PharmaSUG 2012 - Paper DG04

iRobot: Listing Creator

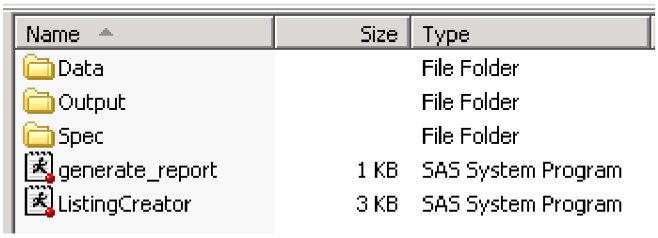
David Gray, PPD, Austin, Texas
Zhuo Chen, PPD, Austin, Texas
Suneela Gogineni, PPD, Austin, Texas

ABSTRACT

This paper introduces a method to automate the creation of a set of SAS® listing programs, each generating a listing based on input specifications. A MS Word macro inputs the listing specifications and outputs a MS Excel file containing the key information needed to generate the listing programs. A SAS® macro reads the MS Excel file and creates a set of SAS® programs, which then can be further customized.

INTRODUCTION

This listing creator is simple to use. Please see a snapshot below for folder structure:



- Data folder: all raw SAS® data sets used to create listings
- Output folder: all listing outputs
- Spec folder: original listing specifications, it could be one spec, or could be as many as you need.
- Generate_report.sas: a macro to create listing output (which can be PDF, or MS Excel or any output format you need). Each listing program will automatically call this macro.
- ListingCreator.sas: a macro to create a set of SAS® programs, each program will generate one listing.

STEP 1: USE THE MS WORD MACRO TO CREATE SPEC.XLS

This MS Word macro will run through each listing specification file in the Spec folder. It first opens one specification file, then searches for the key word "**Report Title:**" for the title, searches for the key word "**Program Name:**" for the SAS® program name, and searches for the key word "**Variable Names**" for a list of variables in the output (and the order in the listing output). Then the macro saves this information into SPEC.XLS file in the Spec folder. After one specification has been completed, it moves to the next, repeating the same steps until all specifications has been searched and all information has been saved in SPEC.XLS.

A completed example specification is shown in Display 1:

iRobot: Listing Creator, continued

"The version first placed into production should be Version 1

Version Number: "	0.01		Requested by:	
Report Title:	Listing 6: List of All Adverse Events		Request Date:	
			Report Due Date:	
Report Name:	Listing 6: List of All Adverse Events		Lead Programmer:	
Study Name:	ABCDEFG			
Sponsor Name:	XYZ		Internal/External Delivery:	☒ Internal
Program Name:	StudyAAA_06.sas			☐ External (To Client/3rd party)

Brief Overview of the Report's Intent:

DM REVIEW LISTING FOR ADVERSE EVENTS

Expected Output Format: ☒ .XLS File ☐ .PDF File ☐ .RTF File ☐ .CSV File ☐ Post Script for Printing ☐ Other _____

Report Format Criteria (for Output Formats of Report Only)**:

***Report Header and Footer are modified ONLY if sponsor requests modifications; otherwise PPD standard below is pulled into report c*

Report Header: Line 1 = Client: {Sponsor Name – left justified}
Line 2 = Protocol: {Study Name – left justified}
Line 3 = {Report Title -Center}

Report Footer: Line 1 = {{Program Path}.{File Name}, {Date / Time} - left justified}
Line 2 = {SAS version used to create data listing} – left justified
Line 2 = PPD, Inc – centered

