

Using tagsets MSOFFICE2K to customise your Excel report

Dominique Pinet, KEYRUS Biopharma, Paris, France

Paper TS10

Abstract.

The purpose of this paper is to show how SAS9 can be use to customise Excel listings. This method minimises the programming in SAS with the use of templates and ODS TAGSETS MSOFFICE2K. The format management on the Excel file is performed by XML code that is inserted in the template. This method does not use Access to PC File.

Introduction

Excel is a very quick and pratical way to present data in a form allowing control and validation of data. With SAS9, it is possible to customize Excel reporting using ODS TAGSETS. This paper demonstrates how to develop automated reports in Excel format. It also demonstrates how to put a link to another file into an Excel file using the report procedure.

The use of templates allows SAS development time to be reduced and avoids much of the manipulation in Excel.

The purpose of this paper is to represent the different stages in the production of the Excel file below:

1	Country	patient	Prefered term	Start date	End date	Time
2	FRANCE	0000000001	Anterigram coronary	15/01/2006	23/01/2006	8
3	FRANCE		Calculus urinary	13/01/2006	22/01/2006	9
4	FRANCE	0000000003	Atrial fibrillation	12/01/2006	13/01/2006	1
5	FRANCE	0000000004	Bronchospasm	19/01/2006	21/01/2006	2
6	FRANCE	0000000005	Erectil dysfunction	20/01/2006	23/01/2006	3
7	FRANCE	0000000006	Skin lesion excision	01/02/2006	05/02/2006	4
8	FRANCE	0000000007	Gastric disorder	03/01/2006	08/01/2006	5
9	FRANCE	0000000008	Hyperhidrosis	24/01/2006	30/01/2006	6
10	FRANCE	0000000009	Anxiety	08/02/2006	15/02/2006	7
11	FRANCE	0000000011	Hyperhidrosis	12/01/2006	31/01/2006	5
12	FRANCE		Dizziness postural	11/01/2006	29/01/2006	5
13	FRANCE	0000000012	Dizziness postural	08/01/2006	18/01/2006	10
14	FRANCE	0000000013	Metabolic syndrome	10/01/2006	21/01/2006	3
15	FRANCE	0000000014	Arteriogram coronary	05/02/2006	17/02/2006	3
16	FRANCE	0000000015	Myocardial infarction	04/02/2006	17/02/2006	4
17	FRANCE	0000000016	Hyperhidrosis	05/01/2006	19/01/2006	4
18	FRANCE	0000000017	Atrial fibrillation	09/01/2006	24/01/2006	4
19	FRANCE	0000000018	Gastric disorder	12/02/2006	28/02/2006	4
20	FRANCE	0000000019	Atrial fibrillation	14/02/2006	03/03/2006	5
21	SPAIN	0000000021	Hyperhidrosis	09/01/2006	06/02/2006	7
22	SPAIN		Anxiety	28/01/2006	26/02/2006	8
23	FRANCE	0000000022	Atrial fibrillation	30/01/2006	19/02/2006	5
24	FRANCE	0000000023	Bronchospasm	06/01/2006	27/01/2006	6
25	FRANCE	0000000024	Anxiety	07/01/2006	29/01/2006	6
26	FRANCE	0000000025	Gastric disorder	22/01/2006	14/02/2006	6
27	FRANCE	0000000026	Atrial fibrillation	06/02/2006	02/03/2006	6
	SPAIN			10/01/2006	06/02/2006	7

