

Using SAS® Output Delivery System (ODS) Markup to Generate Custom PivotTable and PivotChart Reports

Chevell Parker, SAS Institute

ABSTRACT

This paper illustrates how to use ODS markup to create PivotTable and PivotChart reports. You can use these reports to help your organization with business requirements such as analyzing expense trends over time or planning for inventory. Users of all experience levels will learn how to specify options in ODS statements that enable them to perform the following tasks:

- customize PivotTable layouts
- customize cell format
- specify statistics for the analysis
- modify the display of the analysis
- generate a PivotTable report and a PivotChart report for each worksheet
- format tables that you create
- customize worksheets

INTRODUCTION

One of Microsoft Excel's most powerful features is its ability to create PivotTable reports. A *PivotTable report* is an interactive table that enables you to summarize output as well as to change the view of the data dynamically by dragging and dropping the various table components.

PivotTable reports are designed to be very flexible, and analysis is fast because the data is stored in a special cache. In the SAS® System, the closest equivalent to a PivotTable report is what seasoned SAS users call *banner reports*, which are created with the TABULATE procedure. However, PROC TABULATE does not enable you to change the data view without re-running the analysis. You can also add a visual component to your analysis by creating a PivotChart report. A *PivotChart report* provides a visual summary of the data in the corresponding PivotTable report, enabling you to see comparisons and trends more easily.

One of the great benefits of PivotTable reports is that you can easily alter them according to your information preferences. In addition, the data-analysis features of the report make it a powerful tool that is gaining favor with corporate management.

This paper discusses the purpose of PivotTable and PivotChart reports, and it illustrates how you can create customized PivotTable reports using SAS Output Delivery System (ODS) markup language.

THE PIVOTTABLE REPORT: AN INDISPENSABLE BUSINESS TOOL

Common business questions can be difficult to answer if you don't have an effective way to extract and analyze the data. PivotTable reports provide the means to analyze and extract meaning from massive amounts of detailed data, either in a Microsoft Excel workbook or from external files. Some of the more common reasons to use a PivotTable report include the following:

- to summarize lengthy data
- to find relationships within data that are otherwise difficult to see
- to organize data in a visual format that is easy to understand

With these reports, you have full control of how Excel summarizes the data by specifying the statistics or the analysis that you want for the fields in the source data. With a PivotTable report, you can inspect your data in various ways, including rotating (pivoting) rows and columns of data, filtering data, hiding data, and displaying specific details. In addition, data can be broken down in any measurable way. For example, you can look at the data by region to see how a particular product sells within a region or whether this product sells better during a particular month of the year. After you create a PivotTable report, you can visualize that data by creating a PivotChart report. Based on the trends, patterns, and comparisons that are made visible by these reports, you can answer crucial questions about your organization that help you make smart business decisions.

Consider the following scenario. Suppose that you are the CEO of an organization that sells widgets internationally. Each year, you base your company's forecast on historical data. You receive monthly reports that display the sales of

these widgets. You might also receive a separate report for each region and each country. With a PivotTable report, you can quickly analyze your data and extract the following information:

- Which widgets were bestsellers?
- In which month were the best widget sales generated?
- Which region had the best sales?
- Should inventory be increased or decreased for a particular month?

These are typical questions that an organization needs to know in order to make fundamental business decisions. PivotTable reports can be used in any industry. For example, with this type of analysis, a pharmaceutical company can analyze the effects of a certain drug over time. Likewise, an educational institution might want to perform analyses on the success of first-year students compared with their economic backgrounds and use the results as a basis for providing additional support for students entering their first year.

ANATOMY OF A PIVOTTABLE REPORT

You create a PivotTable report in Excel, but the method that you use to create the report varies according to which version of Excel you have. For Excel 2003 and earlier, you created a report using the PivotTable wizard or by dragging fields onto the report. However, the terminology and the default interface for creating PivotTable reports has changed in Excel 2007. In Excel 2007, you select **Insert ► PivotTable**. From the resulting menu, you then select **PivotTable** to open the Create PivotTable dialog box. This dialog box guides you through the creation process. In addition, you build or modify the layout via the **Pivot Table Field List** by dragging and dropping fields within areas of the field list rather than dragging and dropping them into the report, as is done in earlier versions of Excel. If you are more comfortable with the method that you used in the earlier versions of Excel, you have the option to select **Classic Pivot Table Layout** from the PivotTable options menu and continue to use that method in Excel 2007.

