

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 65, 3rd Quarter 2021

This newsletter issue only contains articles from me, but I am hoping you will find generating bookmarked PDF files and some more annoying programming styles interesting. As usual I have publicised some SAS-related events for the next few months, and I have also added some more SAS formats, options and functions that you may or may not have come across before, but you might find interesting, and even useful.

If you would like to contribute an article, to re-visit and improve an existing article, or just discuss the possibility of doing so, please feel free to send an email to me at views-uk@hollandnumerics.org.uk.

Philip R Holland (Newsletter Editor)

Membership of VIEWS is **free** and you can register for notification of all future VIEWS News issues by registering at blog.hollandnumerics.org.uk, and then emailing views-uk@hollandnumerics.org.uk, or using the blog site menu, to request access to the VIEWS UK forum. Members will receive an email notification whenever a new issue of the free quarterly VIEWS News is published on the web site, please inform us when you change your email address, otherwise your registration may be cancelled.

Did You Know?

If you have a useful hint or tip, please send it to the Editor, and share it with the VIEWS membership.

Generating bookmarked PDF files from plain text files using SAS

[SAS 9] Many pharmaceutical companies routinely write reports into plain text files, which are then converted into other formats such as PDF. SAS software includes a number of facilities to generate PDF files that use ODS (Output Delivery System). This article will describe two of the methods, but only the second generates PDF bookmarks that are accepted by the FDA, and the first method, while able to process the sample files, is extremely slow when processing large files, which can typically hold over 10,000 pages.

The sample files contain an output from SASHELP.CLASS, which has been paged by the **sex** variable. Note that I have deliberately duplicated the page where **sex='M'** to put 3 pages in **text_listings1**, so that you would be able to see that the bookmarks only reference the first page of each subgroup.

Protocol: 123456

Population: All Subjects

Listing 99.1
Test Listings 1

Page 1 of 3
Data as of 27APR2021

sex: F

Name	Age (years)	Weight (lb)	Height (in)
Alice	13	84	56.5
Barbara	13	98	65.3
Carol	14	102.5	62.8
Jane	12	84.5	59.8
Janet	15	112.5	62.5
Joyce	11	50.5	51.3
Judy	14	90	64.3
Louise	12	77	56.3
Mary	15	112	66.5

Method 1: PROC DOCUMENT

The following program will look for new pages, which begin with '0C'X, and extract the output number and title. Line 3 of each page contains the output type and number, and line 4 of each page contains the output title. The REPLAY statements for each page of the final PDF are stored in a SOURCE entry of a catalog:

```
%MACRO import_text(
    file=
    ,orient=landscape);
    FILENAME source CATALOG 'work.import';

    DATA _NULL_;
        LENGTH line $120;
        RETAIN i prev_n 0;
        INFILE &file. TRUNCOVER END = eof;
        INPUT line $CHAR120.;
        FILE source(import.source);
        IF _N_ = 3 THEN
            CALL SYMPUT('line3', STRIP(line));
        ELSE IF _N_ = 4 THEN
            CALL SYMPUT('line4', STRIP(line));
        ELSE IF SUBSTR(line, 1, 1) = '0C'X THEN DO;
            ** If first character is page break,
            ** then generate replay **;
            i + 1;
            IF i = 1 THEN DO;
                PUT 'replay textfile'
                    ' (WHERE = (_obs_ LT ' _N_ '));';
            END;
        ELSE DO;
            PUT 'replay textfile'
                ' (WHERE = (' prev_n ' LE _obs_ LT '
                    _N_ '));';
        END;
        prev_n = _N_;
    END;
    IF eof THEN DO;
        PUT 'replay textfile'
            ' (WHERE = (' prev_n ' LE _obs_));';
    END;
RUN;
```

