

%XTAB2HTM: macro using ODS to convert proc FREQ cross-tabulation to HTML with a Table of Contents and back buttons

Ronald Fehd, Centers for Disease Control and Prevention, Atlanta GA

ABSTRACT

SAS® V8 software Output Delivery System (ODS) provides programmers with the ability to write HyperText Markup Language (HTML) files. A Table of Contents may be added to these files by adding hypertext links and destinations to SAS® data sets before they are directed to output. This paper discusses the syntax of HTML anchors and links and illustrates a method of writing an index and forward- and backward-linked files.

The intended audiences are intermediate SAS® and macro programmers.

Key words:

list processing, proc FREQ, cross-tabulation table, HTML, anchor, link, href, name, goto, label.

INTRODUCTION

The task for which I wrote this macro was to supply a print-out of text responses associated with a particular question on a survey data collection form. My customer needed to occasionally check the hand-written marginalia to certain questions. Since 'occasionally' was the key in this request, I considered whether I might be able to save some paper and maybe half a tree by writing the output to an HTML file. In my first effort I delivered a single file of 2E7 bytes, which crashed the customer's PC! Back to the drawing board!

But first a plea to SAS-L and a search of SUGI Proceedings to find Ray Pass's excellent paper on writing HTML anchors. No sense in reinventing the wheel! A search of the internet yielded the World Wide Web Consortium (W3C) recommendations for HTML 4.

After examining my first effort, I considered how to break the large file into smaller pieces. What emerged is a utility that produces many small HTML files which can be quickly loaded and are cross-referenced both forward and backward, through the index and Table of Contents. Back buttons are provided to return to top of file and to the index.

Writing HTML anchors with links, sources and destinations

An anchor contains a link to a file. This is called a Hypertext Reference, (href). After clicking on an anchor, the browser opens the file and is positioned at the top of file.

The basic form of an anchor with a link embedded is as follows:
chapter two.

An anchor contains four elements:

1. Open: <A ...>
2. Link: href="filename.extension"
Note double quotes enclose the file reference.
3. Text: describing the link
4. Closure:
Note that the anchor and its closure can be in lower case:
<a ...> ...

An anchor link has two additional longer forms: source and destination, which are text strings appended to the file reference after a pound sign (#). The anchor source acts like a goto; the

anchor destination contains the same text string as the source and acts as the label for the goto.

1. source: href="filename.extension#location-in-file"
 2. destination: name="location-in-file"
- where the string "location-in-file" is the label.

These are the three HTML anchors used in this application.

Problem discussion: structure of FREQ cross-tabulation

A two-way cross-tabulation table from proc FREQ looks like this:

proc FREQ; tables VarA * VarB;

VarA	VarB	Count	Percent
A1	B1	10	10.0
	B2	20	20.0
	B3	30	30.0
A2	B1	10	10.0
	B2	10	10.0
	B3	20	20.0

As a general example it is clear that there are fewer levels of VarA than there are levels of VarB. We can use this understanding of the data set later when constructing the HTML index of such a table.

For the large table that I originally began working with, I saw that splitting the table on the values of VarA would significantly reduce the size of any one html file. VarA had twenty levels; I could reduce the size of the largest file by an order of magnitude by creating subsets of the data on VarA.

Task Overview

First I needed to write the main HTML directory for the cross-tabulation; this file is named index.html. This I could do by adding anchor links to my proc FREQ output data set of the first cross-tabulation variable, VarA.

For each level of VarA, I needed a pair of files: one cross-tabulation and one listing. The names of the pair of files would be linked in the index.html. I added links to the cross-tabulation to the VarA value column and links to the listing to the VarA Count column. This allowed the user to choose either the summary provided by the cross-tabulation, or to go directly to the detail listing, which had the row identifiers. The cross-tabulation needed to contain anchor links to the listing. And both needed back buttons to return to top-of-file and index, which were inserted in footnotes.

Macro overview

This macro consists of two parts: the list processor XTAB2HTM which writes the ODS file INDEX.HTML and its subroutine HTMWRTIE which writes all the pairs of cross-tabs and details that INDEX.HTML links to.