Note: Unless otherwise specified, the examples in this paper are based on Excel 2007.

A PivotTable report comprises the following parts:

- **Page area**—a field from the source data that you assign to a page (or filter) orientation in a PivotTable. You can use this field to subset or filter the data so that it reduces the amount of data that is viewed at one time.
- **Data area**—a field from the source data that contains values to be summarized.
- **Column area**—a field from the source data that you assign to a column orientation in a PivotTable report.
- **Row area**—a field from the source data that you assign to a row orientation in a PivotTable report.
- **data items**—the cells in a PivotTable report that contain summarized data.

When you add a row field, you want the table to display data that is organized by the items in this row. If no other layout is specified, you get the items in the list. If you have multiple row fields, one field becomes the inner row field and the other field becomes the outer row field (see [Example 6](#)). In this case, data is organized starting from the outer row field to the inner row field. You can specify the inner and output row fields based on the subtotals that you need.

Adding variables to the **Column** area generates separate columns and uses the variables that are added as column headings for the report. This generates separate variables that display on the column layout.

When you add data to the **Data** area, the data values are summarized based on the row and the column layouts. The values are summarized by default. However, in Excel, you can change the analysis from SUM to a different calculation by clicking the arrow in the **Values** section of the **Pivot Table Field List** in the report. This selection displays the Value Field Settings dialog box. When you add multiple variables to the **Data** area, multiple variables are added to a cell. This behavior enables you to perform an analysis of multiple variables within the same table.

USING THE EXAMPLES IN THIS DOCUMENT TO GENERATE PIVOT TABLES

By default, the examples that are used in this paper create an HTML Web page that can be opened in the SAS Results Viewer window. This page is used as the data source for the PivotTable reports. When any options that are specified to modify Excel output are added to the TableEditor tagset, a button is automatically added to this Web page that enables you to analyze the data further. When you click the button, a PivotTable report is generated based on the options you added. If you do not care about the detailed data, you can click the **Export** button that appears on the Web and export your data directly to Excel. You can also create a temporary Web page (see [Example 1](#)) if you only want to add or create the PivotTable report. The temporary Web page is added to the temporary work area.

The PivotTable reports are created with JavaScript, which adds the ActiveX object. This object enables the Internet Explorer browser to launch Excel. Adding various options from the tagset generates the necessary JavaScript that

modifies the Excel objects. The internal browser that is used by the Results Viewer is an Internet Explorer browser; therefore, every PC user will have access to the viewer.

The sample TableEditor tagset, which is not included with the SAS® System, can be downloaded by submitting the following statements:

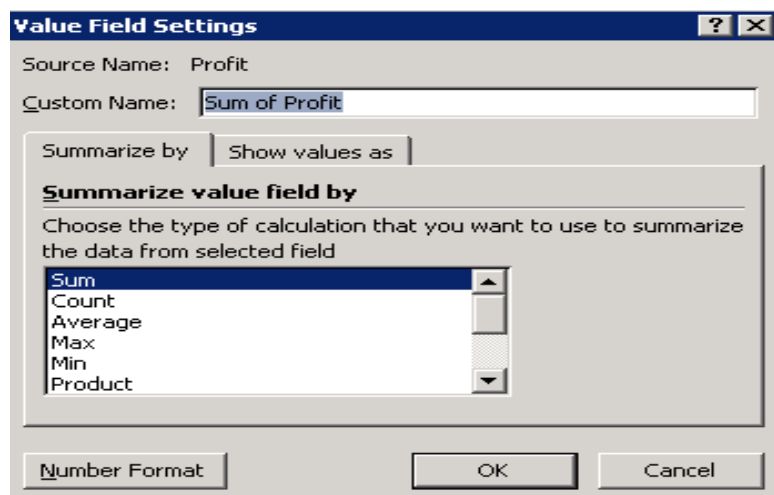
```
filename temp url
'http://support.sas.com/rnd/base/ods/odsmarkup/tableeditor/tableeditor.tpl';
%include temp;
```

