Advanced PROC REPORT:
Getting Your Tables Connected Using Links

Arthur L. Carpenter
California Occidental Consultants

ABSTRACT

Gone are the days of strictly paper reports. Increasingly we are being asked to render our tables and reports using a variety of electronic file types that can be browsed and read using our computers. Besides the cost savings associated with the purchase of paper, paperless reports also minimize the costs associated with the printing and distribution of the reports. Another, often overlooked and potentially more important, side benefit of the paperless report is the increased availability of linked graphs, reports and tables. Navigation of the paper report is dependent on the table of contents, a strong index, and visual aids such as captions and footnotes; however in paperless reports we can easily jump from one table to its supporting and dependent tables with the click of a mouse. Easily that is if we have created the necessary links.

Within PROC REPORT and with the help of the Output Delivery System, there are a number of techniques that we can use to build and maintain these links. Individually these techniques are not complicated, however we do need to be aware of the syntax, alternative approaches, and issues associated with the automation of the process that coordinates the links between tables.

KEYWORDS

PROC REPORT, ODS, PDF, RTF, HTML, hyperlink, links, CALL DEFINE, STYLE=

INTRODUCTION

As we move away from reports that are generated strictly for printing on paper, we can take advantage of a number of techniques that can be used to link one table to another. In its more sophisticated application, these techniques allow us to even link individual cells of our report to another report or table. These links, or hyperlinks as they are more formally known, are used to point from a specific location in one table to another table.

Generally your linked tables will all be of the same type (HTML, PDF, or RTF), but there is no reason why this has to be the case. In the examples shown in this paper HTML tables link to HTML tables and so on, however when you create a reference to a file, it rarely matters which of these file types you are pointing to or from. An HTML table can link to a PDF file and so on.

The process of moving from one table to a linked table of finer detail is known as drilling down, and this is one of the most common applications of linked tables.

Obviously the process does not apply for documents that are being viewed on paper, however through ODS we now have a number of choices of destinations that allow us to display our documents in a variety of formats that lend themselves to electronic display.

In the following paper the discussion includes examples that show how:

- Titles and footnotes can be used to form links to other documents or locations within a document

- To use the STYLE= option to create links
- The CALL DEFINE can be used to create links for HTML, PDF, or RTF files.
- To use destination specific techniques for HTML, PDF, and RTF files.
- Automate the process of building links through the use of the SAS macro language to save time and to increase accuracy.
- Build links through the use of user defined formats.

Because of the overlap among destinations, if you are new to linked documents or are not very well versed in ODS, it will probably be wise to read over all of these sections not just those associated with the destination of primary interest.

LINKING TITLES AND FOOTNOTES

HTML Anchor Tags

Although some knowledge of HTML is helpful, it fortunately is not particularly necessary to create linked HTML tables. You will, however, need to understand the basic structure of the HTML anchor tag statement. Its general syntax is:

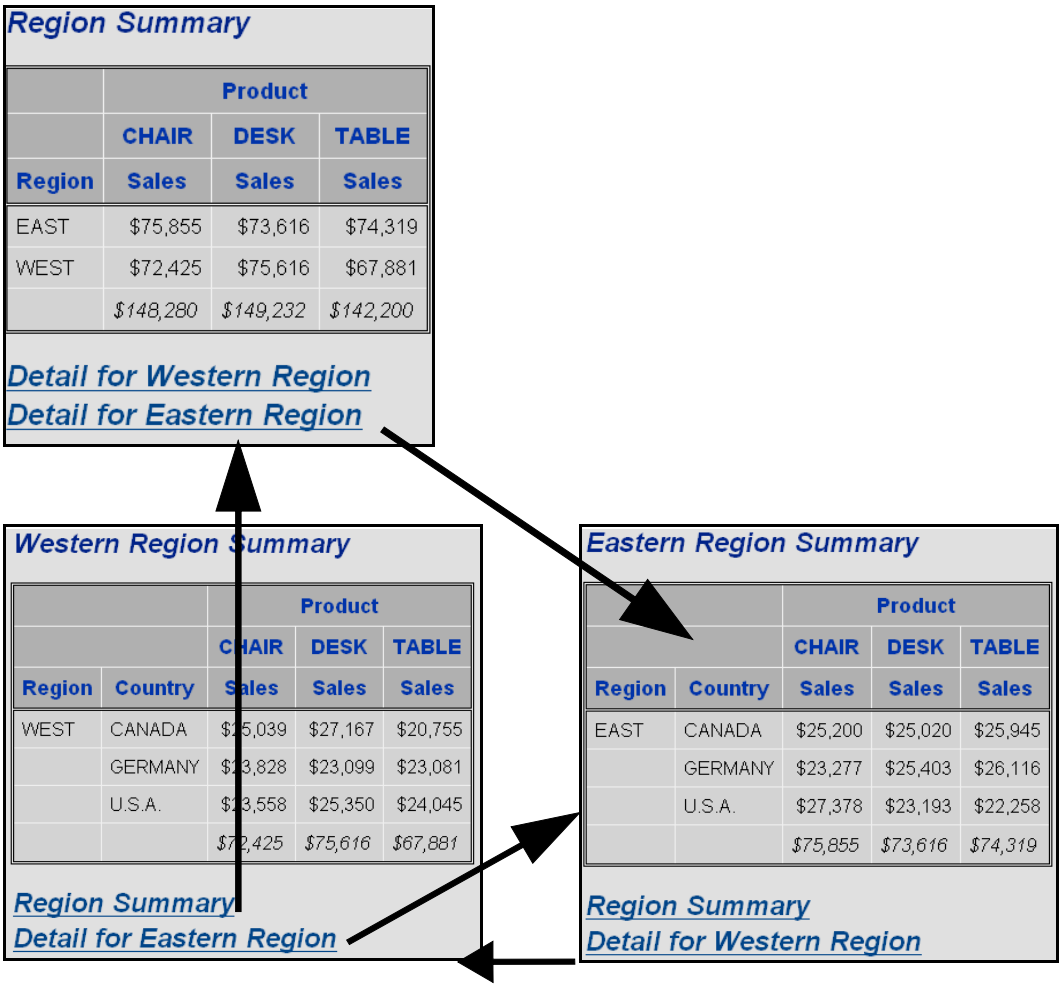
```
<a href='file_name.html' >display_text</a>
```

