Paper 52-28

A Beginner's Guide to Incorporating SAS® Output in Microsoft® Office Applications
Vincent DelGobbo, SAS Institute Inc., Cary, NC

ABSTRACT

This paper provides techniques for incorporating the output from SAS software, regardless of the install platform, in Microsoft Excel and Word (versions 2000 and later). The paper focuses on the use of the Output Delivery System (ODS) and SAS servers.

INTRODUCTION

After reading this paper you should be able to incorporate just about any output from your SAS code into a Microsoft Excel or Word document. In addition, the data and sample SAS code used in this paper bring to light formatting issues that you may encounter when attempting to import HTML output generated by the Output Delivery System (ODS) into Office documents.

For information about generating other types of output and techniques for using those types of output in Excel and Word, refer to this author's SUGI 27 paper (DelGobbo, 2002).

SAMPLE SCENARIO

Figure 1 shows the SAS output that we will incorporate into both Excel and Word.

Figure 1. ODS-generated HTML output in a Web browser.

The chart in the top half of Figure 1 was generated using PROC GCHART while the table was generated with PROC TABULATE. ODS was used to generate the HTML file that contains both the chart and the table. As we will see later, when you open this HTML file in Excel or Word, it will look very much as it does when viewed in a Web browser (Figure 1).

The SAS data set used in this scenario is a modified version of the data set available in the SAS Sample Library program named

ODSREP2.SAS. The modified data set and associated SAS formats are shown in their entirety in the Appendix.

The high-level steps to place your SAS output into Excel or Word documents are:

1. Run your SAS code, using ODS to generate HTML.
2. Store the HTML output in a network-accessible location.
3. Open the output file(s) in Excel or Word.

The remainder of this paper discusses these steps in detail, and provides solutions to common formatting problems you may encounter.

GENERATING AND STORING THE OUTPUT

The sample PROC GHART and PROC TABULATE code used in this paper can be found in the Appendix. The following sections explain the SAS code and how to store the output file. All code was tested using Version 9 SAS software.

PROC GCHART CODE

The GCHART code used in this example is fairly straightforward. A horizontal, 3D bar chart is generated from the SALES data set. This chart displays wholesale and retail sales information by region for three different products.

The chart is output as a GIF image using the GIF570 SAS/GRAPH® device driver. Both Excel and Word can render GIF images when referenced in HTML files. While you can use the ACTIVEX driver to render the chart using the SAS/GRAPH ActiveX® Control, Excel had problems importing the control at the time of this writing.

STORING THE OUTPUT

When it comes to choosing where to store your output, you have a number of options, including:

*   Local disk (where SAS software is installed)
*   Network accessible disk
*   Web server

There are several good reasons to store your SAS output on a Web server:

1. The files are available to anyone with network access.
2. The HTML files can be accessed by a Web browser in addition to Excel and Word.
3. You can take advantage of Web server authentication and security models.

If you place the files where others can access them over a network, you should set file permissions to prevent accidental alteration.

In our sample scenario, we will store the zODS-generated files on a Web server.

GENERATING THE HTML OUTPUT

Although there are several options that you can specify to control the appearance of your HTML output, the sample code presented in this paper uses only a few of these options.

To generate the HTML output and store it on a Web server, use the following ODS statements:

```
%let HTMROOT=path-to-Web-server/sugi28/;

❶  ods listing close;
❷  ods html path  = "&HTMROOT" (URL="")
            body  = 'salesreport.htm'
            style = Banker;
```

The ODS statement at ❶ turns off the standard "line printer" ODS destination, since we are only concerned with generating HTML output.

The ODS statement at ❷ generates the HTML output and GIF image, and stores them on the Web server. The PATH attribute is used to specify where the HTML output and GIF image will reside. The name of the HTML file will be "salesreport.htm", and the output color scheme is controlled by the "Banker" style.

The value of HTMROOT must point to a subdirectory your Web server can read because this is where the HTML and GIF files will be stored. In this case, the directory "sugi28" is under the main root of the Web server. If you are using Microsoft Internet Information Server (IIS), HTMROOT would typically be c:\inetpub\wwwroot\sugi28\.

Based on the values of HTMROOT and URL, ODS will generate an HTML <IMG> tag with this format:

```
<img src="salesrep.gif">
```

The file name of the GIF image is controlled by the NAME option of the HBAR3D statement (see code in the Appendix).

