PhUSE 2006

Paper TS02

SAS to PDF and JPEG
(or viewing outputs in Windows Explorer)

Xavier Passera, Detour Solutions Ltd., UK

ABSTRACT

a) Create a PDF document (with bookmarks) of all outputs found in directory.
Outputs are produced in different formats such as landscape, portrait, graph. This paper shows how to produce 1 pdf file with all the outputs and use bookmarks to describe what they are.

b) Convert your cgm files to jpeg and view your graphs in a more friendly way.
Reviewing cgm files can be a slow process on unix. Knowing that jpeg files can be viewed using thumbnails, I thought I'd convert the cgm files to jpeg and view them in Windows Explorer. This paper describes the SAS code that does this.

INTRODUCTION

An increasing number of people are working from home, more people need to review outputs away from the office. This has led to new requests to view study outputs in a form which is simple and quick. This paper shows 2 methods that can be used, the first by converting outputs to PDF and the second by converting CGM files to JPEG.
The first methods shows the technical aspects of converting outputs in PDF files, adding bookmarks to order and identify outputs.
The second method shows the technical aspect of converting CGM files (figures) to JPEG. Again the windows environment is used to view them.

CONVERTING OUTPUTS TO PDF

Initially we need to check the different types of outputs found in the directory (landscape, portrait, figures). If you have a tool to identify your outputs then this can be used to create bookmarks. For this presentation we'll assume we have a dataset that contains the output names and the associated title. The title will be used to create a bookmark.

A PDF file is created for every type of output found. There will be a landscape, portrait and figure file in PDF format.

OUTPUT TITLES USED AS BOOKMARKS

The study titles are stored in a permanent dataset and contains a key that is used to link with the matching filename on the server.

Dataset structure:
Type: $8 - Type of output. Eg. Efficacy, Safety, Exploratory
Output: $20 – Complete output name. Eg. Etdemo.out
SAP: $8 – Output short name. First letter indicates type of output (Table, Listing, Figure) followed by digits
Title: $400 – Output title as specified in SAP

Type	SAP	Title	Output
Efficacy	T0001	Summary of Primary Endpoint	ET0001.lis
Efficacy	L0001	Listing of Primary Endpoint	EL0001.lis

LIST ALL OUTPUTS FOUND IN DIRECTORY AND KEEP THE ONES REQUIRED

1) Get list of all outputs found in directory specified

```
%let folder = $HOME;

** Get outputs in directory specified **;
filename entree pipe "ls  &folder/*.*";

data server (label="list of outputs found");
  length output $100 ;
  infile entree;
  input output ;

  output = reverse( scan( reverse(output), 1, '/') );
run ;
```

2) Combine outputs found with outputs in dataset (ie. outputs with titles). Indeed the directory may contain other types of outputs so it's important that only the study outputs are kept.

```
** Sort Study outputs **;
proc sort data=sasuser.SAP (keep=type output sap title) out=SAP nodupkey  ;
  by output ;
run ;

data Study (label="List of outputs to import" drop=sap);
  attrib fileext  length=$5   label="Output extension type"
         outname  length=$20  label="Output name from SAP"
         outnum   length=8    label="Output number from SAP"
  ;
  merge server (in=A)
        SAP    (in=B);
  by output ;
  if A and B  ;

  fileext = scan(output, 2, '.') ;
  outnum  = input( substr(sap, 2, length(sap)-1 ), 8.) ;
  outname = put( substr(sap, 1, 1), $type.)  ;
run ;
```

3) Find the number of different types of outputs and create macro variables to setup the page size of each one

```
** Get list of different catgories and file types found **;
proc sort data=Study (keep=fileext type)
    out=difftype (label="list of different file types found") nodupkey ;
  by type fileext ;
run ;

data _null_;
  set difftype end=fin ;

  if fileext = "OUT" then do;
    ls = 93; ps=73; orient="Portrait";
  end;
  else do;
    ls = 132; ps=52; orient="Landscape";
  end;

  call symput('nls' || left(_n_), trim(ls) );
  call symput('nps' || left(_n_), trim(ps) );
  call symput('norient' || left(_n_), trim(orient) );
  call symput('ntype' || left(_n_), trim(type) );
  call symput('file'  || left(_n_), trim(fileext) );
```

```
  if fin then call symput('totfile', left(_n_) );
run ;
```

LOOP ON EACH TYPE OF OUTPUT (CATEGORY AND TYPE) AND CREATE A PDF FILE

```
%do zz = 1 %to &totfile;
  %put ************************************************;
  %put ** Looping on extensions : &&ntype&zz &&file&zz ;
  %put ************************************************;
```

Graph files

```
  %if &&file&zz. =cgm %then %do;
```

Setup ODS environment. This includes pagesize, linesize and orientation

```
    title;
    footnote;
    options topmargin=0.25 leftmargin=0.25 rightmargin=0.25;
    options ls=&&nls&zz ps=&&nps&zz orientation=&&norient&zz papersize="ISO
     A4" ovp;
```