When the HTML statement appears in a title or footnote, the *display_text* is displayed. If the *display_text* is selected by the reader, the browser then links to and displays the file named by the HREF= option.

In the following, somewhat silly, example three reports are generated. The first is the summary of the two regions and then the detail reports for each of those regions. Each report is directly linked to the other two through the FOOTNOTE statements, each of which contains HTML anchor tags.

```
* Regional Report  **********************;
ods html style=default
         path="&path\results" (url=none) ❶
         body='Exercise1_Region.html';

title1 'Region Summary';
footnote1 "<a href='Exercise1_RegionWEST.html' ❷
            >Detail for Western Region</a>";
footnote2 "<a href='Exercise1_RegionEAST.html'
            >Detail for Eastern Region</a>";

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
            nowd;
   column region product,actual;
   define region  / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
```

```
ods html close;

* Western Region Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise1_RegionWEST.html';

title1 'Western Region Summary';
footnote1 "<a href='Exercise1_Region.html'
             >Region Summary</a>";
footnote2 "<a href='Exercise1_RegionEAST.html'
             >Detail for Eastern Region</a>";

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='WEST')) ❸
            nowd;
   column region country product,actual;
   define region  / group;
   define country / group; ❹
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
ods html close;

* Eastern Region Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise1_RegionEAST.html';

title1 'Eastern Region Summary';
footnote1 "<a href='Exercise1_Region.html'
             >Region Summary</a>";
footnote2 "<a href='Exercise1_RegionWEST.html'
             >Detail for Western Region</a>";

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='EAST'))
           nowd;
   column region country product,actual;
   define region  / group;
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
ods html close;
```



❶ The URL=NONE option allows indirect addressing in the internal HTML code. Generally a good idea anyway, this option allows you to move your linked images to other locations.

❷ Each set of footnotes always references the other two tables.

❸ The WHERE= option includes a subsetting clause for REGION.

❹ COUNTRY is added to the COLUMN statement as a grouping variable.

Links can also be created through the use of the LINE statement. The linked footnotes used in the previous example are replaced by LINE statements in the following example

The REPORT step that creates the overall summary becomes:

```
title1; ❺
footnote1;

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
            nowd;
   column region product,actual;
```

```
    define region  / group;
    define product / across;
    define actual  / analysis sum
                     format=dollar8.
                     'Sales';
    rbreak after / summarize;

    compute before _page_; ❻
       line @3 'Region Summary'; ❼
    endcomp;

    compute after; ❽
       line @3 "<a href='Exercise2_RegionWEST.html'
               >Detail for Western Region</a>";
       line @3 "<a href='Exercise2_RegionEAST.html'
               >Detail for Eastern Region</a>";
    endcomp;
    run;
ods html close;
```

❺ No title or footnotes are defined. Instead both will be controlled with LINE statements.

❻ The LINE statement is to write at the top of the page.

❼ This is to be the report title.