The "Banker" style is new for Version 9. To see a list of ODS styles that are available for use, submit this SAS code:

```
ods listing;
proc template; list styles; run;
```

You are free to use any style you wish, but Excel or Word may import some styles better than others. Methods for correcting formatting problems you may encounter are presented in the section "Correcting Common Formatting Problems in Excel and Word".

OPENING THE OUTPUT IN EXCEL AND WORD
To open the ODS-generated HTML file with Microsoft Word, follow these steps:

1. In Word select File ➡ Open…
2. In the "File name" field, specify
   http://*Web-server*/sugi28/
   where *Web-server* corresponds to the domain name for your Web server. Click Open. You should see a list of files available on your Web server.
3. Select salesreport.htm and click Open to import the HTML file.

If you don' t see a list of available files after step 2, you may have to configure your Web server to allow directory listing. Alternatively, you can specify the full path to the file you wish to open:

http://*Web-server*/sugi28/salesreport.htm

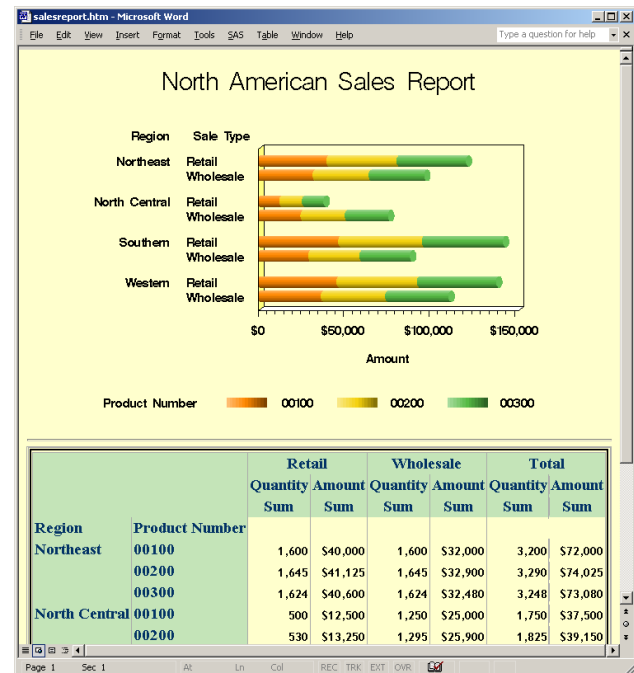When the file is displayed in Word, it should look very similar to Figure 2:

Figure 2. salesreport.htm opened with Word 2002.

Word does a good job of importing the HTML file; it looks very much like the file viewed with a Web browser (Figure 1). However, depending on the version of SAS and MS Office software, you may notice that cell borders are drawn incorrectly, as is the case with Figure 2.

When you follow similar steps to open the file with Excel, you will notice that the background color for some of the cells is gray instead of green, the leading zeros in the product numbers have been dropped, and there are no cell borders (see Figure 3).
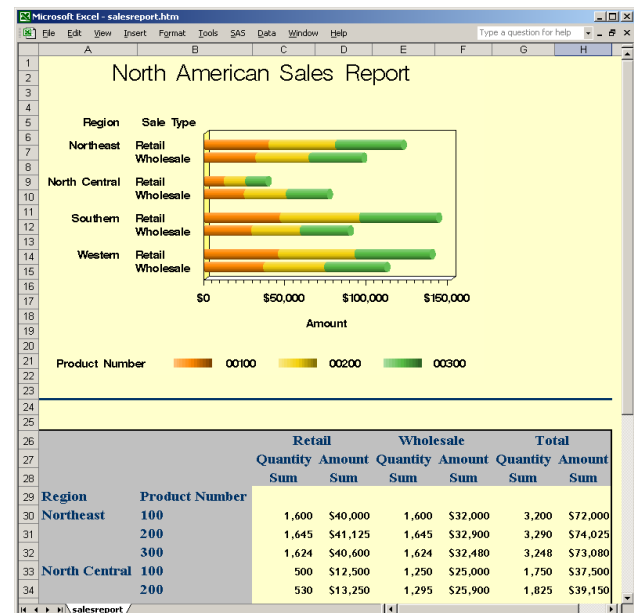

Figure 3. salesreport.htm opened with Excel 2002.