The macro XTAB2HTM does the following:

- * sorts and creates a copy of the input data set: DETAILS
- * proc FREQ of DETAILS.VarA, to data set INDEX
- * adds HTML links to INDEX
- * writes ODS file INDEX.HTML

* makes subsets of DETAILS for sub-routine HTMWWRITE

The subroutine HTMWWRITE does the following:

```
* proc FREQ of DETAILS.VarB to data set TBL_CONTENTS
* adds links to TBL_CONTENTS
* adds back buttons in footnotes
* writes ODS file *XTAB.HTML
* adds destination labels – table of contents – to DETAILS data
set
* writes ODS file *LIST.HTML
```

CONCLUSION

Writing HTML files containing anchors with links and destinations is a straight-forward process, once the syntax is understood.

REFERENCES

W3C: World Wide Web Consortium: <http://www.w3.org/>
HTML V4.0, Links:
<http://www.w3.org/TR/REC-html40/intro/intro.html>
<http://www.w3.org/TR/REC-html40/struct/links.html#h-12.1>

Examples of HTML

These examples use the List Item construction – `<li> ... </li>` to simplify the presentation – each line begins with a bullet.

Use text editor to create and save the three demo files in `c:\temp`.

Demo purpose: to illustrate hypertext commands to link to another file, and to link to a destination within a file

1. Open your web browser
2. Open <file:///c:/temp/testindex.htm>
3. Click on `<goto test xTab>`
4. Click on `<goto test Listing>`
5. Use web browser button to Go Back
6. Click on `<down to Listing Level 2>` note you are mid-file
7. Use web browser button to Go Back, twice, to testIndex
8. Click on `<go to test Listing>`
9. Note Table of Contents at top of file testList
10. Click on `<Listing Level 3>`, note you are at end of file

----- testINDEX.HTM -----

```
<HTML> <TITLE>test: index</TITLE>
<BODY> <li><h2>Index</h2></li>
```

```
<li> <a href="testxTab.htm"> goto test xTab</a></li>
<li> <a href="testList.htm"> goto test Listing</a></li>
```

```
</BODY> </HTML>
```

----- testxTab.HTM -----

```
<HTML> <HEAD> <TITLE>test: Summary</TITLE> </HEAD>
<BODY> <li><h2>proc FREQ crosstabulation</h2></li>
```

```
<li> <a href="testList.htm#lv1"> goto Listing Level 1</a></li>
<li> <a href="testList.htm#lv2"> down to Listing Level 2</a></li>
<li> <a href="testList.htm#lv3"> Listing Level 3</a></li>
```

```
</BODY> </HTML>
```

Ray Pass, (2000), *You can get there from here and back again adding hot-link drill-down to %DS2HTM and %TAB2HTM Output* SUGI 25 Advanced Tutorials # 17-25

SAS® is a registered trademark of SAS® Institute, Inc. In the USA and other countries, ® indicates USA registration.

Author: Ronald Fehd **e-mail: RJF2@cdc.gov**
Centers for Disease Control MS-G25
4770 Buford Hwy NE
Atlanta GA 30341-3724 **voice:**
770/488-8102

ACKNOWLEDGMENTS

This knowledge and wisdom was gained over the last decade while I crunched numbers and read SAS-L. Many thanks to the other SAS® Whizards(tm) that contribute to SAS-L. You know who you are. I couldn't have done it without you.