You can also download the tagset and the help file directly from the following location:

support.sas.com/rnd/base/ods/odsmarkup/tableeditor/index.html

CREATING REPORTS AND ANALYZING DATA

When you create PivotTable reports in Excel, the columns that you use are specified as pivot fields, which are listed in a docked box (the Pivot Table Field List) to the right with all of the columns in the data range. The layout consists of the page, column, row, and data areas, which you can see in Example 1. The columns in the PivotTable report are dropped into the various areas to create the table. When you add numeric variables to the analysis, the default calculation is to add (SUM) the data. Otherwise, the statistic that is generated is a simple count. You can modify the default statistic by selecting the statistic that you want from the **Summarize value field by** list in the Value Field Settings dialog box, as shown below:



CREATING BASIC PIVOTTABLE REPORTS

The basic PivotTable report can give you a view or snapshot of the overall picture of the financial position of your company. That view can include revenue, expenses, profits, or some other metric that can be measured based on a product line, the region, the year, or some other category.

To create PivotTable reports in Excel (and depending on what version of Excel you have), you use functionality from the Excel toolbar. In Excel, you generate the table by adding data to the **Row**, **Column**, **Page**, or **Data** areas in the layout. To create PivotTable reports with the TableEditor tagset, you add variables to the default layout by using the following predefined options:

- **PIVOTPAGE= option**—creates the report layout.
- **PIVOTROW= option**—enables you to add one or more variables to the row.
- **PIVOTCOL= option**—enables you to add one or more variables to the column.
- **PIVOTDATA= option**—enables you to supply one or more variables to the analysis.

When you create a PivotTable report, the **Pivot Table Field** list is added to the right of the report, by default, with all of the columns in the data source. However, only the column that is added to the layout is displayed in bold type. The values for layout options can be either column names or column positions. However, only the basic reports can be created with the column numbers. For the true functionality, you will want to use the column names.

The following example creates the HTML Web page from which you can later create a PivotTable report:

Example 1

```
ods tagsets.tableeditor file="%sysfunc(getoption(work))\temp.html"
                        options(button_text="Create PivotTable"
                               auto_excel="yes"
                               open_excel="yes"
                               pivotrow="product_line"
                               pivotcol="quarter"
                               pivotdata="profit"
                               pivotpage="year"
                               excel_save_file="c:\\temp.xls"
                               quit="yes");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

This code creates the following HTML Web page, which summarizes a company's profit by product line and by quarter, for all years shown. From this Web page, you can later create the PivotTable report.



Obs	Year	Quarter	Product_Line	Product_Category	Product_Group	Quantity	Profit
721	2002	2002Q1	Sports	Assorted Sports Articles	Assorted Sports articles	4012	138444.65
722	2002	2002Q1	Sports	Assorted Sports Articles	Darts	1226	23131.20
723	2002	2002Q1	Sports	Assorted Sports Articles	Petanque - Boule	411	10018.05
724	2002	2002Q1	Sports	Assorted Sports Articles	Skates	1942	185717.90
725	2002	2002Q1	Sports	Golf	Golf	1654	91135.40
726	2002	2002Q1	Sports	Golf	Golf Clothes	540	41645.90
727	2002	2002Q1	Sports	Indoor Sports	Fitness	514	51669.10
728	2002	2002Q1	Sports	Indoor Sports	Gymnastic Clothing	340	9255.73
729	2002	2002Q1	Sports	Indoor Sports	Top Trim	335	36919.00
730	2002	2002Q1	Sports	Racket Sports	Badminton	125	13589.90

In the detail information that is generated on the Web page, you can see that sporting goods is the company's most profitable product line over the year 2002. Therefore, this is the category in which you want to ensure that you have adequate inventory, and you might want to add other categories within this product line. You might also use the summary to determine the least-profitable lines so that you can decide whether to discontinue those products.

By default, this HTML page contains a button with which you can export your data to a PivotTable report. (See [Example 1](#).) You can customize the text on this button with the `BUTTON_TEXT=` option. In addition, you can send the link for this Web page to other people, who can use it to create the PivotTable report and perform analyses.

The following display shows the PivotTable report that is generated from the Web page:

The screenshot shows an Excel PivotTable report. The PivotTable is located in the range A4:F9. The PivotTable Fields task pane is open on the right, showing the following configuration:

- Report Filter:** Year
- Column Labels:** Quarter
- Row Labels:** Product_Line
- Values:** Sum of Profit

The PivotTable data is as follows:

Product_Line	2002Q1	2002Q2	2002Q3	2002Q4	Grand Total
Children	81139.5	196442.98	221556.12	219163.82	718302.42
Clothes & Shoes	764662.81	1307279.74	1430452.33	1237411.66	4739806.54
Outdoors	663807.35	1129439.67	979360.57	932202.75	3704810.34
Sports	1191087.19	1774849.63	1869300.95	1900773.29	6736011.06
Grand Total	2700696.85	4408012.02	4500669.97	4989551.52	15898930.36

Annotations in the image point to the following areas:

- Page Area:** The top of the worksheet, including the ribbon and the PivotTable Fields task pane.
- Column Area:** The columns of the PivotTable report.
- Row Area:** The rows of the PivotTable report.
- Data Area:** The data cells of the PivotTable report.
- Pivot Table Field List:** The task pane on the right side of the worksheet.

In the previous example code, the `AUTO_EXCEL=` option opens the Excel application when the Web page is loaded, and the `EXCEL_OPEN=` option determines whether the Excel application is visible (`EXCEL_OPEN=YES`) or whether it runs in the background (`EXCEL_OPEN=NO`). By default, the application is visible.

USING DIFFERENT FUNCTIONS IN THE SUMMARY

By default, the SUM statistic is added for the analysis when you add numeric fields to the data area in a report. If you add a character variable or a text variable, a simple count is computed. The value in a report cell represents the results of analyses on variables that are added to the data field. When you modify the statistic in Excel, you must make the change or addition in the Value Field Settings dialog box, which is shown in the section ["Creating Reports and Analyzing Data."](#)

Changing a data view often involves changing the analysis statistic. To change the statistic with the tagset, you use the `PIVOTDATA_STATS=` tagset option. This option enables you to specify a desired statistic. If you have multiple variables in the **Data** area, each column uses this statistic. If you have multiple variables and you want separate statistics for each variable, you must separate each statistic in the `PIVOTDATA_STATS=` option with a comma. If you want two separate statistics, you can add them by specifying the statistics, each separated by comma. You can use any statistic that can be generated by Excel in the `PIVOTDATA_STATS` option.

The available statistics in Excel are as follows:

- **Average**—the average of the data items
- **Count**—the number of data items
- **Countnums**—the number of data items with numeric values
- **Max**—the data item with the largest value
- **Min**—the data item with the smallest value
- **Product**—the result of multiplying the data items
- **StdDev**—the sample standard deviation
- **StdDevP**—the population standard deviation
- **Sum**—the result of adding the data items
- **Var**—the sample variance
- **Varp**—the population variance

The following example uses the same data that is used in [Example 1](#), but it summarizes the profit for the variable `Product_Line`, based on the year, by quarter:

Example 2

```
ods tagsets.tableeditor file="temp.html" options(pivotrow="product_line,
                                                product_category"
                                                pivotcol="quarter"
                                                pivotdata="profit"
                                                pivotpage="year");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

The following display shows the report that is generated, with the average profit as well as the total sum of the profit. It also includes the variable **Product_Category** in the **Row** area to display analysis based on those categories.

	Product_Line	Product_Category	Data	1999	2000	2001	2002	Grand Total
Children			Sum of Profit	541552.09	618803.21	538462.06	718302.42	2417119.78
			Average of Quantity	650.2954545	733.5605556	600.5555556	840.75	716.0284091
Clothes & Shoes			Sum of Profit	412095.45	4739806.54	18097921.42	1609.024306	
			Average of Quantity	145.778	1674.625	1609.024306		
Outdoors			Sum of Profit	2607430.88	33007104	3222098.41	3704810.34	13400513.15
			Average of Quantity	1557.964286	1916.142857	1697.535714	1963.321429	1783.741071
Sports			Sum of Profit	5653876.74	6945429.6	5834176.75	6736011.06	25169494.15
			Average of Quantity	1456.22619	1767.071429	1507.702381	1780.416667	1627.854167
Total Sum of Profit				13272490.83	16159587.27	13754040.04	15898930.36	59085048.5
Total Average of Quantity				1326.802632	1591.969298	1353.438596	1588.131579	1465.085526

CREATING SEPARATE PIVOTTABLE REPORTS FROM A SINGLE SOURCE

There might be times when you want to analyze the data that is created within the PivotTable report separately. You can use the TableEditor tagset and a single data source to do that. For example, you can create separate PivotTable reports that show the minimum, maximum, and average profit, based on the product line, by quarter. As shown below in Example 3, you create individual reports on separate worksheets by specifying the **PIVOT_SERIES="YES"** option and by separating specific parts of the layout for each worksheet with a vertical bar (|).