CORRECTING COMMON FORMATTING PROBLEMS IN EXCEL AND WORD

Many formatting problems you encounter when importing HTML output into Excel and Word can be corrected with Cascading Style Sheet (CSS) attributes. Starting in version 8 of SAS software, a number of SAS procedures support the STYLE option. This option can be used to alter the CSS attributes that are generated by SAS procedures and are then embedded in HTML pages. The following sections describe how to use style attributes.

For detailed information on the STYLE option, refer to the SAS Online Documentation ("Available Documentation") for ODS and/or the specific procedure you are interested in. Pass and McNeill (2002a, 2002b) have written excellent papers covering this topic with respect to the Tabulate and Report procedures. Extensive information about CSS is available on the W3C Web site ("Cascading Style Sheets").

CORRECTING CELL BACKGROUND COLOR

The HTML generated by ODS specified #C4E4B8 for the green background color of some of the cells, but, because Excel has a limited color palette that does not support that particular color, Excel mapped the green to gray.

Figure 4. HTML colors supported by Excel 2000 and later.

By examining Figures 1 and 4, you can see that color #CCFFCC is similar to the original green generated by ODS, and can be used as a reasonable substitute. Note that using this new color will slightly change the appearance of the original Web page, as well as the output rendered by Excel and Word.

To generate the ODS HTML using the new color, you must modify the original PROC TABULATE code (available in the Appendix) as shown below. The modifications are shown in bold.

```
proc tabulate data=sales;
  class region product saletype /
    style={background=#CCFFCC};
  classlev product /
    style={background=#CCFFCC};
  classlev region saletype /
    style={background=#CCFFCC};
  var quantity amount /
    style={background=#CCFFCC};
  keyword all sum /
    style={background=#CCFFCC};
  keylabel all="Total";
  table region*product,(saletype=' ' all) *
        (quantity*f=comma7.
         amount*f=dollar8.) /
        box={style={background=#CCFFCC}};
run; quit;
```

The CLASSLEV statement assigns style information to class-level variable values. The ODS style attribute `background` generates the following CSS attribute for all the cells that were gray:

```
background-color: #CCFFCC;
```

ALTERING THE CELL BORDERS

You can see from Figures 2 and 3 that some or all of the cell borders of the table are missing in the Word and Excel documents. You can use the STYLE option to explicitly set the border color and width to values that are supported by Excel and Word. These further modifications to the PROC TABULATE code are shown in bold.

```
proc tabulate data=sales
    style={bordercolor=silver borderwidth=1};
  class region product saletype /
    style={background=#CCFFCC
           bordercolor=silver borderwidth=1};
  classlev product /
    style={background=#CCFFCC
           bordercolor=silver borderwidth=1};
  classlev region saletype /
    style={background=#CCFFCC
           bordercolor=silver borderwidth=1};
  var quantity amount /
    style={background=#CCFFCC
           bordercolor=silver borderwidth=1};
  keyword all sum /
    style={background=#CCFFCC
           bordercolor=silver borderwidth=1};
  keylabel all="Total";
  table region*product,(saletype=' ' all) *
        (quantity*f=comma7.
         amount*f=dollar8.) /
        box={style={background=#CCFFCC
        bordercolor=silver borderwidth=1}};
run; quit;
```

The ODS style attributes `bordercolor` and `borderwidth` generate the following CSS attributes, which are added to all table cells:

```
border-color: silver; border: 1;
```

ASSIGNING EXCEL NUMBER FORMATS

When the HTML file is opened with Excel (see Figure 3), Excel applies the General format to the product numbers, thus dropping leading zeros. This, too, can be corrected with CSS.

Up to this point, we have been indirectly setting CSS attributes through the ODS style attributes. For example, the CSS attribute `background-color` was set by the ODS style attribute `background`.

If you know the name of the CSS attribute you want to set, you can set it directly using the SAS STYLE option attribute HTMLSTYLE. To assign an Excel format, use the proprietary Microsoft CSS attribute `mso-number-format`. You also need to know the definition for the Excel format that you want to assign.