----- testList.HTM -----

```
<HTML> <HEAD> <TITLE>test: List</TITLE> </HEAD> <BODY>
```

```
<li><h2>Table of Contents</h2></li>
```

```
<li> <a href="testList.htm#lv1"> goto Listing Level 1</a></li>
<li> <a href="testList.htm#lv2"> down to Listing Level 2</a></li>
<li> <a href="testList.htm#lv3"> Listing Level 3</a></li>
```

```
<li> <h2>Details</h2></li>
```

```
<li><a name="lv1"><h2>level 1</h2></a> </li>
<li>item 1.1 </li> <li>item 1.2 </li> <li>item 1.3 </li>
<li>item 1.4 </li> <li>item 1.5 </li> <li>item 1.6 </li>
<li>item 1.7 </li> <li>item 1.8 </li> <li>item 1.9 </li>
<li>item 1.10</li> <li>item 1.11</li> <li>item 1.12</li>
<li>item 1.13</li> <li>item 1.14</li> <li>item 1.15</li>
<li>item 1.16</li> <li>item 1.17</li> <li>item 1.18</li>
<li>item 1.19</li> <li>item 1.20</li>
<li><a name="lv2"><h2>level 2</h2></a></li>
<li>item 2.1 </li> <li>item 2.2 </li> <li>item 2.3 </li>
<li>item 2.4 </li> <li>item 2.5 </li> <li>item 2.6 </li>
<li>item 2.7 </li> <li>item 2.8 </li> <li>item 2.9 </li>
<li>item 2.10</li> <li>item 2.11</li> <li>item 2.12</li>
<li>item 2.13</li> <li>item 2.14</li> <li>item 2.15</li>
<li>item 2.16</li> <li>item 2.17</li> <li>item 2.18</li>
<li>item 2.19</li> <li>item 2.20</li>
<li><a name="lv3"><h2>Level 3</h2></a></li>
<li>Item 3.1 </li> <li>Item 3.2 </li> <li>Item 3.3 </li>
<li>item 3.4 </li> <li>item 3.5 </li> <li>item 3.6 </li>
<li>item 3.7 </li> <li>item 3.8 </li> <li>item 3.9 </li>
<li>item 3.10</li> <li>item 3.11</li> <li>item 3.12</li>
<li>item 3.13</li> <li>item 3.14</li> <li>item 3.15</li>
<li>item 3.16</li> <li>item 3.17</li> <li>item 3.18</li>
<li>item 3.19</li> <li>item 3.20</li>
</BODY> </HTML>
```

```

; /* XTAB2HTM 00Jun30 * * * * *
USAGE : %XTAB2HTM(DATA,VAR1,VAR2,ID(S),DIRECTORY-PATH);
        %XTAB2HTM(TEST,_NAME_,TEXT,ID,C:\TEMP\);
DESCRIP : FREQ CrossTabulation to HTML
        write ODS file INDEX.HTML & pairs: *XTAB.HTML + *LIST.HTML
PROCESS : 0. SORT and create copy: DETAILS
        1. FREQ of VAR1 to output data set: INDEX
        2. add html anchors/links to INDEX
        3. write ODS file INDEX.html
        4. list processing: make subsets, write HTMWWRITE
KEYWORDS: list-processing HTML ODS anchor href word
AUTHOR : Ronald Fehd the macro maven
        Centers for Disease Control Atlanta GA USA
e-mail : RJF2@cdc.gov voice: 770/488-8102
NOTE : illustration of the Writing for Reading SAS(r) Style Sheet. */
%MACRO XTAB2HTM(*- - - - - */
DATA /* data set name */
,VAR1 /* proc FREQ cross-table first var */
,VAR2 /* proc FREQ cross-table second var */
,ID /* identifier(s) */
,PATH /* destination of HTML files */
,HTML =htm /* file extension in(htm,html) */
,INDEX =index/* name of top-level file in (home,index) */
,LIST =list /* suffix of proc PRINT DETAILS file */
,XTAB =xTab /* suffix of proc FREQ cross-tab file */
,TITLEN =3 /* title line number to write Table of Contents */
,TESTING =0 /* enable testing msgs and prints */
);/*-----*
RJF2 00May24 runs OK w/numerics with number-to-char conversion notes
RJF2 00Jun30 polishing: changed build of call execute data set
. . . . . */
%IF &TESTING %THEN %DO;options mprint; %END;
%*0;proc SORT data = &DATA.
        (keep= &VAR1. &VAR2. &ID.)
        out = DETAILS;
        by &VAR1. &VAR2. &ID.;
%*1;proc FREQ data = DETAILS;
        tables &VAR1.
        /out = INDEX
        %IF not &TESTING %THEN %DO;
                noprint %END;
        ;
%local LEVELWID;/**order of magnitude of number of data sets to write
        ,LEVELWID=1 max: 9 levels LEVELWID=2 max:99 levels */
data _NULL_;
        if 0 then set INDEX nobs=Nobs;
        call symput("LEVELWID",compress(length(compress(put(Nobs,32)))));
                                                stop;run;
                                                %put LEVELWID<&LEVELWID>;

%*2 add links to FREQ:INDEX of &VAR1.;
data INDEX;
attrib Ndx&VAR1. length= $ 200 label="&VAR1. summary"
        NdxCount length= $ 200 label="&VAR1. count"
        FileName length= $ %eval(1 + &LEVELWID.)
        VarLabel length= $ 40 label="Var Label";
retain FileNbr 0 VarLabel '';
drop FileNbr ;
set INDEX;
call label(&VAR1.,VarLabel);
FileNbr +1;
FileName = ' ' !! put(FileNbr,z&LEVELWID.);