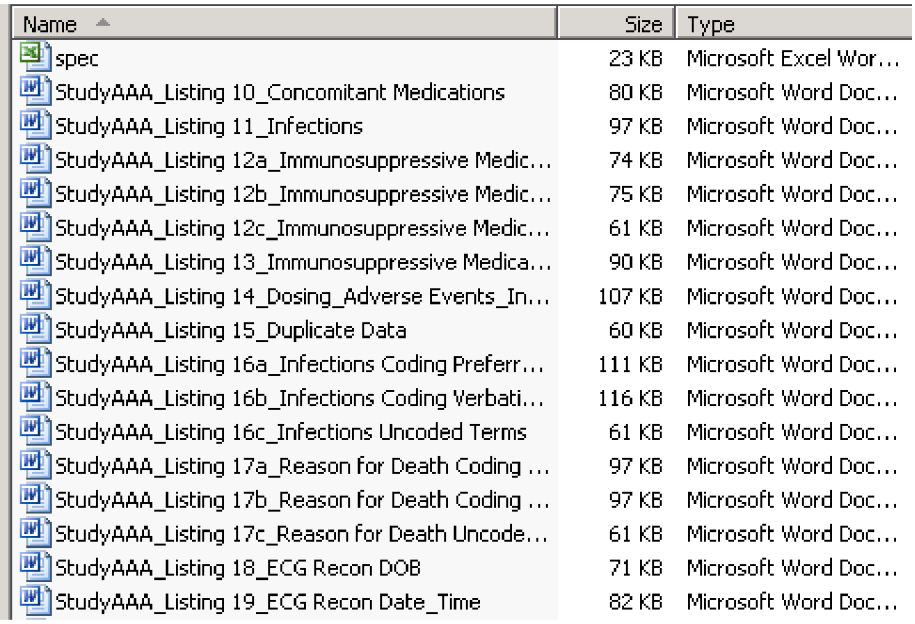
Data Requirements Section:

NOTE: Place the variables based on the order of display from left to right

Order	Annotated DCM/Data Source	Variable Names	Special Column Headings For Special Fields or Custom Variables	Criteria or Calculation definition ~Complete for required variables only	Compare (X)	Count (X)
1.	AEV	INV	Country		X	X
2.	AEV	PT	Patient No.	Report only first occurrence	X	X
3.	AEV	REC1N	Event No.		X	X
4.	AEV	AEVNAM1A	Adverse Event		X	X
5.	AEV	AEVSER1C	Was Event Serious?	Decode Serious [$CRN11_]	X	X
6.	AEV	AEVSEV1C	Severity	Decode Severity [$AEVS21_.]	X	X
7.	AEV	AEVSMR1C	Relationship	Decode Relation [$AEVS11_.]	X	X
8.	AEV	ACNTAK	Action Taken	Concatenate ACNTAK1N-ACNTAKAN (8 variables) when it is not blank, use "," to separate them.	X	X
9.	AEV	AEVSTT1O	Start Date	Output format DDMMMYYYY	X	X
10.	AEV	AEVEND1O	Stop Date	Output format DDMMMYYYY	X	X
11.	AEV	AEVCTU1C	Continuing?	Decode Continuing [$CRN11_.]	X	X

Display 1. Listing specification file.

Display 2 below is a snapshot of the Spec folder after the macro has been run. It contains individual listing specifications and the SPEC.XLS created by the word macro.



Name	Size	Type
spec	23 KB	Microsoft Excel Wor…
StudyAAA_Listing 10_Concomitant Medications	80 KB	Microsoft Word Doc…
StudyAAA_Listing 11_Infections	97 KB	Microsoft Word Doc…
StudyAAA_Listing 12a_Immunosuppressive Medic…	74 KB	Microsoft Word Doc…
StudyAAA_Listing 12b_Immunosuppressive Medic…	75 KB	Microsoft Word Doc…
StudyAAA_Listing 12c_Immunosuppressive Medic…	61 KB	Microsoft Word Doc…
StudyAAA_Listing 13_Immunosuppressive Medica…	90 KB	Microsoft Word Doc…
StudyAAA_Listing 14_Dosing_Adverse Events_In…	107 KB	Microsoft Word Doc…
StudyAAA_Listing 15_Duplicate Data	60 KB	Microsoft Word Doc…
StudyAAA_Listing 16a_Infections Coding Preferr…	111 KB	Microsoft Word Doc…
StudyAAA_Listing 16b_Infections Coding Verbati…	116 KB	Microsoft Word Doc…
StudyAAA_Listing 16c_Infections Uncoded Terms	61 KB	Microsoft Word Doc…
StudyAAA_Listing 17a_Reason for Death Coding …	97 KB	Microsoft Word Doc…
StudyAAA_Listing 17b_Reason for Death Coding …	97 KB	Microsoft Word Doc…
StudyAAA_Listing 17c_Reason for Death Uncode…	61 KB	Microsoft Word Doc…
StudyAAA_Listing 18_ECG Recon DOB	71 KB	Microsoft Word Doc…
StudyAAA_Listing 19_ECG Recon Date_Time	82 KB	Microsoft Word Doc…