❽ At the end of the report we write the two anchor tags this time using the LINE statement instead of the FOOTNOTE statement.

Region Summary

	Product		
	CHAIR	DESK	TABLE
Region	Sales	Sales	Sales
EAST	$75,855	$73,616	$74,319
WEST	$72,425	$75,616	$67,881
	$148,280	*$149,232*	*$142,200*

Detail for Western Region
Detail for Eastern Region

HTML using the DEFAULT style

In some versions of SAS, you may need to have a <DIV> and </DIV> tag surrounding the anchor tag in the title or footnote so that the parser/processor will not interpret the < and > as 'less than' and 'greater

than' comparison operators.

Using the LINK= Option

Both the TITLE and FOOTNOTE statements support the LINK= option. This option allows you to directly specify the link without using the anchor tags shown earlier in this section. Also, unlike the anchor tags, which are used with the HTML destination, the LINK= option can generally also be used with the PDF and RTF destinations.

The following PDF example takes the first example of this section and replaces the HTML anchor tags with the LINK= option. Since PDF and RTF footnotes tend to be at the bottom of the page (rather than at the bottom of the report), the footnotes have been replaced with titles for this example.

```
* Regional Report  **********************;
ods pdf style=printer
        file="&path\results\Exercise3_Region.pdf";

title1 'Region Summary';
title2 link='Exercise3_RegionWEST.pdf'
             "Detail for Western Region";
title3 link='Exercise3_RegionEAST.pdf'
             "Detail for Eastern Region";
```

Usually the LINK= option will work for both the PDF and RTF destinations. However depending on the level of PDF file created by your system, and the word processor opening an RTF file, sometimes LINK= will not be able to create a valid link for those destinations.

Building a series of tables like these can be time consuming and tedious. Fortunately the macro language excels at building this type of code.

HTML ANCHOR TAGS AS DATA VALUES

Anchor tags can also be placed in data fields as well as column and row labels. The tags can be built into a data value in a DATA step or in a compute block. Since the latter is more fun, this is approach taken in the next example.

```
ods listing close;

* Regional Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise4_Region.html';

title1 'Region Summary';
footnote1;

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
      nowd;
   column region regtag product,actual; ❶
   define region  / group noprint; ❷
   define regtag  / computed format=$4. 'Region'; ❸
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
```

```
                    'Sales';
   compute regtag / char length=60; ❹
      if region='WEST' then ❺
         regtag = "<a href='Exercise4_RegionWEST.html'>West</a>";
      else if region='EAST' then
         regtag = "<a href='Exercise4_RegionEAST.html'>East</a>";
   endcomp;
   rbreak after / summarize;
   run;
ods html close;
```

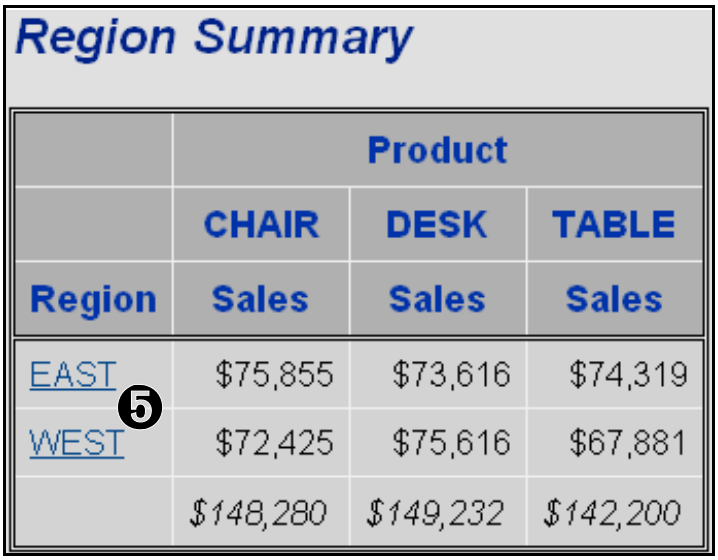
❶ Add the computed variable REGTAG to the COLUMN statement.

❷ The variable REGION will not be printed.

❸ Define the computed variable which will hold the anchor tag.

❹ Even though only four characters are displayed be sure to use a LENGTH= sufficient to hold the whole anchor tag designation.

❺The anchor tag for each region is assigned to the computed variable.