Creation of the template

This template is created with the tagsets MSOFFICE2K. Below, you can see the basic template. The code for layout control will be inserted step by step into this basic template.

```
PROC TEMPLATE;
  DEFINE tagset tagsets.minimal;
    parent=tagsets.MSOffice2K;
  DEFINE event doc;
  START:
    PUT '<html xmlns:x="urn:schemas-microsoft-com:office:excel">' NL;
  FINISH:
    PUT "</html>" NL;
  END;
  DEFINE event doc_head;
  START:
    PUT "<head>" NL;
    PUT VALUE NL;
  FINISH:
    PUT "<!--[if gte mso 9]><xml>" NL;
    PUT "<x:ExcelWorkbook>" NL;
    PUT "  <x:ExcelWorksheets>" NL;
    PUT "    <x:ExcelWorksheet>" NL;
    PUT "      <x:Name>sheet1</x:Name>" NL;
    PUT "      <x:WorksheetOptions>" NL;
    PUT "        <x:Print>" NL;
    PUT "          <x:ValidPrinterInfo/>" NL;
    PUT "        </x:Print>" NL;
    PUT "      </x:WorksheetOptions>" NL;
    PUT "    </x:ExcelWorksheet>" NL;
    PUT "  </x:ExcelWorksheets>" NL;
    PUT "</x:ExcelWorkbook>" NL;
    PUT "</xml><![endif]-->" NL;
    PUT "</head>" NL;
  END;
END;
RUN;
```

Freezing Rows and Columns

Rows and columns can be frozen. To define the row and the columns to be frozen, you have to set the value for the first two tags (SplitHorizontal and TopRowBottomPane) for horizontal split, and the next two tags (SplitVertical and LeftColumnRightPane) for vertical split. In the example below, the first column and the first line are frozen.

```
PUT "  <x:Name>sheet1</x:Name>" NL;
PUT "    <x:WorksheetOptions>" NL;
/***** frozen panes code *****/
put "      <x:FreezePanes/>" NL;
put "      <x:FrozenNoSplit/>" NL;
put "      <x:SplitHorizontal>1</x:SplitHorizontal>" NL;
put "      <x:TopRowBottomPane>1</x:TopRowBottomPane>" NL;
put "      <x:SplitVertical>1</x:SplitVertical>" NL;
put "      <x:LeftColumnRightPane>1</x:LeftColumnRightPane>" NL;
put "      <x:DisplayPageBreak/>" NL;
/*****
PUT "        <x:Print>" NL;
PUT "          <x:ValidPrinterInfo/>" NL;
```

Landscaping and reducing the print size.

The framed text is added to the template code to set the orientation to landscape.

```
PROC TEMPLATE;
    DEFINE tagset tagsets.minimal;
        parent=tagsets.MSOffice2K;
        DEFINE event initialize;
            set $orientation tagset_alias;
            trigger set_just_lookup;
            trigger set_nls_num;
        END;
    DEFINE event doc;
    START:
    PUT '<html xmlns:x="urn:schemas-microsoft-com:office:excel">' NL;
    FINISH:
    PUT '</html>' NL;
    END;
    DEFINE event doc_head;
    PUT '<head>' NL;
    put '<style> @page {mso-page-orientation:landscape} </style>' NL;
    PUT VALUE NL;
    do / if cmp($orientation,'landscape');
    PUT '<!--[if gte mso 9]><xml>' NL;
    PUT '<x:ExcelWorkbook>' NL;
    ...
    PUT '</x:ExcelWorkbook>' NL;
    PUT '</xml><![endif]-->' NL;
    done;
    PUT '</head>' NL;
END;
END;
RUN;
```