Create macro variables with all the cgm output description (ie. title). This will be used to describe the output.

```
    proc sort data=Study out=cgmfile (label="List of CGM outputs") nodupkey;
      by outname outnum output ;
      where index(output, 'cgm') > 0 and index(type, "&&ntype&zz.") > 0 ;
    run ;

    %let totfile=0;
    data _null_;
      length fileext  $10. ;
      set cgmfile end=fin  ;

      call symput('outp' || left(_n_), trim(left(output)) );
      call symput('outn' || left(_n_), tranwrd( trim( left(outnum)), '.', '_')
     );
      if fin then call symput('totfile', left(_n_) );
    run ;

    proc greplay igout=work.gseg nofs;
      delete _all_ ;
    run;
    quit;
```

**** Read all cgm outputs found **;**

```
    %do i = 1 %to &totfile;

      filename gsasfile "&folder./&&outp&i." ;
      goptions reset=global gaccess=gsasfile gsfmode=replace
               noprompt device=cgmcrt;

      goptions reset=all gunit=pct border
               cback=white colors=(black blue green red)
               ftext='courier new' htitle=3 htext=3
               hsize=4in vsize=3.8in;
```

Graphic Outputs are stored in saswork catalog,

```
      proc gimport fileref=gsasfile
               filetype=cgm
               format=binary gout=work.gseg;
      run;
```

Add title to graph description

```
     proc catalog cat=work.gseg  entrytype=grseg;
       change cgm = F_&&outn&i ;
        modify F_&&outn&i (description="&&outp&i.");
      run;
    quit ;
  %end;
```

Create Graph PDF file

```
  ods listing close ;
  ods pdf file="$HOME/&&ntype&zz.._&&file&zz...pdf" style=PrinterSmaller ;

  ** Create cgm outputs in PDF **;
  %do I = 1 %to &totfile;
    goptions reset=goptions border cback=white colors=(black) ;

    ods proclabel = "Figure &&outn&I" ; ← Bookmark
    proc greplay igout=work.gseg nofs;
      replay F_&&outn&i ;
    run;
    quit;

  %end ;

  ods pdf close;
  ods listing;

%end;
```

ASCII files

```
%else %do;
```

Read every ASCII output and store it in 1 dataset. The dataset contains information about the output name, page numbers and the information read.

```
  filename attdir ("&folder./*.&&file&zz.") ;

  Data o2_data1 (label="ASCII Outputs" keep=output text page nline) ;
    retain page 0 orient 93;
    attrib text     length=$400    label='Text Read'
           myfile   length=$200    label='Full Filename'
           output   length=$100    label='File name'
           temp     length=$100    label='File name without extension'
           page     length=8       label='Page number '
    ;
    infile attdir dsd missover lrecl=256 length=linelen filename=myfile;
    input firstvar $1-10 @ ;
    D1_vlen=linelen;
    input @1 text $varying256. D1_vlen ;

    output = reverse( scan( reverse(myfile), 1, '/') );

    temp = trim( scan (output, 1, '.'));
    if index(text, trim(temp) || ' ') > 0 then page = page + 1;
    nline = _n_ ;
  run ;

  proc sort data=o2_data1  ;
    by output page nline;
  run ;
```

Keep only outputs found in tracking. This is in case more outputs are found in the directory which are not part of the study reports, eg, QC outputs.

```
  data o2_data2 (label="ASCII outputs found in tracking");
```

```
  merge Study (in=A)
        o2_data1 (in=B) ;
  by output;
  if A and B;

  if index(type, "&&ntype&zz.") = 0 then delete ;
run ;

proc sort data=o2_data2  ;
  by outname outnum output page nline;
run ;
```

Now that we have read all the outputs and kept the ones we need, macro variables are created of the remaining output names.

```
proc sort data=o2_data2 out=o2_data3 (label="List of ASCII outputs")
    nodupkey;
  by outname outnum output ;
run ;


%let totfile=0;
data _null_;
  length fileext  $10. ;
  set o2_data3 end=fin  ;

  call symput('outp' || left(_n_), trim(left(output)) );
  call symput('outl' || left(_n_), trim(outname) || ' ' ||
trim(left(outnum)) );
  if fin then call symput('totfile', left(_n_) );
run ;
```