Example 3

First sheet using
MIN

Second sheet
using MAX

Third sheet
using Average

```
ods tagsets.tableeditor file="temp.html"
options(sheet_name="Profit_Summary"
        pivot_series="yes"
        pivotrow="Product_Line"
        pivotcol="Quarter"
        pivotdata="Profit"
        pivotdata_stats="Min
                        |Max
                        |Average"
        );

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

The **Profit_Summary_pivot_1**, **Profit_Summary_pivot_2**, and **Profit_Summary_pivot_3** reports are generated by this code. The three reports are viewable from individual tabs, as shown in the following display:

	A	B	C	D	E	F
1	Average of Profit	Year				
2	Product_Line	1999	2000	2001	2002	Grand Total
3	Children	12308.00205	14063.70932	12237.77409	16325.055	13733.63511
4	Clothes & Shoes	58189.28667	69583.10306	57756.98361	65830.64639	62840.00493
5	Outdoors	103122.62	128048.9657	115103.5146	132314.655	119647.4388
6	Sports	67308.05643	82683.68571	69454.48512	80190.60786	74909.20878
7	Grand Total	58212.67908	70875.38276	60324.73702	69732.1507	64786.23739

CHANGING FIELD CALCULATIONS IN REPORTS

In PivotTable reports, data can be displayed in many ways. For example, you can display data as a percentage of a row, a column, or the total. You can also generate a running total. To change the field calculations in Excel, you click the arrow in the **Pivot Table Field List** and select the options that you want. This also opens the Value Field Settings dialog box, which lists all of the arguments in Excel that can be modified. As shown in the following display, you might want to show the data as a percentage (in this case, a percentage of a column):

Value Field Settings

Source Name: Profit

Custom Name: Sum of Profit

Summarize by: Show values as

Show values as: % of column

Base field: Obs

Base item:

Number Format OK Cancel

Displaying the data in percentages in a report enables you to make a visual comparison of one instance of a product line to another instance. To achieve the same modification with the tagset, you must use the **PIVOTCALC=** option. Valid values for this option are **Row**, **Column**, **Total**, **PercentOf**, **Index**, and **RunningTotal**. These values are similar to those in the Value Field Settings dialog box in Excel. The following example creates a report that shows a percentage of the column.

Example 4

```
ods tagsets.tableeditor file="temp.html"
    options(sheet_name="Profit_pct"
           pivotrow="product_line" pivotcol="year"
           pivotdata="profit"      pivotdata_stats="sum"
           pivotcalc="column");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```


This program results in the following PivotTable report that now uses the statistic SUM as its calculation for the variable Profit:

Year	1999	2000	2001	2002	Grand Total
Children	4.08%	3.83%	3.91%	4.52%	4.09%
Clothes & Shoes	31.57%	31.00%	30.23%	29.81%	30.63%
Outdoors	21.76%	22.19%	23.43%	23.30%	22.68%
Sports	42.60%	42.98%	42.42%	42.37%	42.60%
Grand Total	100.00%	100.00%	100.00%	100.00%	100.00%

CUSTOMIZING PIVOTTABLE REPORTS

Once you know how to create PivotTable reports that enable you to make decisions based on that data, you will probably want to learn techniques for making your reports more readable. For example, you might want to add number formats to your data, add PivotTable styles, or remove the **Pivot Table Field List**.

FORMATTING PIVOTTABLE REPORTS

There are a variety of reasons why you might want to format your data (for example, for trafficlighting purposes or to display the data in a format that makes sense). When you create PivotTable reports, the cell formatting is not maintained. Therefore, you must provide the formatting for the various fields, most notably, the **Data** area. To display the formatting for any of the areas via Excel, you select a calculation in the Valid Field Settings dialog box and click the **Number Format** button. This action opens the Format Cells dialog box, shown in the following display, from which you can select various types of formatting to apply:

Format Cells

Number

Category:

- General
- Number
- Currency
- Accounting
- Date
- Time
- Percentage
- Fraction
- Scientific
- Text
- Special
- Custom

Sample

Sum of Profit

General format cells have no specific number format.