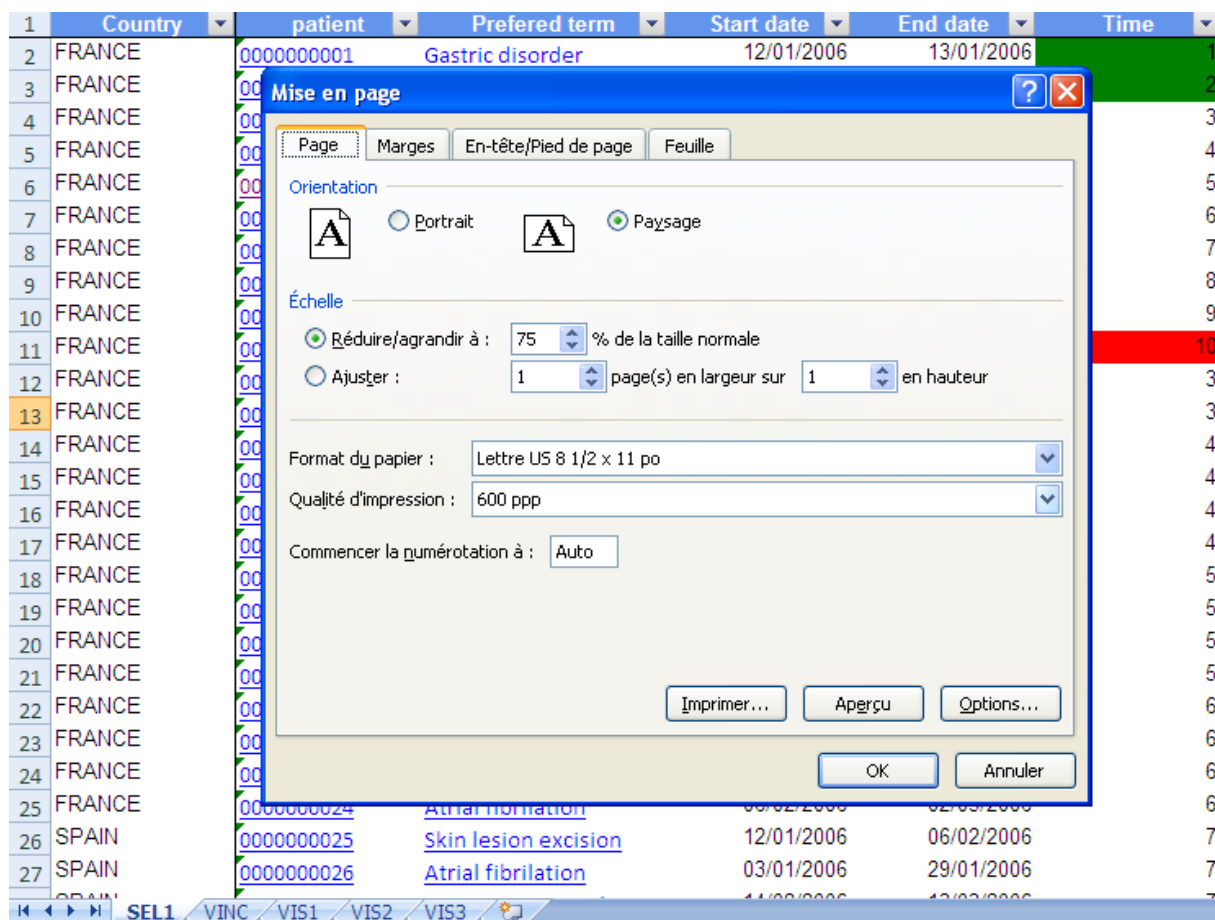
The code below must be inserted during the creation of the Xls file. The use of the option ALIAS= allows the page orientation to be defined.

```
ODS TAGSETS.minimal FILE="&PATH.\&visit..xls" ALIAS='landscape';
```

To reduce the print size to 75%, you have to specify this value in the print option between the SCALE tags.

```
put " <x:Print>" NL;
put " <x:ValidPrinterInfo/>" NL;
/***** Reduce print size to 75% *****/
put " <x:Scale>75</x:Scale>" NL;
put " </x:Print>" NL;
```

PhUSE 2007



Conditional formatting.

