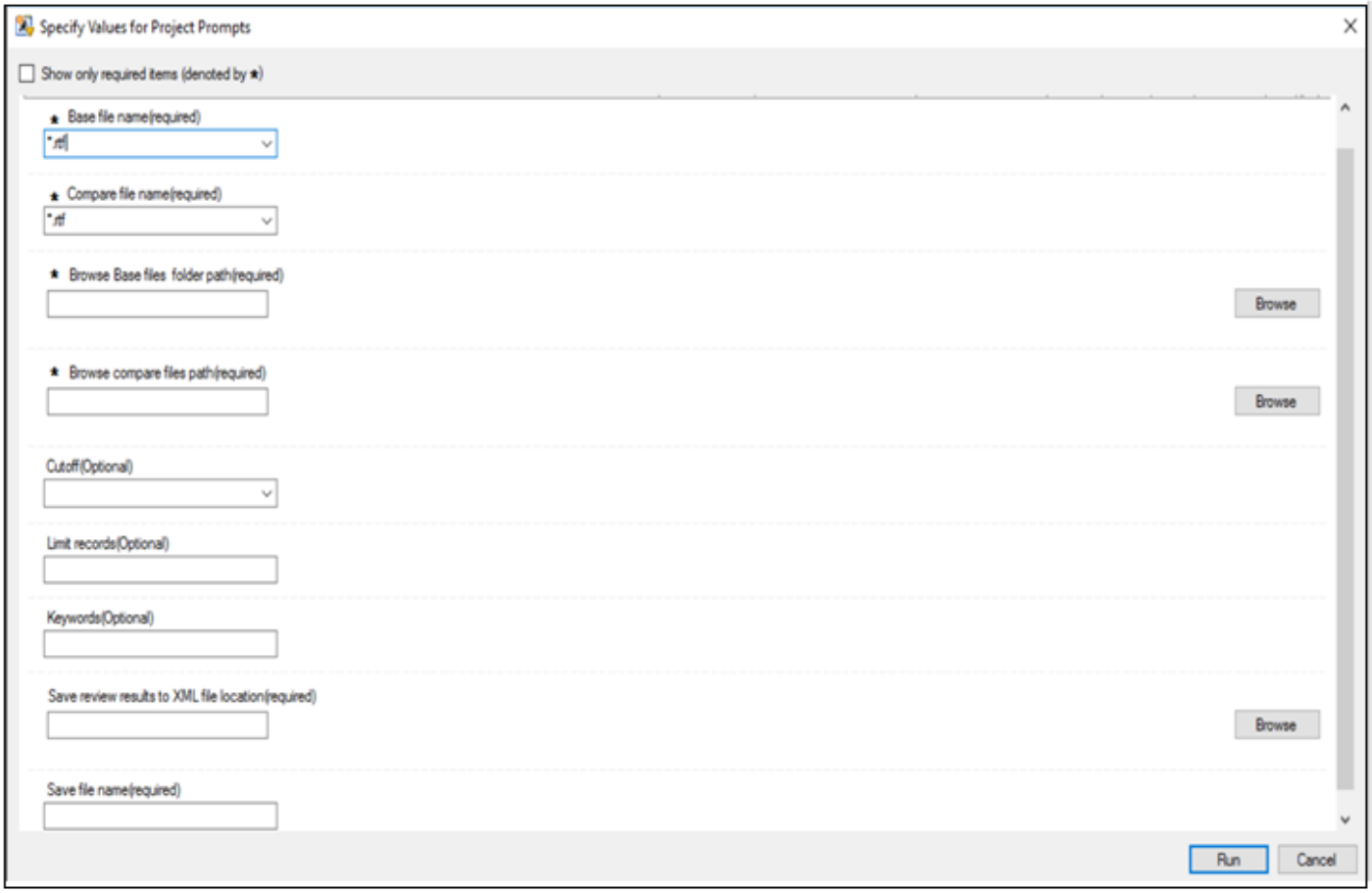

Figure 3: SAS EG Project interface

CREATING A SAS PROMPT

The SAS EG project prompt for passing values to the input macro parameters is described in the series of steps below.