Region Summary

	Product		
	CHAIR	DESK	TABLE
Region	Sales	Sales	Sales
EAST ❺	$75,855	$73,616	$74,319
WEST	$72,425	$75,616	$67,881
	$148,280	*$149,232*	*$142,200*

HTML using the DEFAULT style

The code that generates the detailed reports for each of the two regions are similar to that shown above. Instead of grouping on REGION, however, we are using COUNTRY. The link for the summary tables for the individual regions point back to the regional summary ❼.

```
* Western Region Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise4_RegionWEST.html';

title1 'Western Region Summary';

proc report data=sashelp.prdsale
```

```
                    (where=(prodtype='OFFICE' and region='WEST'))
      nowd;
   column country ctag product,actual;
   define country / group noprint;
   define ctag    / computed format=$7. 'Country'; ❻
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   compute ctag / char length=60;
      if _break_='_RBREAK_' then ❼
         ctag = "<a href='Exercise4_Region.html'>Total</a>";
      else ctag=country;
   endcomp;
   rbreak after / summarize;
   run;
ods html close;
```

❻ Define the computed variable, CTAG, to hold the anchor tag for country.

❼ On the line summarizing the region, place an anchor tag that points back to the overall summary.

Western Region Summary

	Product		
	CHAIR	DESK	TABLE
Country	Sales	Sales	Sales
CANADA	$25,039	$27,167	$20,755
GERMANY	$23,828	$23,099	$23,081
U.S.A.	$23,558	$25,350	$24,045
Total ❼	*$72,425*	*$75,616*	*$67,881*

HTML using the DEFAULT style

ESTABLISHING LINKS USING CALL DEFINE

Rather than creating special computed variables, you can specify the file reference directly by using the CALL DEFINE statement.

The same links are created in the following examples, however here there are no computed variables. Instead the CALL DEFINE statement is used with the URL attribute to assign the URL to the column values. The examples in this section are for the HTML destination, however the URL also works for PDF

and, depending on the word processor, the RTF destination as well.

```
* Regional Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise5_Region.html';

title1 'Region Summary';
footnote1;

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
            nowd;
   column region product,actual;
   define region  / group ;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   compute region; ❶
      rtag = "Exercise5_Region"||trim(region)||".html"; ❷
      call define(_col_,'url',rtag); ❸
   endcomp;
   rbreak after / summarize;
   run;
ods html close;
```

❶ A compute block is established for the variable to which we want to assign the link.

❷ The temporary variable RTAG will be used to store the location. Notice that the value to be displayed is NOT included, only the location. The value of the variable REGION will be the display value. This makes our coding easier. We can also make the assignment of the location directly without first creating a temporary variable ❻.

❸ The location stored in the temporary variable RTAG is a URL attribute value for this column.

Region Summary

	Product		
	CHAIR	DESK	TABLE
Region	Sales	Sales	Sales
EAST	$75,855	$73,616	$74,319
WEST	$72,425	$75,616	$67,881
	$148,280	*$149,232*	*$142,200*

Eastern Region Summary

	Product		
	CHAIR	DESK	TABLE
Country	Sales	Sales	Sales
CANADA	$25,200	$25,020	$25,945
GERMANY	$23,277	$25,403	$26,116
U.S.A.	$27,378	$23,193	$22,258
Region	*$75,855*	*$73,616*	*$74,319*

In this example we have decided that the summary for an individual region (here the 'Eastern Region Summary' is shown) will only link back to the primary table ('Region Summary'). This means that the

detail summaries for the individual regions we will have a link only for one value in the column.

```
* Western Region Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise5_RegionWEST.html';

title1 'Western Region Summary';

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='WEST'))
      nowd;
   column country product,actual;
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   compute country;
      if _break_='_RBREAK_' then do; ❹
         country = 'Region'; ❺
         call define(_col_,'url',"Exercise5_Region.html"); ❻
      end;
   endcomp;
   rbreak after / summarize;
   run;
ods html close;
```

❹ We only want to create the link for the summary line.

❺ The link will ONLY be available if there is something for it to attach to in the cell. Since country is otherwise missing (blank) for this summary row, we have added text to the cell.

❻ Rather than create a temporary variable (as was done at ❷), the value has been placed directly into the CALL DEFINE statement.

In the previous example three HTML files were created each with links that pointed to other files. It is also possible to create links that point to other locations within a document. The following example builds on the previous example, except rather than creating three files, it creates a single file with three internal links.

When pointing to an internal location an extension is added to the file name ❷ using a pound sign. In this case we add the region (EAST or WEST).

```
* Regional Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise6.html'; ❶

title1 'Region Summary';
footnote1;

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
      nowd;
   column region product,actual;
   define region  / group ;
```

```
define product / across;
define actual  / analysis sum
                 format=dollar8.
                 'Sales';
compute region;
   rtag = "Exercise6.html#"||trim(region); ❷
   call define(_col_,'url',rtag);
endcomp;
rbreak after / summarize;
run;
```

❶ A single HTML file will be used to hold all three reports.