The SOURCE entry from the catalog containing the generated REPLAY code from each of the input text files is then included in the PROC DOCUMENT step:

```
OPTIONS ORIENTATION = &orient. NODATE;
TITLE " ";
```

```

PROC DOCUMENT NAME = import(WRITE);
  IMPORT textfile = &file. TO ^;
  OBPAGE \textfile / AFTER;
  SETLABEL ^ "&line3.: &line4.";
  %INCLUDE source(import.source) / SOURCE2;
RUN;
QUIT;
%MEND import_text;

```

The macro is then called for each input text file to write the output to a PDF file with the relevant bookmarks:

```

ODS _ALL_ CLOSE;

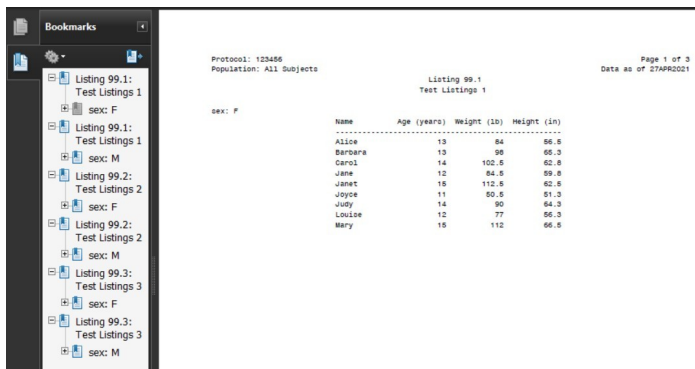
ODS PDF FILE = "./output/osi_listings.pdf"
  PDFTOC = 1 CONTENTS = YES;

%import_text(file="./output/test_listings1.txt")
%import_text(file="./output/test_listings2.txt")
%import_text(file="./output/test_listings3.txt")

ODS PDF CLOSE;

```

Below is the final PDF, with individual bookmarks included for each input file and group, which may be just what you were looking for:



However, PROC DOCUMENT is processing every page of the PDF file individually, so very large PDF files (>10,000 pages) can take a while to run!!

Method 2: FILE PRINT and GhostScript

This method uses the techniques for generating bookmarks for GhostScript described in a 2010 PhUSE paper by Stephen Griffiths in <http://www.lexjansen.com/phuse/2010/ts/TS04.pdf>. This method creates bookmarked PDF files accepted by the FDA!

I like to set the scene in complex programs by defining all of the changeable information at the beginning:

```

* Final Location for Listings and PDF file *;
%LET filelist = ./output;
%LET filepdf = ./output;

%LET offset = 1;

LIBNAME listings "&filepdf.";

* Name of SAS dataset with Bookmarks *;
%LET bookmark_ds = bookmark_listings;

* Name of PS file *;
%LET bm_ds = osi_bookmarks;

```

Having set the scene we will create an empty bookmarks data set, which will be used as a template, and then bookmark records will be appended for each text file, and for each group within:

```

DATA _bookmarks_;
  LENGTH filen $30;
  group = 'temp ' ;
  lpage = 0;
  ltotpage = 0;
  opage = 0;
  ototpage = 0;
  filen = '__temp.xxx';
  OUTPUT;
RUN;