In our case, we want to use the Excel format `00000`, which instructs Excel that the cell will contain a 5-digit number, and that leading zeroes are retained. This is comparable to the SAS Z5. format. To make Excel to use this format when importing HTML,

change the CLASSLEV statement for the variable PRODUCT:

```
classlev product /
   style={background=#CCFFCC
           bordercolor=silver borderwidth=1
             HTMLSTYLE="mso-number-format:00000"
           };
```

This technique is very useful for correcting cases where Excel automatically assigns an incorrect format. Table A-1 in the Appendix shows several common `mso-number-format` settings.

Because the `mso-number-format` is only recognized by Excel, you may use it in any HTML files that you intend to open with other applications, such as a Web browser or Microsoft Word.

Microsoft has published a reference document that describes several of their proprietary CSS extensions that are recognized by Office applications ("Microsoft Office HTML and XML Reference").

For more information about Excel number formats, search for "create a custom number format" in the Excel help system.

Figures 5 and 6 show the final output viewed in Word and Excel. You can see that these two figures are similar to each other and to Figure 1, the original HTML rendered in a Web browser.
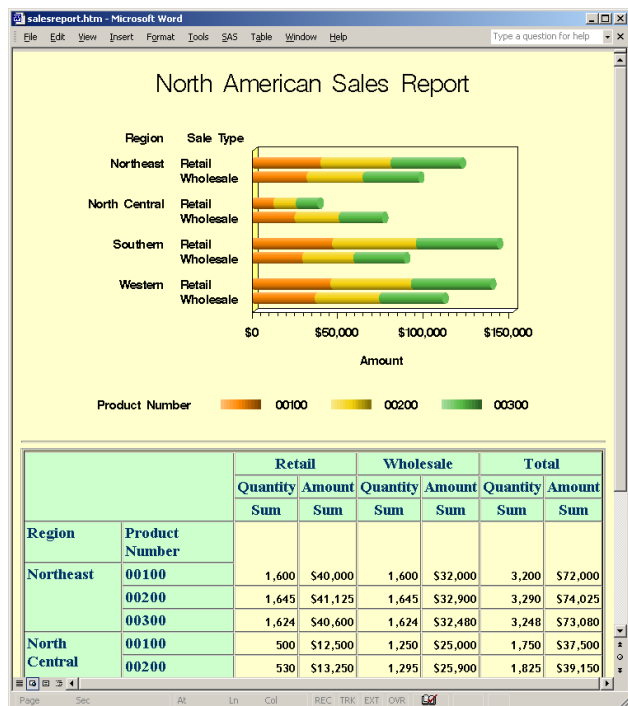
Figure 5. Final HTML file opened with Word 2002.
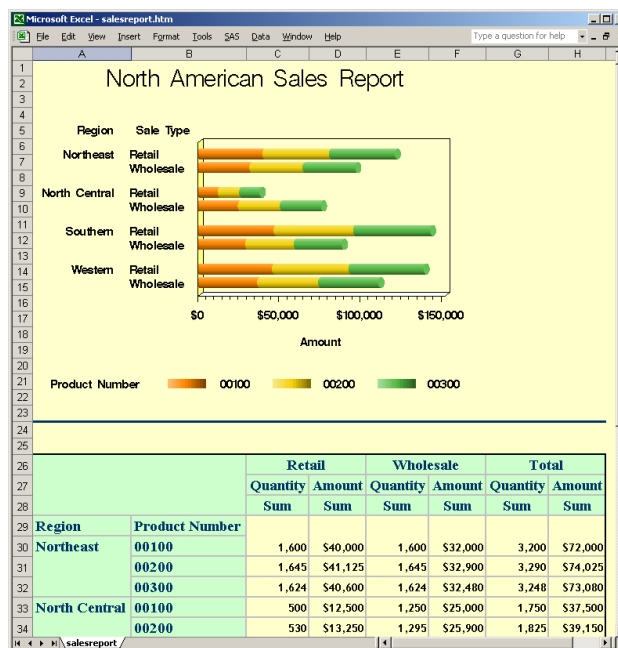
Figure 6. Final HTML file opened with Excel 2002.

CORRECTING OTHER EXCEL FORMAT PROBLEMS
Although we did not encounter them in this example, there are other problems that you may come across when attempting to open HTML files with Excel or Word.