Step1: Navigate to the Prompt Manager. This is one of the five icons located in the middle left of the SAS Enterprise Guide screen. The Prompt Manager is the fourth icon from the left as shown in figure 4. On clicking the “Add” button, an interface will appear that will allow you to create a SAS Prompt. This consists of two tabs: “General” and “Prompt Type and Values”.

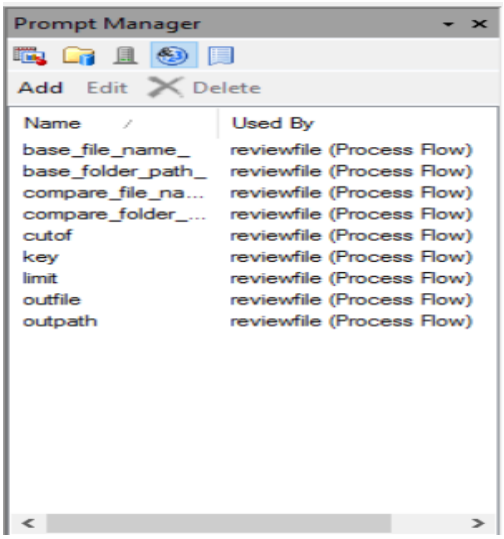

Figure 4: SAS EG Prompt Manager

Step 2: In this step, we add information to the General Tab. This tab allows users to choose the macro variable names, add a description, select a series of options such as “Requires a non-blank value” etc. as shown in figure 5.

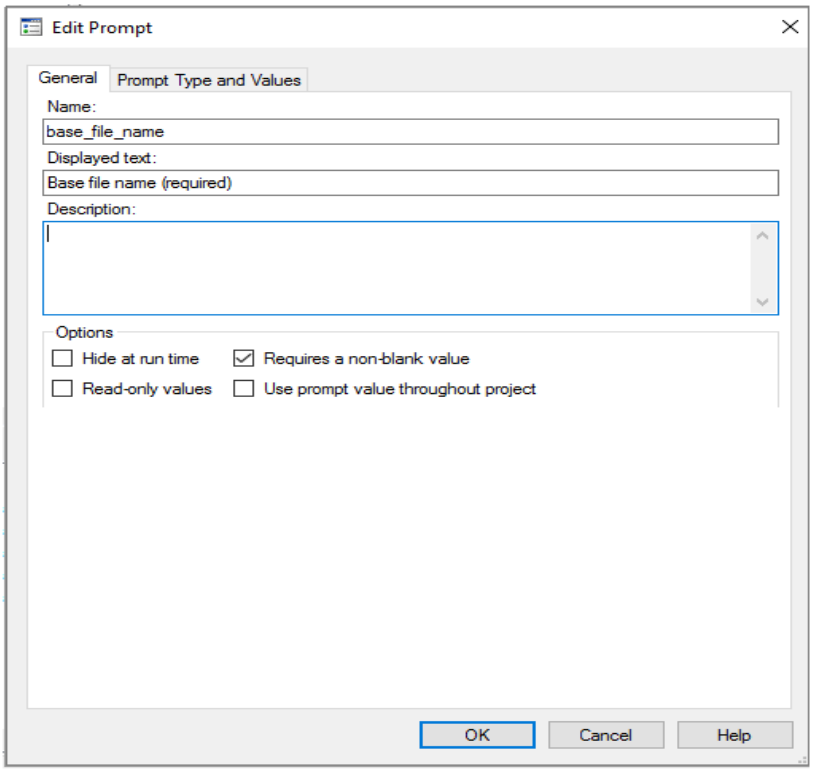

Figure 5: SAS EG Edit prompt General Tab

Step 3: In this step, we add information to the Prompt Type and Values Tab. This tab allows users to select and limit the type of data that can be allowed as input. This ta also allows users to add a default value and provide a “Hint” that helps the users to accurately input the parameters as shown in figure 6.