The code below allows the "TIME" column to be colour coded. If the value is between 10 and 11, the cell background color is set to red. If the value is between 1 and 2, the cell background colour is set to green. It is possible to put several conditions in this section.

```
put " </x:ExcelWorksheets>" NL;
put "</x:ExcelWorkbook>" NL;
put " <x:ConditionalFormatting>" NL;
put " <x:Range>F:F</x:Range>" NL;
put " <x:Condition>" NL;
put " <x:Qualifier>Between</x:Qualifier>" NL;
put " <x:Value1>10</x:Value1>" NL;
put " <x:Value2>11</x:Value2>" NL;
put " <x:Format Style='background:red;mso-pattern:none' />" NL;
put " </x:Condition>" NL;
put " <x:Condition>" NL;
put " <x:Qualifier>Between</x:Qualifier>" NL;
put " <x:Value1>1</x:Value1>" NL;
put " <x:Value2>2</x:Value2>" NL;
put " <x:Format Style='background:green;mso-pattern:none' />" NL;
put " </x:Condition>" NL;
put " </x:ConditionalFormatting>" NL;
put "</xml><![endif]-->" NL;
```

PhUSE 2007

1	Country	patient	Preferred term	Start date	End date	Time
2	FRANCE	0000000001	Gastric disorder	12/01/2006	13/01/2006	1
3	FRANCE	0000000002	Bronchospasm	19/01/2006	21/01/2006	2
4	FRANCE	0000000003	Erectil dysfunction	20/01/2006	23/01/2006	3
5	FRANCE	0000000004	Skin lesion excision	01/02/2006	05/02/2006	4
6	FRANCE	0000000005	Gastric disorder	03/01/2006	08/01/2006	5
7	FRANCE	0000000006	Hyperhidrosis	24/01/2006	30/01/2006	6
8	FRANCE	0000000007	Anxiety	08/02/2006	15/02/2006	7
9	FRANCE	0000000008	Anterogram coronary	15/01/2006	23/01/2006	8
10	FRANCE	0000000009	Gastric disorder	13/01/2006	22/01/2006	9
11	FRANCE	0000000010	Dizziness postural	08/01/2006	18/01/2006	10

If you modify the value in the "TIME" column and if this new value is between the tags values defined in the template, the background colour changes automatically.

Editing of the listing and inserting hypertext links.

The listing editing is done with the REPORT procedure. It allows you to:

- Set automatic filters on all columns.
- Show " Preferred term " as a hypertext with a link to an HTML file which groups the number of appearances of this AE on every visit.
- Show patient's number as a hypertext with a link to an HTML file grouping each patients AEs over the course of the study.
- Format patient's number to keep 0 at the start of the number.

Warning, the automatic filters to more than one column must be set if the columns are continuous,

```
ODS RESULTS OFF;
ODS LISTING CLOSE;
ODS TAGSETS.minimal FILE="&PATH.\&visit..xls" ALIAS='landscape';
```

```
PROC REPORT DATA=phuse07.AE (WHERE=(visit="&visit")) NOWD SPLIT='@'
```

```
STYLE(header)={tagattr="x:autofilter="all"x:autofilterrange=A1:G1"};
```

```
  COLUMN country visit patident dptcod first_date last_date duree;
  DEFINE country / style={cellwidth = 3 cm};
  DEFINE visit / noprint;
```

```
  DEFINE patident / order style={cellwidth = 3 cm htmlstyle="mso-number-format:@"}; /* cell formatting */
```

```
  DEFINE dptcod / order style={cellwidth = 4 cm};
  DEFINE First_date / style={cellwidth = 3 cm};
  DEFINE last_date / style={cellwidth = 3 cm};
  DEFINE duree / style={cellwidth = 3 cm};
```

```
/*Hypertext links */
```

```
  COMPUTE AFTER patident;
    NUM = INPUT(patident,$10.);
  ENDCOMP;
  COMPUTE dptcod ;
    urlstring = "&PATH_REF\"||TRIM(dptcod)||".htm";
    call define(_col_, 'URL', urlstring);
  ENDCOMP;
  COMPUTE patident ;
    urlstring = "&PATH_REFP.\patident.htm#patident" || put(num,3.0 -L));
    CALL DEFINE(_col_, 'URL', urlstring);
  ENDCOMP;
```

```
RUN;
ODS TAGSETS.xls_single close;
```

Generating Multiple Worksheets per Workbook

SAS 9.1 allows you to automatically generate multiple workbooks per worksheet using the ExcelXP tagset. This requires you to have at least Excel 2002. If you have an older version of Excel, then you can use the below example to create multiple worksheets per workbook. The workbook below has five worksheets named by the visit code. This example shows that the final result look like a "view". You have to insert the path into the HREF tags.