One such problem is where Excel misinterprets data that was meant to be represented as a ratio. Consider the following SAS code:

```
data ratio;
   ratio = '1/4'; output;
   ratio = '2/3'; output;
   ratio = '3/4'; output;
run;
ods listing close;
ods html … ;
   proc print data=ratio noobs;
      var ratio;
   run;
ods html close;
```

Figure 7 (left side) shows that Excel misinterprets the ratios as dates when it imports them.

Figure 7. Excel 2002 misinterprets ratios as dates.

You can correct this problem by making a simple modification to the PROC PRINT code:

```
proc print data=ratio noobs;
   var ratio / style(data)={htmlstyle=
             "mso-number-format:\#\/\#"};
run;
```

Note that since PROC PRINT supports multiple VAR statements, you can use additional statements to print other variables in a data set. This makes it very easy to modify the style attributes of any number of variables.

Another problem you may encounter is when blank spaces used for formatting are "compressed" when your HTML is rendered. This is not due to a problem with Excel or Word, but is rather the way HTML is rendered. Unless text appears between HTML tags that preserve blanks, such as <PRE>, multiple spaces are rendered as 1 space. To correct this problem, use the ASIS attribute:

```
proc print data=your-dataset;
   var your-variable / style={asis=yes};
run;
```

ADVANCED TOPICS
If you have licensed SAS/IntrNet® software, you can dynamically incorporate SAS output into Excel and Word using the Application Dispatcher. You can perform similar tasks with the Stored Process Server, which is new for Release 9.1. In addition, both SAS/IntrNet software and the Stored Process Server can be used with Excel Web Queries to simplify the process.

SAS SERVER TECHNOLOGY
The Application Dispatcher and the Stored Process Server enable you to execute SAS programs from a Web browser or any other client that can open an HTTP connection to either of these SAS servers (which can run on **any** platform where SAS is licensed). The SAS programs that you execute from the browser can consist of any combination of DATA step, PROC, MACRO, or SCL code. Thus, all of the code shown up to this point can be executed using either Application Dispatcher or the Stored Process Server.

Program execution is typically initiated by accessing a URL that points to the SAS server program. Parameters passed to that server program are included as name/value pairs in the URL. The SAS server takes these name/value pairs and constructs SAS MACRO variables with the same name. Thus, any parameter included in the URL is available for use by your SAS program.

For an example, we'll look at the PROC TABULATE code we have been using. If you want a user to specify the cell border color, you first need to modify all the STYLE options to use a variable instead of a specific value (recall that previously the border color was hard coded as "silver"):

```
bordercolor=&BDRCOLOR
```

With this change the color can be specified each time the program is executed. The code is executed on the SAS server via a URL such as:

```
http://path-to-server?_program=program-
name&bdrcolor=RED
```

Now, the value specified in the URL for the parameter named BDRCOLOR will be used by the TABULATE code.

This is just a sample of how you can take advantage of SAS server technologies. For details about the operation of the Application Dispatcher or the Stored Process Server, refer to the respective documentation ("Application Dispatcher" and "Overview of SAS Stored Processes"). Other server-based techniques are discussed in this author's SUGI 27 paper (DelGobbo, 2002).

Up to this point we have entered in the Excel and Word Open dialog boxes a URL that points to a static HTML file. To dynamically create your SAS output and place it into an Excel or Word document, type the server-based URL into the Open dialog box.

EXCEL WEB QUERIES
All of the examples we have looked at so far consist of typing a URL into the Excel or Word Open dialog. This URL can point to static HTML or to a server-based SAS program that dynamically generates HTML. While this procedure works well, Excel has a feature that allows the URL to be placed into a text file so you don't have to type it over and over again. This feature is known as a Web Query.

In Excel 97 and later, Web Queries enable you to incorporate HTML from a Web server, a local drive, or a network drive, directly into your worksheet. To do so you must first create a Web Query file that contains the URL of the HTML you wish to retrieve. In the case of a SAS server program discussed in the previous section, that Web Query file would look something like this:

```
WEB
1
http://path-to-server?_program=program-
name&bdrcolor=RED
```

Note that the SAS server program may be run on any platform, but it must output HTML. Any number of parameters can be passed to the SAS program using the URL. The parameter values can be hard coded or dynamic, enabling you to specify the value each time the query executes. Documentation on creating Web Queries is available from Microsoft ("XL97: How to Create Web Query (.iqy) Files", "Get and Analyze Data from the Web in Excel 2000" and "Getting Data from the Web in Excel 2002").