```

```

%*expected: <a href ="VARNAMExTab.html"> VARNAME </a>;
Ndx&VAR1.= '<a href =' ' !! FileName      !! "&XTAB..&HTML." !! ' ">'
           !! &VAR1.          !! '</a>';
NdxCount = '<a href =' ' !! FileName      !! "&LIST..&HTML." !! ' ">'
           !! put(Count,32.) !! '</a>';
%IF &TESTING %THEN %DO; proc PRINT data = INDEX;                                %END;
                                                                    run;
%*3 write INDEX.html;                                ods listing close;
ods html body = "&PATH.&INDEX..&HTML.";

proc PRINT data = INDEX label
    split = ' ';
    format Percent 5.1;
    var     Ndx: Percent;*Count;
    sum     Percent Count;
                                                                    run;
                                                                    ods html close;

ods listing;

/*4 write: 1. data set list:          DATA
                                       _1
                                       _2
                                       ;set DETAILS;
2. subset output:                    if Var = '<value1>' then output _1;
                                       if Var = '<value2>' then output _2;
                                       RUN;
3. HTML                             : %HTMWRITE(_1,...);
                                       %HTMWRITE(_2,...); . . . . . */
data _NULL_;
file PRINT;

%*4.1 write data set list;            call execute('data ');
do until(EndoFile);
    set INDEX(keep = FileName)
        end = EndoFile;            call execute(FileName);    end;
EndoFile = 0;%*reset for next loop;
%*DATA closure, write set statement; call execute(';set DETAILS;');

%*4.2 write subset output statements;
do until(EndoFile);
    set INDEX(keep = &VAR1. FileName)
        end = EndoFile;            call execute("if &VAR1.='
                                       !! trim(&VAR1.)
                                       !! '' then output "
                                       !! FileName !! ';'');
                                       end;

EndoFile = 0;%*reset for next loop;
%*execute make of subsets;            call execute('RUN;');

%*4.3 HTMWRITE(...);
do until(EndoFile);
    set INDEX(keep = &VAR1. FileName VarLabel)
        end = EndoFile;            call execute('%HTMWRITE('
                                       !! FileName !! ','
                                       !! trim(VarLabel)
                                       !! ",&VAR1.,&VAR2.,&PATH.);");
                                       end;
                                                                    stop;run;
%*..... XTAB2HTM; %MEND;

%MACRO HTMWRITE(/*- - - - - */
DATA /* data set name */