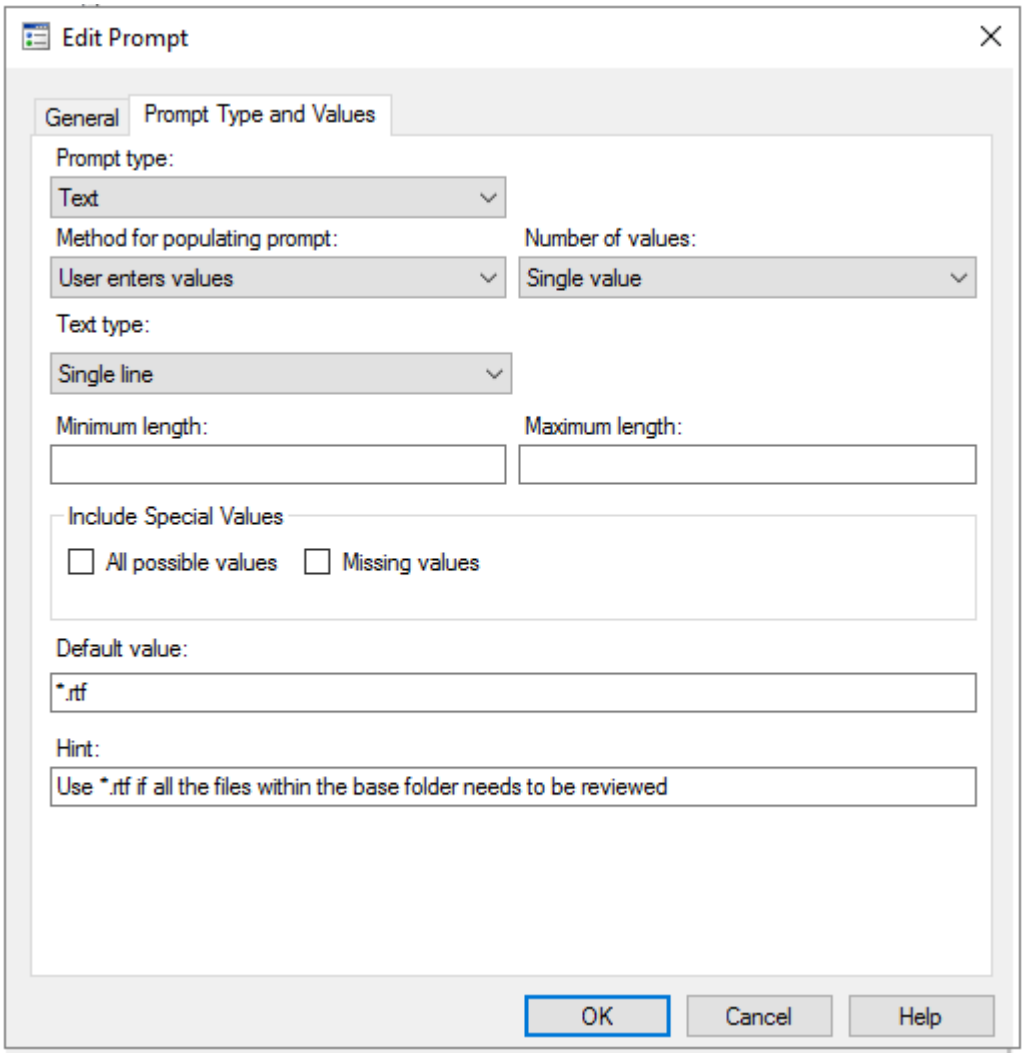

Figure 6: SAS EG Project interface Prompt Type and Values Tab

In this example, we are going to have the users enter in value by choosing the corresponding “Method of Populating prompt”. But we could also have them select one or more values from a static or dynamic list. A static list is a list of values that is entered during the prompt creation or loaded in once from a SAS data set variable. A dynamic list is a list of values from a SAS data set registered in the SAS metadata. If the variable values linked to the prompt changes, so do the values presented to the user when running the SAS prompt.

THE INPUT PARAMETERS FOR THE REVIEW FILE MACRO

This section will describe the purpose of each of the input parameters that are used in the utility with reference to Figure 3.

Base file name and Compare filename parameters are used to specify the RTF file name to be reviewed that is residing in the location specified in the folder path parameters. A value of "*.rtf" in the file name parameter will review all the files residing in the specified location. Users can navigate and select the file folders by clicking the Browse button.

The **Keywords Parameter** which is optional, allows the user to search the files for a specific keyword such as P-value as in figure 3 example.