To run a Web Query, use the appropriate menu selection for your version of Excel:

Excel 97	Data ➥ Get External Data ➥ Run Web Query…
Excel 2000	Data ➥ Get External Data ➥ Run Saved Query…
Excel 2002	Data ➥ Import External Data ➥ Import Data…

You can then navigate to the Web Query file and execute it. The resulting HTML is automatically included in your worksheet.

If you save the workbook and open it at a later date, you can easily retrieve a fresh copy of the HTML (or rerun the SAS server program) by right clicking on a table cell and choosing "Refresh Data".

SAS/ACCESS® TO PC FILES
Starting with Release 9.1 of SAS/ACCESS to PC Files, you have read and write access to Excel workbooks from a UNIX environment. This means that if you license this product on a UNIX platform, you have native access to Excel files. In previous releases, such access was only possible with Windows versions of SAS software. Release 9.1 of SAS/ACCESS to PC Files also includes support for many new file types, as well as enhance-ments to the Import/Export procedure and Wizard. For more information about this product, refer to Plemmons' (2003) SUGI 28 paper.

VISUAL BASIC®
Key and Shamlin (2002) document a number ways you can move data back and forth between SAS and Office using Visual Basic, Visual Basic for Applications (VBA), and ADO. SAS distributes a set of OLE DB providers ("Welcome to the SAS 9 Data Providers: ADO/OLE DB Cookbook") that enable you to implement tools customized to your needs for moving data between Microsoft Office applications and SAS. A sample Microsoft Office addin is

available which illustrates more specifically how such tools can be created ("Overview of the SAS Office Wizard Addin Sample"). This author's paper (DelGobbo, 2002) provides additional techniques for using Visual Basic for Applications with Excel and Word.

FUTURE PLANS

We are currently investigating ways to make SAS content more accessible to Office applications. One such project is determining the feasibility of using ODS to generate XML that is supported by Excel and Word. The benefit would be more robust support for Excel and Word, thus reducing the amount of hand editing you need to do to get the results you desire.

Work is continuing on a product that will provide new ways to access and reuse SAS capabilities directly from Microsoft Office. The first release of this product enables you to

- embed SAS data and analytical results into Office applications
- provide robust data exchange between SAS servers and Office applications
- execute SAS logic running as a SAS Stored Process on remote or local servers
- interact with a SAS server via a user-friendly GUI.

CONCLUSION

Using ODS to generate HTML output is an effective means of incorporating SAS output in Excel and Word documents. Although you may encounter formatting problems when using this technique, you can use ODS and CSS style attributes to overcome many of these problems.

SAS Institute continues to work toward better Microsoft Office integration, and future releases of SAS software will provide more robust means of using SAS content with Office applications.