❷ The internal link is named by appending the link identifier to the file name. The identifier follows the # sign. This identifier will be used in the ANCHOR= option ❸ to tie the individual reports together.

WITHOUT closing the HTML destination, the two reports for the individual regions are generated. The ODS HTML statement ❸ that precedes the REPORT step does not open a new report (there is no BODY= option), rather it only exists to add the ANCHOR= option.

```
* Western Region Report  **********************;
ods html anchor='WEST'; ❸

title1 'Western Region Summary';

proc report data=sashelp.prdsale
                  (where=(prodtype='OFFICE' and region='WEST'))
            nowd;
   column country product,actual;
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   compute country;
      if _break_='_RBREAK_' then do;
         country = 'Region';
         call define(_col_,'url',"Exercise6.html"); ❹
      end;
   endcomp;
   rbreak after / summarize;
   run;
```

❸ The ANCHOR= option provides the identifier that follows the # sign at ❷.

❹ Since there is no anchor specification (no # sign) for this link, this link points back to the top of the report.

FORMING LINKS USING STYLE=

Considering the considerable overlap between the capabilities of the CALL DEFINE routine and the STYLE= option, it should not be surprising that you can form URL links through the STYLE= option as well. Since this option is not executable as is the CALL DEFINE routine it is more suitable when the link is either a constant or at least does not include data dependencies.

In the following example, we want to link from the region specific summary back to the overall summary. The STYLE= option is used to form the link. The code for the Western Region becomes:

```
ods html style=default
         path="&path\results" (url=none)
         body='Exercise7_RegionWEST.html';

title1 'Western Region Summary';

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='WEST'))
            nowd;
   column region country product,actual;
   define region  / group
                    style(header)={url='Exercise7_Region.html'}; ❶
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
ods html close;
```

❶ The header for REGION is made to be a link through the use of the URL attribute.

Western Region Summary

		Product		
		CHAIR	DESK	TABLE
Region ❶	Country	Sales	Sales	Sales
WEST	CANADA	$25,039	$27,167	$20,755
	GERMANY	$23,828	$23,099	$23,081
	U.S.A.	$23,558	$25,350	$24,045
		$72,425	*$75,616*	*$67,881*

HTML using the DEFAULT style

Establishing links with the STYLE= option is also appropriate for PDF and RTF file types.

CREATING LINKS IN A PDF DOCUMENT

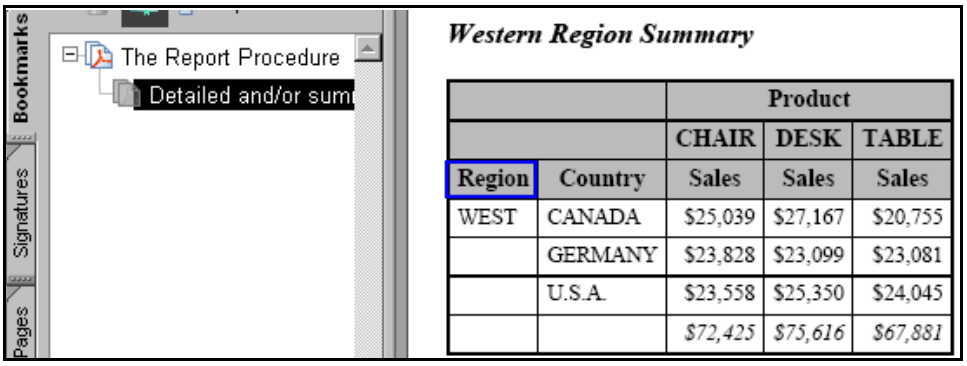
The syntax for creating linked documents when using the PDF destination is very similar each of the

previous methods used with HTML. The difference is in the appearance of the link on the table, and how the files are addressed in the code (with an extension of PDF rather than HTML).

The example below creates a series of linked PDF documents. For the PDF destination the style has been specified as PRINTER. This is the default style for PDF, but I like to explicitly specify the STYLE= option, even when it is the default. The code that creates the table for the Western Region is shown:

```
ods pdf style=printer
        file="&path\results\Exercise8_RegionWEST.pdf";

title1 'Western Region Summary';

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='WEST'))
            nowd;
   column region country product,actual;
   define region  / group style(header)={url='Exercise8_Region.pdf'};
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
ods pdf close;
```

The resulting link is located on the column header. The report table for the Western Region ("Exercise8_RegionWEST.pdf") is:



Western Region Summary

		Product		
		CHAIR	DESK	TABLE
Region	Country	Sales	Sales	Sales
WEST	CANADA	$25,039	$27,167	$20,755
	GERMANY	$23,828	$23,099	$23,081
	U.S.A.	$23,558	$25,350	$24,045
		$72,425	*$75,616*	*$67,881*