The **Cutoff parameter** is used if we only need to fetch the differences beyond a specific cutoff value. In the current example, we need the outputs that have a change in P-value difference within +/-0.5. A value of 0 in the cutoff parameter will fetch all the outputs that have any changes in the P-value. A missing value in the cutoff parameter will prompt a search only by keywords. If both Keywords and Cutoff parameters are set to missing the comparison is done for the file contents between the Base and Compare folders.

The **Limit records parameter** is specified if the user wants to limit the number of records of output that are displayed, which is limited to 6 observations in the current example.

The **"Save review results XML file name" and "Save file name" parameters** allow the user to save the comparison results in an XML file format with the file name specified using the script below.

Clicking on the **Run button** will initiate the review process that will compile and run the %reviewfiles SAS macro.

A sample output summary in the Web-based interface or XML file is shown in figure 7 below. Note that the output summary includes the first title from the original RTF files and has hyperlinks to the files along with findings summary of differences.

Title	Base file	Compare file	Findings summary	Findings on key search
Summary and Analysis of Heart Rate Lowering Medication Use During Study Page 1 of 4	fqcmlowhra1	fqcmlowhra1	Acceptable differences	pvalue:exactly same or acceptable range
Summary and Analysis of Heart Rate Lowering Medication Use During Study Page 1 of 3	fqcmlowhrb1	fqcmlowhrb1	Acceptable differences	pvalue:exactly same or acceptable range
Summary of Heart Rate Lowering Medication Use During Study Page 1 of 2	fqcmlowhrc1_cr	fqcmlowhrc1_cr	Acceptable differences	pvalue:String: pvalue- not found
Summary and Analysis of Serotonergic Medication Use During Study (excluding PRN Use) Page 1 of 5	fqcmseronprna1	fqcmseronprna1	Acceptable differences	pvalue:exactly same or acceptable range
Summary and Analysis of Serotonergic Medication Use During Study (excluding PRN Use) Page 1 of 4	fqcmseronprnb1	fqcmseronprnb1	Acceptable differences	pvalue:exactly same or acceptable range
Summary of Serotonergic Medication Use During Study (excluding PRN Use) Page 1 of 4	fqcmseronprnc1_cr	fqcmseronprnc1_cr	Acceptable differences	pvalue:String: pvalue- not found
Summary of duration for TEAE of interest Page 1 of 2	fqteaedur_hl	fqteaedur_hl	Check for the differences	pvalue:String: pvalue- not found
Summary of TEAE Duration Page 1 of 1	fqteaedura	fqteaedura	Check for the differences	pvalue:String: pvalue- not found

Figure 7: XML output

%REVIEWFILES SAS MACRO

The process flow diagram of the %reviewfiles SAS macro is shown in figure 8. This macro consists of a series of steps that begins with checking the file names that are passed as input parameters to the macro and ends with displaying the summary of findings.