Display 2. Spec folder.

Display 3 below is the snapshot of SPEC.XLS file created by the macro.

	A	B	
1	title	progname	keepvar
2	Listing 5a: Adverse Event Sorted by Preferred	StudyAAA_05a.sas	PT_TXG LLITTXG AEVNAM1A NMLCDG NLLTCDG MLLTCD(
3	Listing 5b: Adverse Event Sorted by Verbatim	StudyAAA_05b.sas	AEVNAM1A LLITTXG PT_TXG NMLCDG NLLTCDG MLLTCD(
4	Listing 6: List of All Adverse Events	StudyAAA_06.sas	INV PT REC1N AEVNAM1A AEVSER1C AEVSEV1C AEVSI
5	Listing 7: List of All Serious Adverse Events an	StudyAAA_07.sas	INV PT REC1N DSN AEVNAM1A AEVSER1C AEVSEV1C A
6	Listing 8: Treatment & Study Completion	StudyAAA_08.sas	INV PT VIS1O LSTSTR1O SBJDCN1C DCNRSN5C DCNRSN
7	Listing 9a: Concomitant Medications Sorted by	StudyAAA_09a.sas	PT_TXTA IT_TXTA CMDNAM1A ATCC1A ATCT1A ATCC2A A
8	Listing 9b: Concomitant Medications Sorted by	StudyAAA_09b.sas	CMDNAM1A ATCC1A ATCT1A ATCC2A ATCT2A ATCC3A A
9	Listing 9c: Concomitant Medications Uncoded	StudyAAA_09c.sas	CMDNAM1A PT REC1N
10	Listing 10: List of All Concomitant Medications	StudyAAA_10.sas	INV PT REC1N SUBSETSN CMDNAM1A CMDRSN1A CMDS
11	Listing 11: Infections	StudyAAA_11.sas	INV PT REC1N INFNAM1A SMPSRC1C ORGNAM1A INFTYI
12	Listing 12a: Immunosuppressive Medications S	StudyAAA_12a.sas	PT_TXTA IT_TXTA CMDNAM1A NDRGPTA NDGCDA DRUGC
13	Listing 12b: Immunosuppressive Medications S	StudyAAA_12b.sas	CMDNAM1A IT_TXTA PT_TXTA NDRGPTA NDGCDA DRUGC
14	Listing 12c: Immunosuppressive Medications U	StudyAAA_12c.sas	CMDNAM1A PT REC1N

Display 3. SPEC.XLS.

The macro can be easily modified if you need to use alternative search strings to extract the information for listing title, program name, and the list of variable names.

The source code for the word macro is on Appendix A of this paper.

STEP 2: USE LISTING CREATOR.SAS TO CREATE A SET OF SAS® LISTING PROGRAMS

This SAS® program contains two macros. The first macro, %create_one program() will create one SAS® listing program each time it is called. Note you can customize the header section to fit your standard.