APPENDIX

Sample SAS Formats and SAS data set

```
proc format;
  value citysize 1 = 'Small'
                 2 = 'Medium'
                 3 = 'Large';

  value region 1 = 'Northeast'
               2 = 'North Central'
               3 = 'Southern'
               4 = 'Western';

  value $saletyp 'R'='Retail'
                 'W'='Wholesale';
run; quit;

data sales;
input region citysize pop product saletype $
      quantity amount;
format region    region.
       citysize  citysize.
       pop       comma7.
       product   z5.
       saletype  $saletyp.
       quantity  comma7.
       amount    dollar8.;

label region   = "Region"
      citysize = "City Size"
      pop      = "Population"
      product  = "Product Number"
```

```
      saletype = "Sale Type"
      quantity = "Quantity"
      amount   = "Amount";
cards;
2 1   25000 100 R 150    3750.00
1 1   37000 100 R 200    5000.00
3 1   48000 100 R 410   10250.00
4 1   32000 100 R 180    4500.00
2 2  125000 100 R 350    8750.00
1 2  237000 100 R 600   15000.00
3 2  348000 100 R 710   17750.00
4 2  432000 100 R 780   19500.00
1 3  837000 100 R 800   20000.00
3 3  748000 100 R 760   19000.00
4 3  932000 100 R 880   22000.00
2 1   25000 100 W 150    3000.00
1 1   37000 100 W 200    4000.00
4 1   32000 100 W 180    3600.00
2 2  125000 100 W 350    7000.00
1 2  237000 100 W 600   12000.00
3 2  348000 100 W 710   14200.00
4 2  432000 100 W 780   15600.00
2 3  625000 100 W 750   15000.00
1 3  837000 100 W 800   16000.00
3 3  748000 100 W 760   15200.00
4 3  932000 100 W 880   17600.00
2 1   25000 200 R 165    4125.00
1 1   37000 200 R 215    5375.00
3 1   48000 200 R 425   10425.00
4 1   32000 200 R 195    4875.00
2 2  125000 200 R 365    9125.00
1 2  237000 200 R 615   15375.00
3 2  348000 200 R 725   19125.00
4 2  432000 200 R 795   19875.00
1 3  837000 200 R 815   20375.00
3 3  748000 200 R 775   19375.00
4 3  932000 200 R 895   22375.00
2 1   25000 200 W 165    3300.00
1 1   37000 200 W 215    4300.00
4 1   32000 200 W 195    3900.00
2 2  125000 200 W 365    7300.00
1 2  237000 200 W 615   12300.00
3 2  348000 200 W 725   14500.00
4 2  432000 200 W 795   15900.00
2 3  625000 200 W 765   15300.00
1 3  837000 200 W 815   16300.00
3 3  748000 200 W 775   15500.00
4 3  932000 200 W 895   17900.00
2 1   25000 300 R 157    3925.00
1 1   37000 300 R 208    5200.00
3 1   48000 300 R 419   10475.00
4 1   32000 300 R 186    4650.00
2 2  125000 300 R 351    8725.00
1 2  237000 300 R 610   15250.00
3 2  348000 300 R 714   17850.00
4 2  432000 300 R 785   19625.00
1 3  837000 300 R 806   20150.00
3 3  748000 300 R 768   19200.00
4 3  932000 300 R 880   22000.00
2 1   25000 300 W 157    3140.00
1 1   37000 300 W 208    4160.00
4 1   32000 300 W 186    3720.00
2 2  125000 300 W 351    7020.00
1 2  237000 300 W 610   12200.00
3 2  348000 300 W 714   14280.00
4 2  432000 300 W 785   15700.00
2 3  625000 300 W 757   15140.00
1 3  837000 300 W 806   16120.00
3 3  748000 300 W 768   15360.00
4 3  932000 300 W 880   17600.00
;
run;
```

Sample PROC GHART and PROC TABULATE Code

```
%let HTMROOT=path-to-Web-server/sugi28/;

ods listing close;
ods html path  = "&HTMROOT" (URL="")
         body  = 'salesreport.htm'
         style = Banker;

goptions reset=all;
goptions device=gif570
         ftext=swissb
         ftitle=swissl
         cback=cxffffcd
         goutmode=replace;
title 'North American Sales Report';
footnote;

pattern1 c=cxffa53d v=solid;
pattern2 c=cxf7df54 v=solid;
pattern3 c=cxc4e4b8 v=solid;

proc gchart data=sales;
  hbar3d saletype / type=sum
                    nostats
                    sumvar=amount
                    subgroup=product
                    group=region
                    discrete
                    shape=cylinder
                    cframe=cxffffcd
                    name='salesrep'
                    noheading;
run; quit;

title; footnote;

proc tabulate data=sales;
  class region product saletype;
  var quantity amount;
  keyword all sum;
  keylabel all="Total";
  table region*product,(saletype=' ' all) *
         (quantity*f=comma7.
          amount*f=dollar8.);
run; quit;

ods html close;
```

Table A-1. Commonly encountered values for mso-number-format.