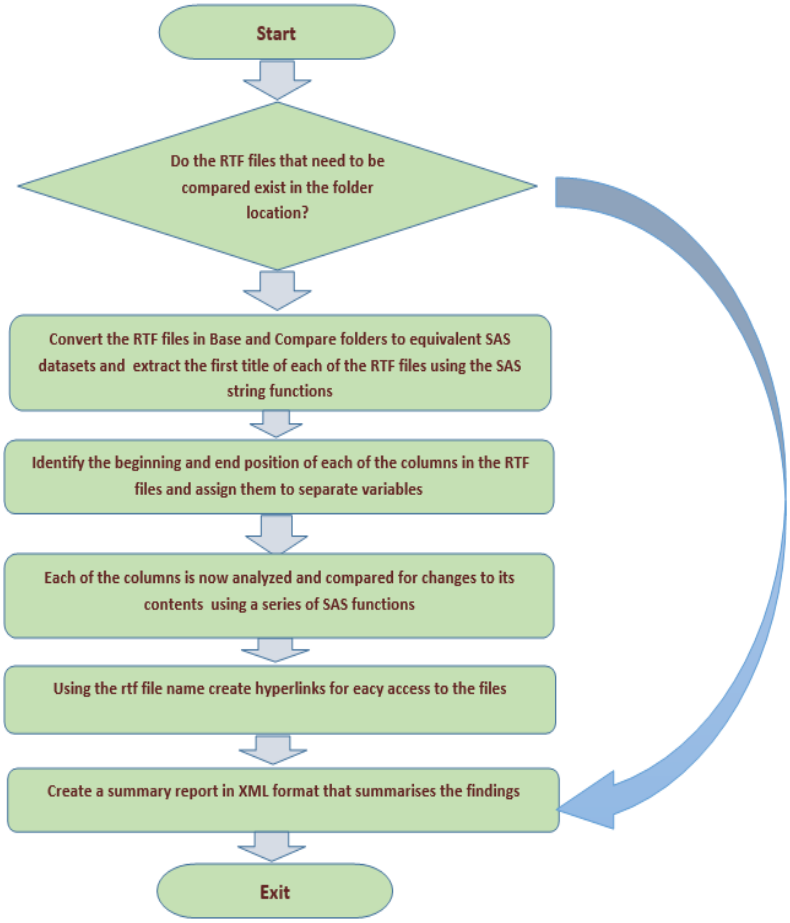

Figure 8: Process flow diagram of %reviewfile macro

The SAS macro call for passing SAS EG project prompt macro parameters into the %reviewfiles macro is as shown below:

```
%REVIEWFILES(BASE_FILE_NAME      = %STR(&BASE_FILE_NAME_),
             BASE_FOLDER_PATH    = %STR(&BASE_FOLDER_PATH_),
             COMPARE_FILE_NAME   = %STR(&COMPARE_FILE_NAME_),
             COMPARE_FOLDER_PATH = %STR(&COMPARE_FOLDER_PATH_),
             CUTOFF              = %STR(&CUTOF),
             LIMIT_RECORDS       = %STR(&LIMIT),
             KEY                 = %STR(&KEYWORDS),
             OUTPATH_            = %STR(&OUTPATH)
             OUTFILE_            = %STR(&OUTFILE);
```

Let's take a closer look at the overall process flow steps.

Step1: Check the validity of the file path
SAS Pipe option is used with the filename statement to check the legitimacy of the file path as shown in the code below.

```
%*=============================================================================================;
* Using the pipe statement check the validity of the file path
%*=============================================================================================;
%let proceed_flg=0;
 Filename Filename   pipe "dir &base_file_path ";
%*=============================================================================================;
* using infile read in the base file folder rtf file names
%*=============================================================================================;
Data _file_exist (keep=message)  ;
  Infile Filename truncover;
  Input filename $200.;
  Put filename=;
  if filename=:"dir: cannot access" then fileok=0;
   else fileok=1;
  if fileok=0 then Message=" NOTE: Unable to open file. Please check the  File path:  &ckprogram " ;
      if fileok=1 then Message=" NOTE: Success File path:  &ckprogram  is accurate" ;
  call symput ("fileok",fileok);
 Run;
```

The Pipe option along with "dir" and the file path will return a value ' dir: cannot access. <reference file path> ' if the file cannot be accessed in the specified location.
If the file location is not accurate, a macro variable (fileok) is assigned with a value 0 and the "message" variable is assigned with corresponding text as shown the code above.

Step 2: Read in all the RTF files that need to be reviewed from the folder using the "Infile" and "Filename" statements. In this step, the actual RTF file contents are extracted using the string functions as shown below.

```
.
data  base_&&dset&i  ;
 infile  "&&read&i" truncover;
 input base_prgm   $1000.;
 retain pagefind pagefind_inc;
 length base $1000;
%*=============================================================================================;
* Using string functions extract the actual rtf contents  from the base_prgm variable        ;
%*=============================================================================================;
 mergevar=_n_;
 x=length(base_prgm);
 rev_qa_prgm=trim(reverse(trim(base_prgm)));
 tot_pagenum=substr(compress(REV_QA_PRGM,'fo'),1,1) ;
%*=============================================================================================;
*in the following steps the obvious differences such as datetime stamp in the titles and file location     ;
* path in the footnotes are identified;
%*=============================================================================================;

if   index(lowcase(base_prgm), 'page') and substr(compress(REV_QA_PRGM,'fo}{'),1,1)+0>=0 then do; pagefind=1;pagefind_inc=pagefind; end;
if index(lowcase(base_prgm),'page')>0 and index(lowcase(base_prgm),'of')>0 then do; pagefind=1;pagefind_inc=pagefind; end;

 pagefind_inc=pagefind_inc+1;
  pagefind=0;
 if index(base_prgm,'fs16') >0 and substr(rev_qa_prgm,1,4) ne '6lsf'  then base=reverse(substr(rev_qa_prgm,1,index(rev_qa_prgm,'6lsf')-1 ));
run;
```