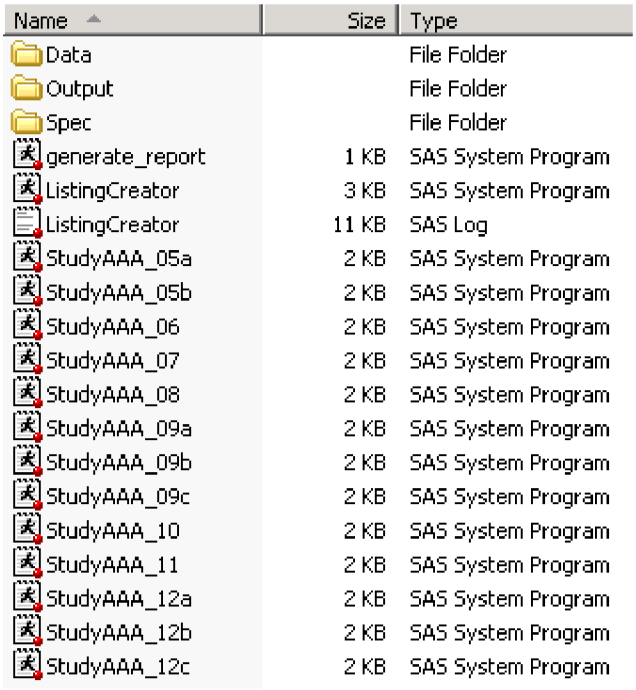
```
%MACRO create_one_program(title=,progname=,keepvar=);
  DATA _null_ ;
    FILE "&&progname..SAS";
      put "/*************************************************************";
      put "*          STUDY NAME: Study AAA                              ";
      put "*            PROTOCOL: Protocol BBB                           ";
      put "*        PROGRAM NAME: &progname..SAS                         ";
      put "*         SAS VERSION: 9.1.3                                  ";
      put "*              AUTHOR:                                        ";
      put "*        DATE CREATED:                                        ";
      put "*             PURPOSE: &title                                 ";
      put "*         INPUT FILES:                                        ";
      put "*        OUTPUT FILES:                                        ";
      put "*************************************************************/";
      put "                                                              ";
      put '%' 'INCLUDE "generate_report.sas";                            ';
      put "                                                              ";
      put '%' "LET title    = &title ;                                   ";
      put '%' "LET progname = &progname ;                                ";
      put '%' "LET keepvar  = &keepvar;                                  ";
      put "                                                              ";
      put 'LIBNAME data ".\data";                                        ';
      put "                                                              ";
      put 'DATA &progname (KEEP=&keepvar);                               ';
      put "  /* put your code here */                                   ";
      put "RUN;                                                         ";
      put "                                                              ";
      put "  /* add footnotes if need */                                 ";
      put 'FOOTNOTE3 j=l "\\path\&&progname..SAS";                       ';
      put '%generate_report(&title,&progname,&keepvar);                  ';
   run;
%MEND create_one_program;
```

iRobot: Listing Creator, continued

The 2nd macro %create_all_programs() reads in the SPEC.XLS, then calls the first macro %create_one_program() one by one for each listing, passing the listing title, SAS® program name and variables to generate one corresponding SAS® program for each listing specification.

```
%MACRO create_all_programs();
  %LET specification=.\spec\spec.xls;

  PROC IMPORT DATAFILE="&specification" OUT=spec;
    GETNAMES=yes;
  RUN;

  DATA spec;
    SET spec;
    temp = INDEX(progname,".sas");
    IF temp>0 THEN progname=SUBSTR(progname,1,temp-1);
  RUN;

  DATA _null_;
    SET spec END=eof ;
    CALL SYMPUT('title'||TRIM(LEFT(_N_)), TRIM(LEFT (title)));
    CALL SYMPUT ('progname'|| TRIM(LEFT (_N_)), TRIM(LEFT (progname)));
    CALL SYMPUT ('keepvar'|| TRIM(LEFT (_N_)), TRIM(LEFT (keepvar)));
    IF eof THEN do;
      CALL SYMPUT ('MAXDS', TRIM(LEFT (_N_)));
    END;
  RUN;

  %LET max = &MAXDS.;
  %LOCAL j;
  %DO j=1 %TO &max ;
      %create_one_program(title=&&title&j,
                          progname=&&progname&j,
                          keepvar=&&keepvar&j);
  %END;
%MEND create_all_programs;
```

Display 4 below is a snapshot of the directory after execution of %create_all_programs.