PROC TEMPLATE;

```

    DEFINE tagset tagsets.xls_multi;
    parent=tagsets.msoffice2k;
    DEFINE event initialize;
        set $orientation tagset_alias;
        trigger set_just_lookup;
        trigger set_nls_num;
    END;
    DEFINE event doc;
    START:
    put '<html xmlns:x="urn:schemas-microsoft-com:office:excel"' NL;
    FINISH:
    put "</html>" NL;
    END;
    DEFINE event doc_head;
    START:
    put "<head>" NL;
    put "<style> @page {mso-page-orientation:landscape} </style>" NL;
    put ' <meta name="Excel Workbook Frameset">';
    FINISH:
    do / if cmp($orientation,'landscape');
    put "<!--[if gte mso 9]><xml>" NL;
    put "<x:ExcelWorkbook>" NL;
    put " <x:ExcelWorksheets>" NL;
    put " <x:ExcelWorksheet>" NL;
    put "         <x:Name>SEL1</x:Name>" NL;
    put "         <x:WorksheetSource HRef='&PATH_XLS\SEL1.xls'/" NL;
    put "         </x:ExcelWorksheet>" NL;
    put " <x:ExcelWorksheet>" NL;
    put "         <x:Name>VINC</x:Name>" NL;
    put "         <x:WorksheetSource HRef='&PATH_XLS\VINC.xls'/" NL;
    put "         </x:ExcelWorksheet>" NL;
    put " <x:ExcelWorksheet>" NL;
    put "         <x:Name>VIS1</x:Name>" NL;
    put "         <x:WorksheetSource HRef='&PATH_XLS\VIS1.xls'/" NL;
    put "         </x:ExcelWorksheet>" NL;
    put " <x:ExcelWorksheet>" NL;
    put "         <x:Name>VIS2</x:Name>" NL;
    put "         <x:WorksheetSource HRef='&PATH_XLS\VIS2.xls'/" NL;
    put "         </x:ExcelWorksheet>" NL;
    put " <x:ExcelWorksheet>" NL;
    put "         <x:Name>VIS3</x:Name>" NL;
    put "         <x:WorksheetSource HRef='&PATH_XLS\VIS3.xls'/" NL;
    put "         </x:ExcelWorksheet>" NL;
    put " </x:ExcelWorksheets>" NL;
    put " <x:WindowHeight>8070</x:WindowHeight>" NL;
    put " <x:WindowWidth>10380</x:WindowWidth>" NL;
    put " <x:WindowTopX>480</x:WindowTopX>" NL;
    put " <x:WindowTopY>45</x:WindowTopY>" NL;
    put " <x:ActiveSheet>2</x:ActiveSheet>" NL;
    put "</x:ExcelWorkbook>" NL;
    put "</xml><![endif]-->" NL;
    done;
    put "</head>" NL;
    END;
    END;

```

RUN;

Editing the multiple worksheets per workbook.

The code below allows the creation of multi_sheet.xls file.

```
ODS LISTING CLOSE;
ODS MARKUP file="&PATH_XLS.\multi_sheet.xls" TAGSET=tagsets.xls_multi
alias='landscape' style=styles.sasweb;

data _null_;
    file print;
run;

ODS MARKUP close;
```