Value Field Settings

Source Name: Profit

Custom Name: Sum of Profit

Summarize by: Show values as

Summarize value field by

Choose the type of calculation that you want to use to summarize the data from selected field

- Sum
- Count
- Average
- Max
- Min
- Product

Number Format OK Cancel

You can also apply formatting with the PIVOTROW_FMT=, PIVOTCOL_FMT=, PIVOTDATA_FMT=, and PIVOTPAGE_FMT= tagset options. The option names indicate which area of the layout it is used to format. .Currently, you can add only a single format for any of the areas except the **Data** area. You can specify a format for

each field of that area by using the PIVOTDATA_FMT= option. With number formatting, you can do anything from subsetting the data to providing a more descriptive value.

Note: While commas are used to separate multiple variables in the tagset code, individual formats must be separated by a tilde (~), as is illustrated in the next example.

Example 5

```
ods tagsets.tableeditor file="temp.html"
  options(pivotrow="product_line"
         pivotcol="year"
         pivotdata="Profit,Quantity,Total_Retail_Price"
         pivotdata_stats="Sum,Average,Max"
         pivotdata_fmt="[blue] #,###~[red] $#,###.##~$#,###.##"
         pivotpage="year");

proc print data=sashelp.orsales;
run;

ods tagsets.tableeditor close;
```

As shown in the following display, the PIVOTDATA_FMT= option specifies colors as well as the comma and currency characters for the first two data fields, **Profit** and **Quantity**, respectively. The currency format is also specified for the **Total_Retail_Price** data field.

Product_Line	Year	1999	2000	2001	2002	Grand Total
Children	Sum of Profit	\$541,552	\$618,803	\$538,462	\$718,302	\$2,417,120
	Average of Quantity	650.3	733.5	639.57	840.75	716.03
	Max of Total_Retail_Price	\$82,469.83	\$99,103.21	\$79,298.79	\$104,920.01	\$104,920.01
Clothes & Shoes	Sum of Profit	\$4,189,629	\$5,009,983	\$4,158,503	\$4,739,807	\$18,097,921
	Average of Quantity	1,499.33	1,786.24	1,475.9	1,674.63	1,609.02
	Max of Total_Retail_Price	\$686,906.59	\$775,016.01	\$642,071.32	\$730,896.25	\$775,016.01
Outdoors	Sum of Profit	\$2,887,433	\$3,585,371	\$3,222,898	\$3,704,810	\$13,400,513
	Average of Quantity	1,557.96	1,916.14	1,697.54	1,963.32	1,783.74
	Max of Total_Retail_Price	\$606,291.	\$726,506.6	\$666,669.4	\$737,287.5	\$737,287.5
Sports	Sum of Profit	\$5,653,877	\$6,945,430	\$5,834,177	\$6,736,011	\$25,169,494
	Average of Quantity	1,456.23	1,767.07	1,507.7	1,780.42	1,627.85
	Max of Total_Retail_Price	\$940,072.82	\$1,159,837.26	\$962,485.5	\$996,883.02	\$1,159,837.26
Total Sum of Profit		\$13,272,491	\$16,159,587	\$13,754,040	\$15,898,930	\$59,085,049
Total Average of Quantity		1,326.8	1,591.97	1,353.44	1,588.13	1,465.09
Total Max of Total_Retail_Price		\$940,072.82	\$1,159,837.26	\$962,485.5	\$996,883.02	\$1,159,837.26

PIVOTTABLE STYLES

After you create a PivotTable report with the fields that you want to display, you will want to enhance the appearance of the report to improve readability and make the report more visually appealing. *PivotTable* styles enable you to add highlights, styles, and formats to your data. In Excel 2007, you select PivotTable styles from the application's command ribbon. The application also enables you to see dynamically how a style is displayed by moving your cursor over that style. Microsoft has created new styles specifically for PivotTable reports in Excel 2007 and later versions. These styles are categorized in three groups (Light, Medium, and Dark), and there are 13 styles in each group.

To add styles via the tagset, you use the PIVOT_FORMAT= option, as shown Example 6. This option enables you to pass the style directly to the PivotTable report from the tagset. Valid values for the option are LIGHT1–LIGHT13, MEDIUM1–MEDIUM13, and DARK1–DARK13. Older styles (such as REPORT1–REPORT10, TABLE1–TABLE10, and so on) that are added via the **AutoFormat** menu in Excel 2003 or earlier can still be used with the tagset option.