Name	Size	Type
Data		File Folder
Output		File Folder
Spec		File Folder
generate_report	1 KB	SAS System Program
ListingCreator	3 KB	SAS System Program
ListingCreator	11 KB	SAS Log
StudyAAA_05a	2 KB	SAS System Program
StudyAAA_05b	2 KB	SAS System Program
StudyAAA_06	2 KB	SAS System Program
StudyAAA_07	2 KB	SAS System Program
StudyAAA_08	2 KB	SAS System Program
StudyAAA_09a	2 KB	SAS System Program
StudyAAA_09b	2 KB	SAS System Program
StudyAAA_09c	2 KB	SAS System Program
StudyAAA_10	2 KB	SAS System Program
StudyAAA_11	2 KB	SAS System Program
StudyAAA_12a	2 KB	SAS System Program
StudyAAA_12b	2 KB	SAS System Program
StudyAAA_12c	2 KB	SAS System Program

Display 4. Directory after execution of %create_all_programs.

STEP 3: MODIFY EACH LISTING PROGRAM THEN CREATE LISTING OUTPUT

Each SAS® listing program already has listing template and key information for that corresponding listing. Let's look at one example, StudyAAA_06.sas, which is generated automatically by this macro based on the specification for listing 06 in this StudyAAA.

As you can see below the SAS® program name, listing title and the variables have been populated in the SAS® listing programs. All you need are a few minor modifications to finalize each program. See the ***bolded italicized*** code below.

```
/*************************************************************
*           STUDY NAME: Study AAA
*              PROTOCOL: Protocol BBB
*         PROGRAM NAME: StudyAAA_06.SAS
*          SAS VERSION: 9.1.3
*                AUTHOR:
*         DATE CREATED:
*              PURPOSE: Listing 6: List of All Adverse Events
*          INPUT FILES:
*         OUTPUT FILES:
*************************************************************/


%INCLUDE "generate_report.sas";
```

```
%LET title    = Listing 6: List of All Adverse Events ;
%LET progname = StudyAAA_06 ;
%LET KEEPVAR  = INV PT REC1N AEVNAM1A AEVSER1C AEVSEV1C AEVSMR1C ACNTAK AEVSTT1O
AEVEND1O AEVCTU1C;


LIBNAME DATA ".\data";


DATA &progname (KEEP=&keepvar);

  /* put your code here */


RUN;

  /* add footnotes if needed */


FOOTNOTE3 j=l "\\path\&&progname..SAS";
%generate_report(&title,&progname,&keepvar);
```

You can manipulate the ***bolded italicized*** code below so it can read in the proper raw data from Data folder or derive certain variables. You can also add footnotes if needed.

```
DATA &progname (KEEP=&keepvar);
  SET DATA.AEVAEV;
  LENGTH ACNTAK $10. ;
  ACNTAK = TRIM(LEFT(acntak1n))||TRIM(LEFT(acntak2n));
  LABEL ACNTAK = "Action Taken";
RUN;
FOOTNOTE1 j=l "Note: Continuing at final exam 1=Yes, 0=No. ";
```

After execution, the listing will be output to the OUTPUT folder and will have the same name as the SAS® listing program, e.g. StudyAAA_06.RTF in this case. Please see Display 5.