Creating Native Excel Files

SAS 9 does not have the ability to generated native Excel files. With the method below, there is no more need for xls files by visit. It is a more practical means for file transport.

```
/* Open Excel */
x "'C:\Program Files\Microsoft Office\Office12\excel.exe'";
DATA _NULL_;
    X=SLEEP(5);
RUN;

OPTIONS noxwait noxsync;
FILENAME cmds dde 'excel|system';
/* Send information to DDE */
DATA _NULL_;
    FILE cmds;
    PUT '[open("C:\Perso\PHUSE2007\OUTPUT\multi_sheet.xls")]';
    PUT '[SAVE.AS("C:\Perso\PHUSE2007\OUTPUT\final_result.xls",1)]';
    X=SLEEP(1);
    PUT '[close("false")]';
    PUT '[quit()]';
RUN;
QUIT;

/* delete the original file */
DATA _NULL_;
    rc=filename('fref',"C:\Perso\PHUSE2007\OUTPUT\multi_sheet.xls");
    rc=fdelete('fref');
    rc=filename('fref','');
RUN;
```

PhUSE 2007

Final result

To consult information concerning the patient, click on the patient's number to open the patients listing directly at the patient's data.

Use the option ANCHOR= to insert a tag into the HTML file for each patient.

```
PROC SORT DATA=phuse07.AE OUT=pat;
    BY patident visit;
RUN;

ODS LISTING CLOSE;
ODS HTML FILE="&PATH_REFP.\patient\patident.htm" ANCHOR="patident";

TITLE;
PROC REPORT DATA=pat NOWD SPLIT='@';
    BY patident;
RUN;

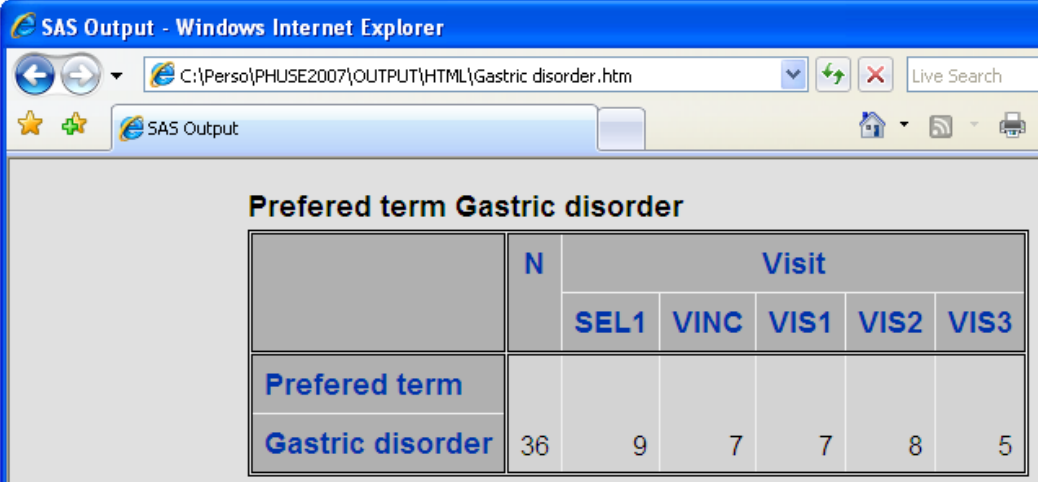
ODS HTML CLOSE;
ODS LISTING;
```

Country	patient	Preferred term	Start date	End date	Time	
FRANCE	0000000001	Gastric disorder	12/01/2006	13/01/2006	1	
FRANCE	0000000002	Bronchospasm	19/01/2006	21/01/2006	2	
FRANCE	0000000003	Erectil dysfunction	20/01/2006	23/01/2006	3	
FRANCE	0000000004	Skin lesion excision	01/02/2006	05/02/2006	4	
SAS Output - Windows Internet Explorer						
file:///C:/Perso/PHUSE2007/OUTPUT/HTML/patient/patident.htm#patident9						
SAS Output						
patient=0000000010						
Country	Visit	patient	Prefered term	Start date	End date	Time
FRANCE	SEL1	0000000010	Dizziness postural	08/01/2006	18/01/2006	10
patient=0000000011						
Country	Visit	patient	Prefered term	Start date	End date	Time
FRANCE	SEL1	0000000011	Gastric disorder	10/01/2006	21/01/2006	3
FRANCE	VINC	0000000011	Dizziness postural	08/01/2006	18/01/2006	10
FRANCE	VINC	0000000011	Hyperhidrosis	12/01/2006	31/01/2006	5

PhUSE 2007

To determine the number of a certain preferred term at each visit, click on the required preferred term to open the corresponding HTML file.