```

The %outpage macro reads in the input text files, extracts paging info and combines the files into a single PDF file. This is similar to the %import_text macro, but is not generating REPLAY statements:

```

%MACRO outpage(
  dsout = /* bookmark list */
  ,fname = /* name of input text file */
  ,ftype = /* file type */
  ,title = /* file title */
  ,orient = landscape /* Orientation */
  ,prefix = Sex /* Group text */
);
  OPTIONS ORIENTATION = &orient.
    NODATE NONUMBER;
  TITLE " ";

  DATA &fname. (KEEP = order group lpage
    ltotpage);

    LENGTH line2 $150
      group $6
      ;

    * Page number *;
    RETAIN lpage 1 ltotpage 1;
    INFILE "&filelist./&fname..&ftype."
      TRUNCOVER END = eof;

    INPUT;
    * variable to retain the ordering *;
    order = _N_;
    FILE PRINT;
    * at each page break insert a new page *;
    SELECT;
      WHEN (SUBSTR(_INFILE_, 1, 1) EQ '0C'X) DO;
        line2 = SUBSTR(_INFILE_, 2);
        PUT _PAGE_ @&offset. line2;
        lpage + 1;
      END;
      WHEN (_INFILE_ EQ '')
        PUT @&offset. _INFILE_;
      OTHERWISE PUT @&offset. _INFILE_;
    END;
    * Total pages *;
    IF _N_ = 1 THEN DO;
      ltotpage =
        INPUT(SCAN(_INFILE_, -1), BEST8.);
      CALL SYMPUT('lpages',
        PUT(ltotpage, ??BEST.));
    END;
    IF INDEX(UPCASE(STRIIP(_INFILE_)),
      UPCASE("&prefix.")) EQ 1 THEN DO;
      group = STRIP(SCAN(STRIIP(_INFILE_),
        2, ':'));

      OUTPUT;
    END;
  RUN;

```

```

* Keep the first page for each group *;
PROC SORT DATA = &fname. OUT = &fname.;
  BY group order;
RUN;

DATA &fname. (DROP = order);
  SET &fname.;
  BY group order;
  IF FIRST.group;
  LENGTH filen $30 ltitle $200;
  filen = "&fname.";
  ltitle = "&title.";
RUN;

* Add the file to overall bookmarks dataset *;
* Determine the total number of pages in the
  files that have been bookmarked so far *;
PROC SORT DATA = &dsout. OUT = _maxp;
  BY DESCENDING ototpage;
RUN;

%LET totp=;

DATA _NULL_;
  SET _maxp;
  IF _N_ = 1
    THEN CALL SYMPUT('totp', ototpage);
RUN;

DATA &fname.;
  SET &fname.;
  opage = lpage + &totp.;
  ototpage = ltotpage + &totp.;
RUN;

DATA &dsout.
  (WHERE = (group NOT IN ('temp' ' ')));
  SET &dsout. &fname.;
  * Update total overall pages *;
  ototpage = &lpages. + &totp.;
RUN;
%MEND outpage;

```

The %create_bmfile macro creates the *.PS bookmarks file from the list of pages in the bookmark data set:

```

%MACRO create_bmfile(
  dsin = /* bookmarks data set */
  ,psout = /* PS file to hold bookmarks */
);
PROC SORT DATA = &dsin.
  OUT = &dsin._listings
  (KEEP = ltitle opage ototpage filen)
  NODUPKEY;
  BY ltitle;
RUN;

PROC SORT DATA = &dsin.
  OUT = &dsin._groups (KEEP = group)
  NODUPKEY;
  BY group;
RUN;

PROC SQL;
  CREATE TABLE &dsin._template AS
  SELECT *
  FROM   &dsin._groups
        ,&dsin._listings
  ORDER BY
    ltitle
    ,group
  ;
QUIT;