1

Study AAA
Protocol BBB
Listing 6: List of All Adverse Events

Investigator	Patient	Record number	Adverse Event	Seriousness	Severity	Relationship to study drug	Action Taken	Adverse event start (date)	Adverse event end (date)	Continuing at final exam
AUS	018000002	1.00	PULMONARY THROMBOEMBOLISM	1	3	0		20100625	20100625	
AUS	018000002	2.00	LEFT LEG DEEP VEIN THROMBOSIS	1	3	0	0	20100625	20100625	
AUS	018000002	3.00	ISCHAEMIC HEART DISEASE	1	3	0	0	20100625	20100625	
AUS	018000003	1.00	TRANSPLANT WOUND PAIN		2	0		20100529	20100610	
AUS	018000003	2.00	URINARY URGENCY-INTERMITTENT		1	0	0	20100617		1
AUS	018000003	3.00	PARASTHESIA BILATERAL FEET		2	0	0	20100625		1
AUS	018000003	4.00								
AUS	018000003	5.00	DIARRHOEA		2	0	0	20100810	20100812	
AUS	018000003	6.00	ACHE OVER KIDNEY-INTERMITTENT		1	0	0	20100823		1
AUS	018000003	7.00	BILATERAL LIMB OEDEMA		2	0		20100906	20100924	
AUS	018000003	8.00	DIARRHOEA-INTERMITTENT		1	0		20100919		1
AUS	018000003	9.00	ANXIETY		2	0		20110113		1
AUS	018000003	10.00	MUSCULOSKELETAL BACK PAIN		3	0		20110522	20110524	
AUS	018000003	11.00	MUSCULOSKELETAL BACK PAIN		1	0	0	20110525		
AUS	018000004	1.00	NAUSEA-INTERMITTENT		2	1		20100603	20100914	
AUS	018000004	2.00	VOMITING-INTERMITTENT		1	1		20100604	20100914	
AUS	018000004	3.00	FEBRILE		2	0		20100604	20100608	
AUS	018000004	4.00								
AUS	018000004	5.00	HAEMATURIA POST RENAL BIOPSY	1	1	0		20100611	20100613	
AUS	018000004	8.00	WORSENING GASTROOESOPHAGEAL REFLUX DISEASE		2	0		20100605		1

Note: Continuing at final exam 1=Yes, 0=No.

\\path\StudyAAA_06.SAS

Display 5. StudyAAA_06.RTF

MACRO %GENERATE_REPORT()

The source code of macro %generate_report() is below. Since this is such a simple macro, you can easily customize it to PDFor EXCEL etc. output format as desired.

```
%MACRO generate_report(title,progname,keepvar);
  OPTIONS MISSING="" NODATE ORIENTATION = landscape ;
  TITLE1 "Study AAA";
  TITLE2 "Protocol BBB";
  TITLE3 "&title";

  ODS RTF FILE=".\output\&&progname..rtf";
  PROC REPORT DATA=&progname HEADLINE HEADSKIP SPLIT='$' MISSING NOWD;
    COLUMN &keepvar;
  RUN;

  ODS RTF CLOSE;
%MEND generate_report;
```

CONCLUSION

When you have a large number of listings to produce, utilizing a standardized specification format and the automation macros presented in this paper you will realize substantial time savings. The macros presented may be further customized to further automate the program generation and output format. Time is money, so let iRobot do the heavy lifting!

Appendix A: SOURCE CODE FOR THE WORD MACRO:

Note that this macro requires the SPEC.XLS file to be present and will replace the contents each time it is executed. In order to execute this macro the MS Excel 11.0 or 12.0 Object Library reference must be enabled.

```
Sub ReadSpecs()
' MS Word Macro to read in the specification documents and output to SPEC.XLS
' Modify the directory path to the SPEC.XLS Document below
    mydirectory = "."
    ChangeFileOpenDirectory mydirectory

    Dim CurrFile As String
    Dim CurrTitle As String
    Dim CurrPgmName As String
    Dim CurrVar As String
    Dim VarLst As String
    Dim row As Integer
    Dim col As Column
    Dim rng As Range
    Dim CurrDoc As Document
    Dim XlApp As Excel.Application
    Dim IdSheet As Integer
    Dim XlSheet As Excel.Worksheet
    Dim OpenStatus As Boolean
    Dim ws As Worksheet
    Dim row_counter As Integer

' Open the excel workbook
    Set XlApp = New Excel.Application
    XlApp.Visible = False
    Set xlbook = XlApp.Workbooks.Open(mydirectory & "\SPEC.xls")
    Set XlSheet = xlbook.ActiveSheet
    XlSheet.Cells.Clear
    row_counter = 1
    XlSheet.Cells(row_counter, 1) = "title"
    XlSheet.Cells(row_counter, 2) = "progname"
```