```

```

,LABEL /* data label: variable value */
,VAR1 /* proc FREQ cross-table first var */
,VAR2 /* proc FREQ cross-table second var */
,PATH /* destination of HTML files */
);/*this macro called by XTAB2HTM
NOTE: called via call execute: must have zero complexity:
      NO call symput nor %IF
assumption: data set name is a subset
      data set containing only VAR1 values equal to level# of data set name
output: two HTML files
1. proc FREQ : cross-table      name: VARxTab
2. proc PRINT: detail list: see IDs name: VARlist
note: references to: HTML INDEX LEVELWID LIST XTAB TITLEN
      these are local to XTAB2HTM
. . . . . */
proc FREQ data = &DATA.;
      label &VAR2.="&LABEL.";
      tables &VAR2.
      /      noprint list missing nocum
      out = TBL_CONTENTS;

/*process TBL_CONTENTS: add anchor-href and -names: goto labels
      <who> is the label, aka fragment identifier
<a href="index.html#who">What's That?</a>
<a href=      "#who">What's That?</a>
<a name=      "who">What's That?</a> * . . . . . */

DATA TBL_CONTENTS;
attrib Link length = $200 label = "&LABEL."
      ToC length = $200 label = "&LABEL."
      Suffix length = $200
      Prefix length = $ 9;%*123456789;
retain Prefix ' <a href=" ' %*dQuote open;
      Row 0;
drop Row ;
set TBL_CONTENTS;
Row +1;
Suffix = '#lv1'
      !! compress(put(Row,32.)) !! '>' %*dQuote close;
      !! trim(&VAR2.) !! '</a>';
ToC = Prefix !! Suffix;
Link = Prefix !! "&DATA.&LIST..&HTML." %*double dots: fileref.ext;
      !! Suffix;

DATA &DATA.;
attrib Label length = $200;
retain Level 0;
drop &VAR2.;
set &DATA.;
by &VAR2.;
if first.&VAR2. then do; Level + 1;
      Label = ' <a name="lv1' %*dQuote open ;
      !! compress(put(Level,32.))
      !! '>' %*dQuote close;
      !! trim(&VAR2.) !! '</a>'; end;
else Label = &VAR2.; run;
ods listing close;

ods html body = "&PATH.&DATA.&XTAB..&HTML.";
FOOTNOTE ' <a href = " ' "&DATA.&XTAB..&HTML." ' ">top of file </a>'
'&nbsp;' ' <a href = " ' "&INDEX..&HTML." ' ">back to index</a>';
%*&nbsp;: No BackSpace;

```

```

proc PRINT data = TBL_CONTENTS label;
    var    Link Count Percent;
    sum    Count Percent;
                                run;
                                ods html close;

ods html body = "&PATH.&DATA.&LIST..&HTML.";
FOOTNOTE ' <a href = "' "&DATA.&LIST..&HTML." '">top of file </a>'
'&nbsp;' ' <a href = "' "&INDEX..&HTML." '">back to index</a>';

TITLE&TITLEN. "Table of Contents";
proc PRINT data = TBL_CONTENTS label;
    var    ToC Count Percent;
    sum    Count Percent;
                                run;
                                TITLE&TITLEN.;

proc PRINT data = &DATA.
    (drop = &VAR1.)
    label;
    label Label = "&LABEL.";
    by    Level;
    id    Level;
                                run;
                                footnote;
                                ods html close;
ods listing;%*.....HTMWWRITE; %MEND;

/*TEST DATA ***** enable w/slash at end of line: **
DATA TEST;
*label _Name_ = 'text variable';%*enable to use label instead of varname;
infile CARDS;input @ 1 ID $char4. @ 1 Nbr 1.
@ 6 _Name_ $char8. @15 Text $char20.; cards;
0811 TEST TEST PERFORMED DAILY
0811 CODENONT TEST PERFORMED DAILY
0811 CODENONT TEST PERFORMED DAILY
2541 CODENONT TEST NOT PERFORMED I
4591 CODENONT TEST NOT PERFORMED I
9951 CODENONT TEST ONLY PERFORMED
7251 PRCNAMET ABBOT AXSYM HIV-1/HI
9941 PRCNAMET ABBOTT AXSYM
3631 PRCNAMET ABBOTT AXSYM
6661 PRCNAMET ABBOTT AXSYM
2771 PRCNAMET ABBOTT AXSYM
1422 PRCNAMET ABBOTT AXSYM HIV-1/H
2061 PRCNAMET ABBOTT AXSYM HIV-1/H
3961 PRCNAMET ABBOTT AXSYM HIV-1/H
1321 PRCNAMET ABBOTT AXSYM HIV1/HI
;%*run only one test at a time: character or numeric;
%XTAB2HTM(TEST,_NAME_,TEXT,ID,C:\TEMP\,TESTING=1);
%*XTAB2HTM(TEST,_NAME_,TEXT,ID,C:\TEMP\);
%*XTAB2HTM(TEST, Nbr ,TEXT,ID,C:\TEMP\);%*numeric to char conversion;
/*****/

```