```

```

DATA &dsin._template (DROP = opage ototpage);
  SET &dsin._template;
  order = opage;
  nodatapage = ototpage + 1;
RUN;

* Merge template and input dataset *;
PROC SORT DATA = &dsin. out=&dsin.1;
  BY ltitle group;
RUN;

DATA &dsin.2;
  MERGE &dsin._template (IN = a)
        &dsin.1 (IN = b)
  ;
  BY ltitle group;
  * groups not in files *;
  IF a AND NOT b THEN DO;
    opage = nodatapage;
  END;
RUN;

* Bookmarks need to be ordered by group with
  the titles as sub-bookmarks *;
PROC SORT DATA = &dsin.2 OUT = &dsin.3;
  BY group order opage;
RUN;

* Count the number of files per group. This is
  used to specify the number of subordinate
  bookmarks needed for each group *;
PROC FREQ DATA = &dsin.3 NOPRINT;
  TABLE group / OUT = &dsin._group_sub
    (DROP = percent
     RENAME = (count = sub_bm));
RUN;

DATA &dsin.4;
  MERGE &dsin.3
        &dsin._group_sub
  ;
  BY group;
RUN;

DATA &dsin.4 (KEEP = group ltitle order opage
  sub_bm filen)
  listings.&bookmark_ds
  (KEEP = group ltitle filen opage
  sub_bm filen)
  ;
  SET &dsin.4;
  * Make the number negative so that the
    bookmark is closed *;
  sub_bm = sub_bm * -1;
RUN;

DATA _NULL_;
  FILE "&psout." LS = 1000;
  SET &dsin.4;
  BY group order;
  IF FIRST.group THEN DO;
    PUT "[/Count " sub_bm "/Title (" group
      " ) /Page " opage " /OUT pdfmark";
  END;
  IF FIRST.order THEN DO;
    PUT "[/Title (" ltitle " ) /Page "
      opage " /OUT pdfmark";
  END;
RUN;
%MEND create_bmfile;

```

Because the text files may not be read and then displayed exactly the same by default in the PDF files, a user-specified style template **groupfile** can be used:

```
PROC TEMPLATE;
  DEFINE STYLE groupfile;
    parent = styles.Printer;
    STYLE fonts FROM fonts /
      "BatchFixedFont" =
        ("Courier New, Courier", 9.5PT)
    "docFont" =
      ("Courier New, Courier", 9.5PT)
    "FixedFont" =
      ("Courier New, Courier", 9.5PT)
    ;
  END;
RUN;

ODS PDF FILE = "&filepdf./osi_listings.pdf"
  STYLE = groupfile;
ODS PDF NOBOOKMARKGEN;

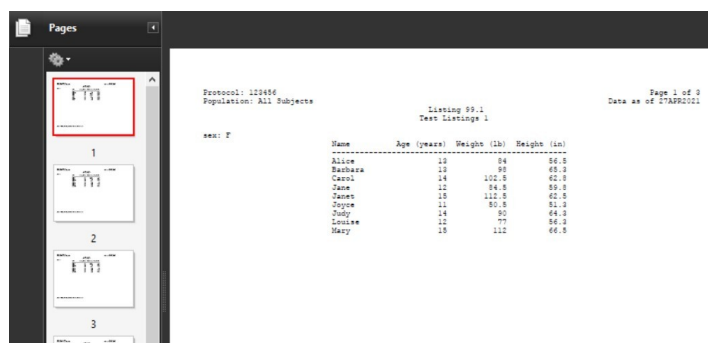
%outpage(
  dset = &bm_ds.
  ,fname = test_listings1
  ,ftype = txt
  ,title = %nrstr(Listing 99.1)
  ,prefix = Sex
);
%outpage(
  dset = &bm_ds.
  ,fname = test_listings2
  ,ftype = txt
  ,title = %nrstr(Listing 99.2)
  ,prefix = Sex
);
%outpage(
  dset = &bm_ds.
  ,fname = test_listings3
  ,ftype = txt
  ,title = %nrstr(Listing 99.3)
  ,prefix = Sex
);
```

Create a page in the final PDF file to cater for missing pages in the input data files. Note that this is not referenced in this example:

```
DATA _NULL_;
  FILE PRINT;
  PUT _PAGE_ @@offset.
    'There is no data available.';
RUN;

ODS PDF CLOSE;
```

This PDF file is created without bookmarks from the program above:



Create the bookmark PS file:

```
DATA &bm_ds.;
  SET _bookmarks_;
RUN;

ODS _ALL_ CLOSE;
```

Now generate the PS file using the **%create_bmfile** macro:

```
%create_bmfile(
  dsin = &bm_ds.
  ,psout = &filepdf./&bm_ds..ps);
```

The PS file generated from the sample text files looks like this:

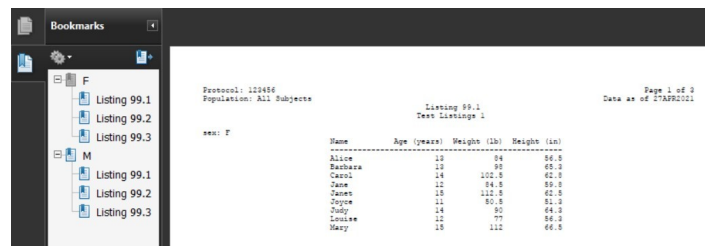
```
[/Count -3 /Title (F ) /Page 1 /OUT pdfmark
[/Title (Listing 99.1 ) /Page 1 /OUT pdfmark
[/Title (Listing 99.2 ) /Page 4 /OUT pdfmark
[/Title (Listing 99.3 ) /Page 6 /OUT pdfmark
[/Count -3 /Title (M ) /Page 2 /OUT pdfmark
[/Title (Listing 99.1 ) /Page 2 /OUT pdfmark
[/Title (Listing 99.2 ) /Page 5 /OUT pdfmark
[/Title (Listing 99.3 ) /Page 7 /OUT pdfmark
```

The following GhostScript call can be used to create the Final PDF file, using the PS file and the original PDF file. This example is using the GhostScript command line program (**gswin64c**) on Windows 64-bit. This post-processing could also be carried out using the corresponding command line programs for Windows 32-bit (**gswin32c**) and Linux (**gs**) by just replacing **gswin64c** below. Note that some of the filenames may require folder locations to be added when you run this code on your own SAS platform:

```
FILENAME cmd PIPE 'gswin64c -dBATC -dNOPAUSE
  -sDEVICE=pdfwrite
  -sOutputFile=osi_listings_bm.pdf
  osi_listings.pdf osi_bookmarks.ps';

DATA _NULL_;
  INFILE cmd;
  INPUT;
  PUT _INFILE_;
RUN;
```

This PDF file is created with bookmarks from the program above:



Note that the bookmarks are now nested by subgroup, and then by output, as required by the FDA. The code even introduces a “no data” page, which would be added to the bookmarks if needed.

Philip R Holland (Newsletter Editor)

Ask The Consultant

This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

Annoying SAS Programming Styles: Part 6

Q: I find %IF .. %THEN macro statements very difficult to use and understand. Can they be re-written so they are easier to read and debug?

A:[SAS] For example, the following SAS code needs to be made more flexible in a macro, so that different pieces of the code can be switched on/off with **where=** and **debug=** macro parameters:

```
DATA newclass;
  SET sashelp.class (WHERE=(sex="M"));
  PUT 'DEBUG: ' _ALL_;
RUN;
```

- **where=** will be used to specify the value in the **sex** filter, but will also remove the filter if it is missing.
- **debug=** will be used to switch on/off the **PUT** statement, which, by default, is switched off.

Firstly we will implement the macro using %IF and %THEN:

```
%MACRO _filter1(where=, debug=N);
  DATA newclass;
    SET sashelp.class
    %IF &where. NE %THEN (WHERE=(sex="&where.));;
    %IF &debug. EQ Y %THEN PUT 'DEBUG: ' _ALL_;;
  RUN;
%MEND _filter1;
```

Note the potentially confusing and risky “;;” at the end of each statement, one to end the SAS statement and the other to end the %IF statement. Omitting a “;;” could generate errors or incorrect logic.

Secondly we will implement the same macro using %IF, %THEN and %DO, which visibly separates the SAS Data Step code from the macro code, avoiding the potentially risky syntax:

```
%MACRO _filter2(where=, debug=N);
  DATA newclass;
    SET sashelp.class
    %IF &where. NE %THEN %DO;
      (WHERE = (sex = "&where."))
    %END;
    ;
    %IF &debug. EQ Y %THEN %DO;
      PUT 'DEBUG: ' _ALL_;
    %END;
  RUN;
%MEND _filter2;
```

Is %_filter1 or %_filter2 easier to read and debug?