1	Country	patient	Prefered term	Start date	End date	Time
2	FRANCE	0000000001	Gastric disorder	12/01/2006	13/01/2006	1
3	FRANCE	0000000002	Bronchospasm	19/01/2006	21/01/2006	2
4	FRANCE					
5	FRANCE					
6	FRANCE					
7	FRANCE					
8	FRANCE					
9	FRANCE					
10	FRANCE					
11	FRANCE					
12	FRANCE					
13	FRANCE					
14	FRANCE					
15	FRANCE					
16	FRANCE					
17	FRANCE					
18	FRANCE					



The screenshot shows a SAS Output window titled 'SAS Output - Windows Internet Explorer'. The address bar shows the file path: C:\Perso\PHUSE2007\OUTPUT\HTML\Gastric disorder.htm. The main content area displays a table titled 'Preferred term Gastric disorder'.

	N	Visit				
		SEL1	VINC	VIS1	VIS2	VIS3
Preferred term						
Gastric disorder	36	9	7	7	8	5

The code below allows the HTML file concerning preferred term to be edited.

```
%MACRO gnr_hm;
/* Find all preferred term*/
PROC SORT DATA=phuse07.AE(keep=dptcod) out=dptc nodupkey;
    by dptcod;
RUN;
%LET nbdpt =0;
DATA _NULL_;
    SET dptc END=EOF;
    IF EOF THEN CALL SYMPUT("nbdpt",PUT(_n_,best. -L));
    CALL SYMPUT("dpt"||PUT(_n_,best. -L),trim(dptcod));
RUN;

%DO i=1 %TO &nbdpt;
    /* create one HTML file by preferred term */
    ODS LISTING CLOSE;
    ODS HTML FILE="%PATH_REF.\&dpt&i...htm";
    TITLE;
    PROC TABULATE DATA=phuse07.AE (WHERE=(TRIM(dptcod)="%&dpt&i"));
        CLASS visit /ORDER=DATA MISSING;
        CLASS dptcod / ORDER=DATA MISSING;
        TABLE dptcod, dptcod,N visit*N="";
    RUN;
    ODS HTML CLOSE;
    ODS LISTING;
%END;

%MEND gnr_hm;
%gnr_hm;
```

CONCLUSION

There are many ways of moving SAS output to Excel. This paper has described a method that can be used to produce both listings and patient summaries. The programming is kept to a minimum by preprocessing the data in SAS.

This paper also described how to create hypertext links in Excel using the Report procedure. The quantity of information you can put on the HTML file is not limited.

ACKNOWLEDGMENTS

Internet search engines for finding previous papers with code or similar issues.

RECOMMENDED READING

- SUGI :
 - SUGI 28 – Paper 12-28 – Generating custom Excel spreadsheets using ODS (Chevell Parker, SAS institute, Cary, NC)
 - SUGI 30 – Paper 136-30 – moving data and analytical results between SAS and Microsoft Office (Vincent DelGobbo, SAS institute, Cary, NC)
- SAS 9.1 MS OFFICE Integration
- Paper 85-30 – ODS MARKUP : The SAS reports you've always dreamed of (Eric A. Gebhart, SAS Institute, Cary, NC)

CONTACT INFORMATION

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

Dominique Pinet
Keyrus
69 rue Carnot
92300 Levallois-Perret
Work Phone: +33 (0)1 41 34 10 00
Email: Dominique.pinet@keyrus.com
Web: www.keyrus.com

Other brand and product names are trademarks of their respective companies.