Step 3: The converted RTF file SAS dataset equivalent data is now analyzed for the differences and the key parameter search across all the variables and records are conducted. A sample code that is used to identify the key is shown in the code sample below.

```
 %let keyfound=;
data base_pval_&datanm(keep=  base typeb org_base indexmerge index   pval_full pval_location z bspval);
set base_&datanm;

retain pval_location typeb;
/*Key value search in both rows and columns to identify the values corresponding to key*/
 index= index(lowcase(compress(org_base,'-')),"&key");
 indexmerge= substr(base,1, index(trim(base),'  ' ));
  if index(lowcase(compress(org_base,'-')),"&key") >0 and index(lowcase(compress(indexmerge,'-')),"&key")>0 and typeb=' ' then typeb='row';
  else if index(lowcase(compress(org_base,'-')),"&key") >0 and typeb=' ' then typeb='col';
    if typeb ne '' then CALL SYMPUT('keyfound' , 'Y');
     else CALL SYMPUT('keyfound' , 'N');
  /*Convert < amd > symbol values in pvalues to an equvalent values for analysis*/

org_base_n=TRANWRD (org_base_n,' <.001','0.0001');
org_base_n=TRANWRD (org_base_n,'>.99','0.9999');
  keytyp="&key";
 array arr(*)$20 pv1 -pv30;
  array arrn(*)  pvb1 -pvb30;
  cnt=0;
 do j= 1 to 30;
 if  location>0 then do; arr(J)=scan(org_base_n,j);
 arrn(J)=TRANWRD (arr(J),'_','.') +0;
 if   arrn(J)>=500 then arrn(J)=.;
 if arrn(J) ne . then cnt+1;
  end;
 end;
```

Step 4: In this step hyperlinks for one-click access to the RTF files and XML summary output are being created using the code below. A sample output XML file is shown in figure 9.

```
       base_link=cats('=HYPERLINK("', 'file:///\\xxxx', "&base_folder_path_.",
tflname,
       '.rtf",',"',tflname,'")');

       ods path work.templ(update) sashelp.tmplmst(read) sashelp.styles(read);
       ods tagsets.ExcelXP style=sansPrinter options(embedded_titles='Yes'
                                  embed_titonce='no' suppress_bylines='yes'
                                  sheet_interval='bygroup' sheet_label=' '
                                  absolute_column_width="&colsp."
                                  autofit_height='yes'
                                  frozen_headers='5'
                                  frozen_rowheaders='yes'
                                  embedded_footnote='Yes' tabcolor="blue")
                                  file="&outpath./&outfilenm..xml" ;
       PROC REPORT DATA=allreprots2 NOWD HEADLINE HEADSKIP SPLIT="!" MISSING;
         columns Study base_rtf base_link compare_link summary_n
                   %if &keyn ne %then %do; summary_key;%end;;
        define study/"Study" width=25 left flow;
        define base_rtf/"Title" left WIDTH=100 ;
        define base_link/"Base file" left WIDTH=10 flow ; * flow;
        define compare_link/"Compare file" left WIDTH=10 flow;
        define summary_n/"Findings summary" left WIDTH=20 flow;
        %if &keyn ne %then %do;
        define summary_key/"Findings on key search" left WIDTH=20 flow;
        %end;
```

```
RUN;
```

Figure 9 shows a sample XML output with a keyword search for the string “p-value” at a cutoff value of >0.5 as search criteria. The index tab contains the extracted titles along with hyperlinks to the original RTF files as well as links to the summary tab for easy navigation. It can be noted that in the file fqteaedur_hl.rtf file there is no p-value to be searched. By clicking on the hyperlinks “findings” column, the page will be navigated to detailed summary reports as shown in figure 9 and figure 10.