Philip R Holland (Newsletter Editor)

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions. Please read the SAS documentation to find out more about them.

[SAS] The SAS format **Fw.d** is actually an alias for **w.d**, and will display numeric values after rounding to fit in the width available. Warning: if PROC DATASETS is used to update a format with the MODIFY statement, it will automatically change **w.d** to **Fw.d**, but PROC COMPARE may not treat **Fw.d** the same as **w.d**.

[SAS 9.1.3] The height and width of graphic output can be specified using the **HEIGHT=** and **WIDTH=** options on the **ODS GRAPHICS** statement. Units of **CM**, **MM**, **IN**, **PCT**, **%** or **PX** can be used to specify the height and width.

[SAS] The SAS macro function %SCAN() extracts text just like the SCAN() Data Step function, except that, as macro variables are always strings, the text should not be quoted, unless it includes special characters that could be misinterpreted when parsing the macro call, like commas, quotes, etc., when it should be quoted using a macro function, e.g. %STR().

In Brief

- Back issues of *VIEWS News* are available from the VIEWS UK forum at **blog.hollandnumerics.org.uk**. Please register for free membership on the blog, and then request free access to the VIEWS UK forum.
- The SAS Professional Forum group on LinkedIn, which currently has over 36,600 members. Everyone is welcome to request to join, but discussions here are restricted to SAS-related topics: **www.linkedin.com/groups?gid=70702**.
- The SAS Opportunities group on LinkedIn, which currently has over 14,000 members. Everyone is welcome to request to join, but discussions here are also restricted to SAS-related topics: **www.linkedin.com/groups?gid=805217**.
- In the UK and Ireland the SUGUKI group had been holding regular meetings in London, and also occasional meetings in Bristol, Dublin, Edinburgh, Leeds and Manchester. Unfortunately lockdown has necessitated running webinars instead. For more details and to register for future events please go to **meetup.com/SUGUKI**.
- Fancy joining the SAS EMEA Analytics Explorers, then please send your **country**, **name**, **email address** and **company** with your permission to pass these details on to SAS to **views-uk@hollandnumerics.org.uk**. Other regions include Asia Pacific, Canada, Latin America and US, but to join these communities based on where you live, you will now need to provide this information to a current member of these communities, instead of me.

- WPS Community Edition provides users with free access to WPS Analytics software including a SAS language compiler with coding and workflow tools on unlimited data volumes. A Community Edition licence key will be issued that will activate the software for up to six months. At the end of this term, users may be invited to renew their licence and to upgrade to the latest available version of the software. No technical support service is offered with the Community Edition. Download from www.worldprogramming.com/try-or-buy/editions. Note that the software is made available without warranty or indemnity, and is not made available to US residents.

Diary

Are you organising an event that would be of interest to the VIEWS readership? Let us know as we are interested in all non-profit making events related to SAS.

October 2021

29 **PharmaSUG North Carolina**, Chapel Hill, NC, USA

For more details about times and speakers at this Single Day Event please go to the PharmaSUG web site at www.pharmasug.org.

November 2021

15-19 **Virtual PhUSE EU Connect**

For more details about times and speakers please go to the PhUSE web site at www.phuse-events.org.

April 2022

10-13 **SAS Global Forum**, San Diego, CA, USA

For more details about times and speakers please go to the SAS Global Forum web site at www.sasglobalforum.org.

Contacts

SAS UK

Tel: +44-(0)1628-486933

Web: www.sas.com/uk

VIEWS

Web Site

Web: blog.hollandnumerics.org.uk

Newsletter Editor

Email:
views-uk@hollandnumerics.org.uk

***VIEWS News** is published quarterly for the VIEWS International SAS Programmer Community by Holland Numerics Limited.*