Setup ODS environment and PDF file name.

```
options topmargin=0.01 nodate nonumber papersize="ISO A4" ovp ;

proc template;
  define style PrinterSmaller;
parent=styles.Printer ;
replace fonts /
   'TitleFont2'= ("SAS Monospace, Courier",9pt)
   'TitleFont'= ("SAS Monospace, Courier",9pt)
   'StrongFont'= ("SAS Monospace, Courier",9pt)
   'EmphasisFont'= ("SAS Monospace, Courier",9pt)
   'FixedEmphasisFont'= ("SAS Monospace, Courier",9pt)
   'FixedStrongFont'= ("SAS Monospace, Courier",9pt)
   'FixedHeadingFont'= ("SAS Monospace, Courier",9pt)
   'BatchFixedFont'=("SAS Monospace, Courier New",&linesize.pt)
   'FixedFont'= ("SAS Monospace, Courier",9pt)
   'headingEmphasisFont'= ("SAS Monospace, Courier",9pt)
   'headingFont' = ("SAS Monospace, Courier",9pt)
   'docFont' = ("SAS Monospace, Courier",9pt)
;
replace body from document /
  leftmargin   = -1in
  rightmargin  = -1in
  topmargin    = -1in
  bottommargin = -1in
;
style table /
  cellspacing=0
  framespacing=0
;
  end;
run ;
```

```
    ods escapechar="^";
    ods listing close ;
    options ls=&nls ps=&nps orientation=&norient ;
    ods pdf file="$HOME/&&ntype&zz.._&&file&zz...pdf" style=PrinterSmaller ;
```

loop on outputs found and print to PDF file

```
    %do I = 1 %to &totfile;

      title "&&outp&I" ;

      %put ** Printing &I - &&outp&I to PDF **;
      ods proclabel = "&&outl&I" ; ← PDF bookmark

      data _null_;
         set o2_data2 ;
         by outname outnum output page nline;
         where output = "&&outp&i." ;
         file print notitles ;

         ** force page break **;
         if first.page then put _page_ ;
         text = trim(text) ;

            put text 1-&nls;
      run ;
      run ;
    %end ;
```

Close PDF file

```
    ods pdf close ;
    ods listing  ;
  %end; /** ascii outputs **/

%end ;  /** zz loop on file extensions found **/
```

CGM TO JPEG AND WINDOWS

The concept is similar to the GRAPH section previously described. Initially we need to obtain the list of all CGM files in a directory. However the settings for importing the graphs has to be correct, otherwise the convertion to JPEG doesn't produce a decent looking graph.

The CGM files are read into a temporary SAS catalog and then exported to a JPEG file in the same directory. The CGM files are not modified in anyway.

IMPORT CGM FILES

… (see code in previous section)

…

```
      filename gsasfile "&folder./&&outp&i." ;
      goptions reset=global gaccess=gsasfile gsfmode=replace
               noprompt device=cgmcrt;
```

Trial and error settings that seem to import CGM files the way I wanted.

```
      goptions reset=all gunit=pct border
               cback=white colors=(black blue green red)
               ftext='courier new' htitle=3 htext=3
               hsize=4in vsize=3.8in;
```

Read CGM output and store it in temporary catalog

```
      proc gimport fileref=gsasfile
               filetype=cgm
               format=binary
               gout=work.gseg;
      run;
```

Rename CGM output in catalog. Add description to graph using method described in previous section

```
      proc catalog cat=work.gseg  entrytype=grseg;
       change cgm = F_&&outn&i ;
       modify F_&&outn&i (description="&&outp&i.");
       run;
      quit ;
    %end;
```

SAVE GRAPHS AS JPEG FILES

Loop on every output found in the saswork catalog

```
    %do i = 1 %to &totfile;
      filename OUTGRAPH "&folder./&&outf&i." ;
```

Options that produce a JPEG file with the correct settings

```
      goptions
        device  =jpeg
        gsfname =OUTGRAPH
        gsflen  =80
        gsfmode =replace
        hsize   =8.4in
        vsize   =6.3in
      ;

      proc greplay igout=WORK.GSEG nofs;
        replay F_&&outn&i ;
      quit;

      Filename OutGraph clear;
    %end ;
```

SEND THE JPEG TO WINDOWS ENVIRONMENT

Different methods can be used to send the JPEG files, for example using FTP. Once the files are in the Windows environment, you can use the preview option to view the files, zoom in/zoom out and manipulate the files easily.

This is particularly useful when you're want a general view of the graphs or if you're looking for examples of graphs.

CONCLUSION

This paper has shown how to convert outputs to PDF and group them by type, allowing the client to use bookmarks to review outputs quickly. Using bookmarks means memorizing outputs name is no longer necessary. The advantage of using PDF is that the files can be viewed quickly across platforms.

We've also seen how to convert outputs from CGM to JPEG format and (pre)view the files on the PC/Windows platform. This can save time when reviewing or QCing many figures

An advantage of using PC/windows is that it's easy to share outputs. Regarding this paper, if the PDF files and JPEG files are stored on a network drive then they can be reused in the future to identify outputs and hence programs, thereby (potentially) reducing programming time.

ACKNOWLEDGMENTS

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

CONTACT INFORMATION

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

Xavier Passera
Detour Solutions Ltd
Welwyn Garden City, UK
Mobile +44 (0)7930 472 690
Email: Xavier.Passera@detoursolutions.co.uk
Web: www.detoursolutions.co.uk