PDF using the PRINTER style

In the previous example three different documents are linked. It would also have been possible to create a single document with links pointing to other locations within that one document. In the following example, a single PDF file is created with the same three interconnected tables as in the previous example. However the links all point to other places within the same PDF file rather than to other PDF files.

```
* Regional Report  **********************;
ods pdf style=printer
        file="&path\results\Exercise9.pdf"; ❶

title1 'Sales Summary';
```

```
ods proclabel='Sales Summary'; ❷
proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
            nowd
            contents='Overall' ❸
            ;
   column region product,actual;
   define region  / group ;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   compute region;
      rtag = "#"||trim(region); ❹
      call define(_col_,'url',rtag);
   endcomp;
   rbreak after / summarize;
   run;

* Western Region Report  **********************;
ods pdf anchor="WEST" ❺
        startpage=now; ❻
ods proclabel="Western";

title1 'Western Region Summary';

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='WEST'))
            nowd; ❼
   column country product,actual;
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;

* Eastern Region Report  **********************;
ods pdf anchor="EAST"
        startpage=now;
ods proclabel="Eastern";


title1 'Eastern Region Summary';

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='EAST'))
            contents='' ❽
            nowd;
   column country product,actual;
   define country / group ;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
```

```
ods pdf close;
```

❶ A single PDF file is written for all three REPORT steps.

❷ The ODS PROCLABEL= can be used to provide a name for the bookmark tab.

❸ The CONTENTS= option provides an additional location with the associated text on the bookmark panel.

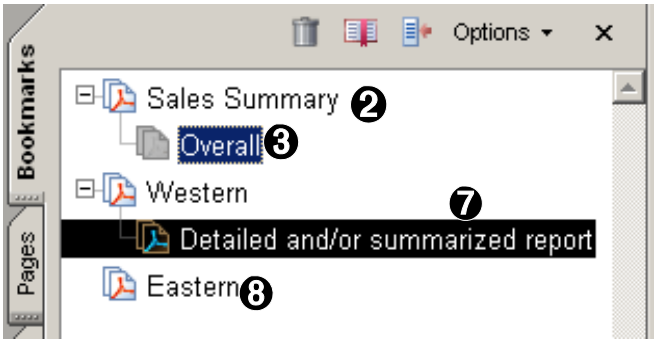
❹ The CALL DEFINE statement is used to create the association with the text and the specified link. Notice that since the link is to be internal to the document, it is prefixed with a # sign. For this report the two links will be #WEST and #EAST.

❺ The ANCHOR= option is used to specify the link to the output from the upcoming procedure. Notice that the link does NOT include a # sign (it is assumed).

❻ The STARTPAGE= option forces a new page in the PDF document.

❼ The bookmark area contains a default value when the CONTENTS= option is not included.

❽ The CONTENTS= option overrides the default contents value ❼ that is displayed in the bookmark section. When you want to suppress the value altogether you should try using `CONTENTS=''` (although there have been some problems reported with this option for SAS9.1).



Typically bookmarks are created for each step between ODS PDF and the ODS PDF CLOSE, however the bookmarks only become linked by using the ANCHOR= option. The display of the bookmarks can be controlled through the use of the BOOKMARKLIST= option.

```
* Regional Report  **********************;
ods pdf style=printer
        file="&path\results\Exercise9.pdf"
        bookmarklist=hide;
```

This option can take on the values of:

- none — the bookmarks are not created
- hide — the bookmarks are created but not displayed (until requested)
- show — the bookmarks are displayed as above (this is the default)

The table of bookmarks can also be turned off by using the NOTOC option on the ODS PDF statement.

CAVEATS
As was mentioned earlier, the ODS CONTENTS= option does not always perform as expected in SAS9.1. This becomes more evident as the tables become more complex and especially when BREAK statements are included. Extensive changes are anticipated for SAS9.2 that should correct these problems.

It is hoped that PROC REPORT and PROC DOCUMENT will work together in SAS9.2. If so tracking bookmarks should become much easier.