Title	Base file	Compare file	Findings summary	Findings on key search
Summary and Analysis of NSAID Use During Study (excluding PRN Use) Page 1 of 3	fqcmnsaidnprna1	fqcmnsaidnprna1	Check for the differences	Check
Summary of duration for TEAE of interest Page 1 of 2	fqteaedur_hl	fqteaedur_hl	Exact Match	String: pvalue- not found

Figure 9: Index tab of the XML summary output

The detailed compare difference report tab provides further details on the actual differences between the two files as shown in figure 10. In this example, the fqteaedur_hl.rtf file differs in numbers as circled in red.

Base file	Compare file	Base file difference	compare file difference	Issue
fqteaedur_hl	fqteaedur_hl	On the >= 6th treated attack 145 (30.1) 96 (20.0) 75 (15.6) 62 (12.9) 33 (6.9) 12 (2.5) 5 (1.0) 481	On the >= 6th treated attack 146 (30.3) 96 (20.0) 75 (15.6) 62 (12.9) 33 (6.9) 12 (2.5) 5 (1.0) 481	Check
fqteaeonset_hl	fqteaeonset_hl	On the 1st treated attack 65 (18.5) 31 (9.2) 24 (7.1) 21 (6.3) 19 (5.7) 7 (2.1) 0 (0.0) 336	On the 1st treated attack 66 (18.6) 31 (9.2) 24 (7.1) 21 (6.3) 19 (5.7) 7 (2.1) 0 (0.0) 339	Check

Figure 10: Detailed compare summary report

The detailed keyword search report tab provides results related to the keyword search findings as shown in figure 11. In this example, the records that are related to “naproxen sodium” within the fqcmnasidnprna1.rtf files have a p-value difference of more than 0.5.

Base file	Compare file	Base file difference	compare file difference	Finding
fqcmnsaidnprna1	fqcmnsaidnprna1		nabumetone 1 (0.1) [0.1] 0 (0.0) [0.0] 0.00 .083	new record
fqcmnsaidnprna1	fqcmnsaidnprna1	naproxen sodium 4 (0.3) [0.3] 11 (0.3) [0.3] 1.07 (0.34, 3.39) .906	naproxen sodium 77 (5.7) [5.8] 150 (4.4) [4.4] 0.75 (0.56, 0.99) .043	pvalue difference >.5
fqcmnsaidnprna1	fqcmnsaidnprna1		vicoprofen 0 (0.0) [0.0] 1 (0.0) [0.0] .480	new record
fqteaedur_hl	fqteaedur_hl			String: pvalue- not found

Figure 11: Detailed keyword search report

CONCLUSION

Reviewing the outputs for changes between the multiple versions of outputs due to different data transfers can be very challenging especially when there are numerous files that need to be reviewed to identify the changes. The customary means of going over each file and comparing is not only monotonous but can impact the timeline of submission deliverables. Automating this review process using the reviewfiles utility not only makes the review of files easier but will also act as a tool to control the quality of the overall deliverables.

REFERENCES

Create Directory on Windows Without the Fleeting DOS Window, Available at URL:
www.lexjansen.com/pharmasug/2002/proceed/Coders/cc14.pdf

Using Unnamed Pipes: SAS(R) 9.3 Companion for Windows, Available at URL:
support.sas.com/documentation/cdl/en/.../n16puwsro9pakqn1jamy1vwyaqx6.html

Getting Started with SAS® Prompts, Available at URL:
https://support.sas.com/resources/papers/proceedings17/0818-2017.pdf

ACKNOWLEDGMENTS

Thanks to Yingbin Su, Consultant Statistician-Computation from Eli Lilly and Company for reviewing this paper and for providing valuable suggestions. Special thanks to Wei Zhou Director, Computational Statistics, Eli Lilly and Company for her leadership, persistent support and encouragement in this publication process.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

YouyingYu
Consultant Statistician-Computation, Global Statistical Science, Eli Lilly & Company
Work Phone: +1- 317- 433-1033
Email: yu_youying@lilly.com
Web: http:/www.lilly.com

Hrideep Antony
Principal Statistical Programmer, Clinical Division, Syneos Health
Work Phone: +1- 984 459 4785
Email: hrideep.antony@syneoshealth.com
Web: http:/www.syneoshealth.com