SAS Data Value	mso-number-format
1234567890.1230	\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#.0000
1,234,567,890.1230	\#\,\#\#\#\,\#\#\#\,\#\#\#\#.0000
1.234.567.890,1230	\#\.#\#\#\.\#\#\#\.\#\#\#\#,0000
$1,234,567,890.30	\$\#\,\#\#\#\,\#\#\#\,\#\#\#\.00
$1.234.567.890,30	\$\#\.\#\#\#\.\#\#\#\.\#\#\#\,00
01340	00000
(919) 677-8000	'\[<=9999999\]\#\#\#\-\#\#\#\#\;\(\#\#\#\\)\ \#\#\#\\-\#\#\#\#'
919-677-8000	\[<=9999999\]\#\#\#\-\#\#\#\#\;\#\#\#\-\#\#\#\- \#\#\#\#
919.677.8000	\[<=9999999\]\#\#\#\.\#\#\#\#;\#\#\#\.\#\#\#\.\#\ #\#\#
16/03/02	dd\/mm\/yy
16/03/2002	dd\/mm\/yyyy
16-03-02	dd\-mm\-yy
16-03-2002	dd\-mm\-yyyy
16.03.02	dd\.mm\.yy
16.03.2002	dd\.mm\.yyyy
03/16/02	mm\/dd\/yy
03/16/2002	mm\/dd\/yyyy
03.16.02	mm\.dd\.yy
03.16.2002	mm\.dd\.yyyy

Note: the letter "V" does not appear in any of the `mso-number-format` definitions. What you are seeing is a combination of a backslash ("\") and a forward slash ("/") characters.

REFERENCES

"Application Dispatcher," SAS Institute Inc. Available http://www.sas.com/rnd/web/intrnet/dispatch.html

"Available Documentation," SAS Institute Inc. Available http://www.sas.com/service/library/onlinedoc/

"Cascading Style Sheets," World Wide Web Consortium. Available http://www.w3.org/Style/CSS/

DelGobbo, V. (2002), "Techniques for SAS® Enabling Microsoft® Office in a Cross-Platform Environment," *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, 27, CD-ROM. Paper 174. Available http://www2.sas.com/proceedings/sugi27/p174-27.pdf

"Get and Analyze Data from the Web in Excel 2000," Microsoft Corporation. Available http://office.microsoft.com/assistance/2000/ExWQA.aspx

"Getting Data from the Web in Excel 2002," Microsoft Corporation. Available http://office.microsoft.com/assistance/2002/articles/GetDataFrom TheWebInExcel.aspx

Key, D. and Shamlin, D. (2002), "Using SAS® Data to Drive Microsoft® Office," *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, 27, CD-ROM. Paper 123. Available http://www2.sas.com/proceedings/sugi27/p123-27.pdf

"Microsoft Office HTML and XML Reference," Microsoft Corporation. Available http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnoffxml/html/ofxml2k.asp

"Overview of SAS Stored Processes," SAS Institute Inc.
Available
http://www.sas.com/rnd/itech/doc9/dev_guide/stprocess/index.ht
ml

"Overview of the SAS Office Wizard Addin Sample," SAS
Institute Inc. Available
http://www.sas.com/rnd/eai/samples/OffWiz/index.html

Pass, R. and McNeill, S. (2002), "PROC TABULATE: Doin' It in
Style," *Proceedings of the Twenty-seventh Annual SAS Users
Group International Conference*, 27, CD-ROM. Paper 189.
Available http://www.ita.doc.gov/td/industry/otea/dcsug/HOW-
TABULATE-STYLE.pdf

Pass, R. and McNeill, S. (2002), "PROC REPORT: Doin' It in
Style," *Proceedings of the Twenty-seventh Annual SAS Users
Group International Conference*, 27, CD-ROM. Paper 187.
Available http://www2.sas.com/proceedings/sugi27/p187-27.pdf

Plemmons, H. (2003), "How to Access PC File Data Objects
Directly from UNIX®," *Proceedings of the Twenty-eighth Annual
SAS Users Group International Conference*, 28, CD-ROM. Paper
156.

"Welcome to the SAS 9 Data Providers: ADO/OLE DB
Cookbook," SAS Institute Inc. Available
http://www.sas.com/rnd/eai/oledb/index.htm

"XL97: How to Create Web Query (.iqy) Files," Microsoft
Corporation, *Microsoft Product Support Services*, March 18,
2002. Available
http://support.microsoft.com/default.aspx?scid=kb;EN-
US;q157482

ACKNOWLEDGMENTS

CONTACT INFORMATION

Your comments and questions are valued and encouraged.
Contact the author at:
      Vincent DelGobbo
      SAS Institute Inc
      SAS Campus Drive
      Cary, NC 27513
      Phone: (919) 677-8000
      http://www.sas.com/rnd/web/