iRobot: Listing Creator, continued

```
    XlSheet.Cells(row_counter, 3) = "keepvar"
    row_counter = row_counter + 1
    XlApp.ScreenUpdating = False

 ' Open each specification file
    CurrFile = Dir(mydirectory & "\*.doc")
    Do While CurrFile <> ""
        Documents.Open FileName:=CurrFile

' Process each file
' Get the title
    Selection.Find.ClearFormatting
    With Selection.Find
        .Text = "Report Title"
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute
    Selection.MoveRight Unit:=wdCell
    CurrTitle = Selection.Text

' Get the Program Name
    Selection.Find.ClearFormatting
    With Selection.Find
        .Text = "Program Name"
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
```

```
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute
    Selection.MoveRight Unit:=wdCell
    CurrPgmName = Selection.Text

' Get the variable list
    Selection.Find.ClearFormatting
    With Selection.Find
        .Text = "Variable Names"
        .Replacement.Text = ""
        .Forward = True
        .Wrap = wdFindContinue
        .Format = False
        .MatchCase = False
        .MatchWholeWord = False
        .MatchWildcards = False
        .MatchSoundsLike = False
        .MatchAllWordForms = False
    End With
    Selection.Find.Execute

    ' set col to the column with the variable names are located

    VarLst = ""
    'Store the variables in VarLst separated by a comma

    For row = 2 To Selection.Tables(1).Rows.Count

          ' Strip the last char (cell marker) and add to the list and the CRLF
          CurrVar = Selection.Tables(1).Cell(row:=row, Column:=3).Range.Text
          CurrVar = Left(CurrVar, Len(CurrVar) - 1)
          CurrVar = LTrimCRLF(CurrVar)

         If row = 2 Then
          VarLst = CurrVar
         Else
          VarLst = VarLst & "," & CurrVar
         End If
    Next
```

```
' update the SPEC.XLS file
    XlSheet.Cells(row_counter, 1) = CurrTitle
    XlSheet.Cells(row_counter, 2) = CurrPgmName
    XlSheet.Cells(row_counter, 3) = VarLst
    row_counter = row_counter + 1

' Close the specification file
        ActiveDocument.Close

' Call the Dir command to get the next filename
        CurrFile = Dir
    Loop

' Close the SPEC.XLS file
XlApp.ActiveWorkbook.Save
XlApp.ActiveWorkbook.Close

End Sub

Function LTrimCRLF(s As String) As String
Dim index As Integer, start As Integer, strlen As Integer
Dim c As String
strlen = Len(s)
index = 1
start = -1
Do While (index <= strlen) And (start = -1)
c = Mid(s, index, 1)
If (c = vbCr) Or (c = vbLf) Then
index = index + 1
Else
start = index
End If
Loop
If start = -1 Then
LTrimCRLF = ""
Else
LTrimCRLF = Mid(s, start)
End If

End Function
```

ACKNOWLEDGMENTS

We would like to thank our colleagues in the Biostatistics Department at PPD who provided comments on drafts of this paper.

CONTACT INFORMATION

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

David Gray, Zhuo Chen, Suneela Gogineni
PPD
7551 Metro Center Drive, Suite 300
Austin, TX 78744
E-mail: david.gray@ ppdi.com, zhuo.chen@ppdi.com, Suneela.Gogineni@ppdi.com