CREATING LINKS IN AN RTF DOCUMENT

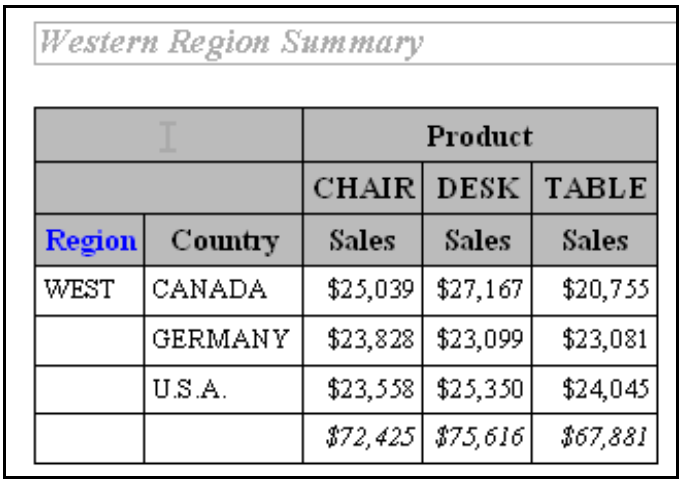
The generation of links in RTF is similar to the process described above.

```
* Western Region Report  **********************;
ods rtf style= rtf ❶
        file="&path\results\Exercise10_RegionWEST.rtf"; ❷

title1 'Western Region Summary';

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE' and region='WEST'))
      nowd;
   column region country product,actual;
   define region  / group style(header)={url='Exercise10_Region.rtf'❸};
   define country / group;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
ods rtf close;
```

❶ The RTF destination and the RTF file extension ❷ create the RTF file that contains the table with the embedded links. The RTF style has been especially designed for use with the RTF destination.

A portion of this report as it is viewed in MSWord is shown here.

❸ This extension could also be PDF or HTML if you wanted to link to a non-RTF file.



Western Region Summary

		Product		
		CHAIR	DESK	TABLE
Region	Country	Sales	Sales	Sales
WEST	CANADA	$25,039	$27,167	$20,755
	GERMANY	$23,828	$23,099	$23,081
	U.S.A.	$23,558	$25,350	$24,045
		$72,425	*$75,616*	*$67,881*

When viewed in a word processor, such as MSWORD, the header 'Region' is linked to overall summary 'Exercise10_Region.rtf'. Although hard to see in black and white, by default the label is shown in an alternate color.

RTF using the RTF style

When you want to follow a link from a RTF document be sure to use the control key with a click, rather than a double click.

AUTOMATION USING THE MACRO LANGUAGE

Building a series of linked tables by hand can be tedious. Two, perhaps three tables, and I have reached my tolerance for repeated code. Fortunately the SAS macro language has a number of extremely powerful techniques that can be used to automate the process of generating the necessary code.

In several of the previous examples one primary table is used as the index to point to a series of secondary tables. In this particular example there are only two secondary tables, one for each region, however, if the number of regions was either unknown or perhaps dependent on the data, we would need to write more flexible code.

The idea is to create code that removes data dependencies or hard coded data elements. The following code generalizes one of the earlier examples to work for any number of regions. When you generalize code in this way, you need to watch for things like:

```
(where=(prodtype='OFFICE' and region='WEST'))
```

In this WHERE clause both the value of PRODTYPE and REGION are hardcoded, and to generalize for all regions we will need to eliminate any hardcoded items that change within the program.

The first step in the process is to determine the number of regions and their individual values. One easy way to do this is to use a PROC SQL step to create a series of macro variables.

```
%macro linked(prod=OFFICE); ❶
%local i;

* Determine the count and list of regions;
proc sql noprint;
   select distinct region ❷
      into :reg1- :reg99 ❸
         from sashelp.prdsale(where=(prodtype="&prod"));
   %let regcnt = &sqlobs; ❹
```

```
quit;
```

❶ Since we will be using macro %DO loops, we need to define a macro.

❷ We are interested in each distinct value of the variable REGION.

❸ Save each individual value of REGION into a macro variable of the form &REG1, &REG2, &REG3, ... This allows up to 99 distinct regions. There is no real penalty for picking a number that is too big.

❹ The SQL step will count the number of distinct values of REGION and store them temporarily in the macro variable &SQLOBS. Save this number in the macro variable &REGCNT.

The REPORT step that creates the index table does not need to change as it will automatically adjust for each value of REGION ❺. In a sense it is already generalized.

```
* Regional Report  **********************;
ods rtf style=rtf
        file="&path\results\Exercise11_Region.rtf";

title1 'Region Summary';
footnote1;

proc report data=sashelp.prdsale
                    (where=(prodtype="&prod"))
            nowd;
   column region product,actual;
   define region  / group ;
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   compute region;
      rtag = "Exercise11_Region"||trim(region)||".rtf"; ❺
      call define(_col_,'url',rtag);
   endcomp;
   rbreak after / summarize;
   run;
ods rtf close;
```

❺ Since we are using the name of the region (which is the value of the variable REGION) as the non-constant part of the name of the file, this portion of the code is already data independent and does not require any further generalization.

Rather than create a separate PROC REPORT step for each region, we will generalize the step and put it inside of a macro %DO loop ❻, which will be executed once for each region (the number of regions is stored in &REGCNT). Whenever we want to code for a particular value of REGION we use the indirect macro variable reference `&&reg&i` ❼.

```
%do i = 1 %to &regcnt; ❻
   * Individual Region Report  **********************;
   ods rtf style=rtf
           file="&path\results\Exercise11_Region&&reg&i❼...rtf";

   title1 "Region Summary for &&reg&i"❼;

   proc report data=sashelp.prdsale
```

```
                  (where=(prodtype="&prod" and region="&&reg&i"❼))
                 nowd;
      column region country product,actual;
      define region  / group
style(header)={url='Exercise11_Region.rtf'};
      define country / group;
      define product / across;
      define actual  / analysis sum
                       format=dollar8.
                       'Sales';
      rbreak after / summarize;
      run;
   ods rtf close;
%end; ❻
%mend linked;❽
%linked(prod=OFFICE) ❾
```

❻ The %DO loop cycles through &REGCNT iterations. For each iteration the macro variable &I is incremented by 1. %DO loop definitions are terminated with a %END statement.

❼ The value of the i^{th} region will be stored in the macro variable &&REG&I. When &I is 2, this becomes &REG2, which for our example becomes WEST.

❽ The macro definition is terminated with a %MEND statement.

❾ The macro %LINKED is called.

USING FORMATS TO BUILD A LINK

The links can also be established through the use of user defined formats. Using this technique allows you to store the link in a format rather than in the code itself. This approach has the advantage of not having hardcoded links embedded within the code. To change a link all we have to do is change the format.

Here one of the previous examples is rewritten using a format to hold the links that point to the secondary tables.

```
proc format;
   value $regtag ❶
      'WEST' = "<a href='Exercise12_WEST.html'>West</a>"
      'EAST' = "<a href='Exercise12_EAST.html'>East</a>";
   run;

* Regional Report  **********************;
ods html style=default
         path="&path\results" (url=none)
         body='Exercise12_Region.html';

title1 'Region Summary';
footnote1;

proc report data=sashelp.prdsale
                    (where=(prodtype='OFFICE'))
      nowd;
   column region product,actual;
```

```
   define region  / group format=$regtag40. ❷ 'Region';
   define product / across;
   define actual  / analysis sum
                    format=dollar8.
                    'Sales';
   rbreak after / summarize;
   run;
ods html close;
```

❶ The format $REGTAG defines the HTML anchor tags that will be used to form the groups in the PROC REPORT step.

❷ The format is used directly against the grouping variable. Only the *display-text* portion of the formatted value appears in the table.

Region Summary

	Product		
	CHAIR	DESK	TABLE
Region	Sales	Sales	Sales
East	$75,855	$73,616	$74,319
West ❷	$72,425	$75,616	$67,881
	$148,280	*$149,232*	*$142,200*

HTML using the DEFAULT style

SUMMARY

PROC REPORT has the ability to take advantage of not only internal options (STYLE=), but also REPORT step specific statements (CALL DEFINE) to form links between tables. With the additional capabilities of the Output Delivery System, ODS, you can create a series of interconnected tables that utilize sophisticated linking structures.

The power and flexibility of ODS and the REPORT procedure gives the informed SAS programmer a wide selection of tools that can be used to generate linked tables.

ABOUT THE AUTHOR

Art Carpenter's publications list includes four books, and numerous papers and posters presented at SUGI, SAS Global Forum, and other user group conferences. His latest book is a guide to PROC REPORT. Art has been using SAS® since 1976 and has served in various leadership positions in local, regional, national, and international user groups. He is a SAS Certified Advanced Professional™ and through California Occidental Consultants he teaches SAS courses and provides contract SAS programming support nationwide.

AUTHOR CONTACT

Arthur L. Carpenter
California Occidental Consultants
10606 Ketch Circle
Anchorage, AK 99515

(907) 865-9167
art@caloxy.com
www.caloxy.com

REFERENCES

Carpenter, Arthur L., 2007, *Carpenter's Complete Guide to the SAS REPORT Procedure*, SAS Institute Inc., Cary NC.

Carpenter, Arthur L. and Richard O. Smith, 2003, "ODS and Web Enabled Device Drivers: Displaying and Controlling Large Numbers of Graphs", *Proceedings of the Pharmaceutical SAS® User Group Conference (PharmaSUG 2003)*, Cary, NC: SAS Institute Inc., paper TT026. Also in the *Proceedings of the 10th Western Users of SAS® Software Conference (WUSS 2002)*, Cary, NC: SAS Institute Inc.

TRADEMARK INFORMATION