Note: To see a full list of available PivotTable styles, specify the DOC="HELP" option of see the help file that is available in the tagset download. When you specify this option, the list of available options appears in the SAS log.

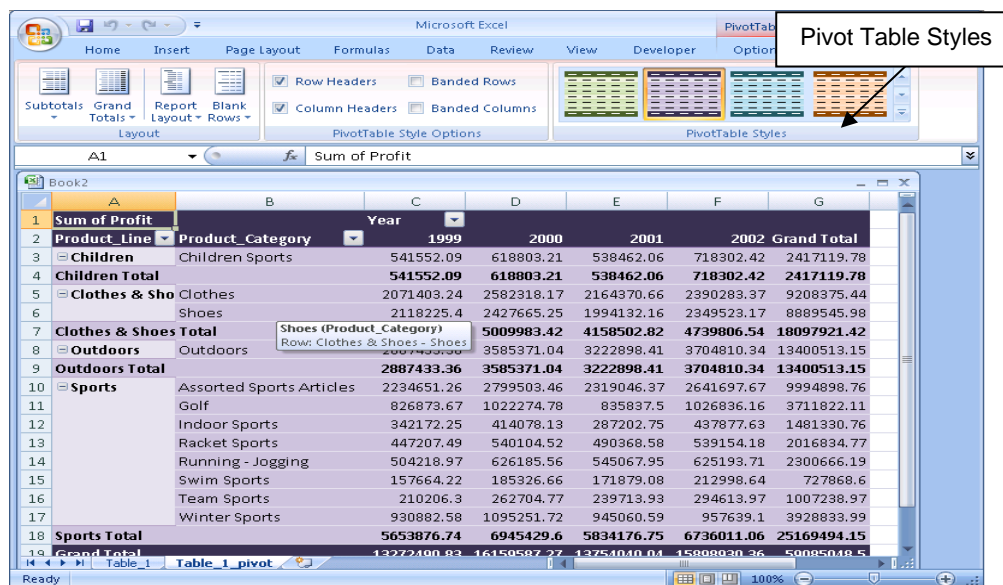
Example 6

```
ods tagsets.tableeditor file="temp.html"
                        options(pivotrow="product_line,product_category"
                                pivotcol="year"
                                pivotdata="profit"
                                pivot_format="dark5");

proc print data=sashelp.orsales;
run;

ods tagsets.tableeditor close;
```

This code generates the following report, using the DARK5 style format:



The screenshot shows an Excel worksheet with a PivotTable. The PivotTable is titled 'Sum of Profit' and is organized by 'Product Line' (Rows) and 'Year' (Columns). The data is displayed in a dark-themed format. The PivotTable includes subtotals for each product line category. A callout box labeled 'Pivot Table Styles' points to the 'PivotTable Styles' task pane on the right side of the Excel window.

Product Line	Year	1999	2000	2001	2002	Grand Total
Children		541552.09	618803.21	538462.06	718302.42	2417119.78
Children Total		541552.09	618803.21	538462.06	718302.42	2417119.78
Clothes & Shoes		2071403.24	2582318.17	2164370.66	2390283.37	9208375.44
Clothes		2118225.4	2427665.25	1994132.16	2349523.17	8889545.98
Shoes						
Clothes & Shoes Total		5009983.42	4158502.82	4739806.54	18097921.42	
Outdoors		3585371.04	3222898.41	3704810.34	13400513.15	
Outdoors Total		2887433.36	3585371.04	3222898.41	3704810.34	13400513.15
Sports		2234651.26	2799503.46	2319046.37	2641697.67	9994898.76
Golf		826873.67	1022274.78	835837.5	1026836.16	3711822.11
Indoor Sports		342172.25	414078.13	287202.75	437877.63	1481330.76
Racket Sports		447207.49	540104.52	490368.58	539154.18	2016834.77
Running - Jogging		504218.97	626185.56	545067.95	625193.71	2300666.19
Swim Sports		157664.22	185326.66	171879.08	212998.64	727868.6
Team Sports		210206.3	262704.77	239713.93	294613.97	1007238.97
Winter Sports		930882.58	1095251.72	945060.59	957639.1	3928833.99
Sports Total		5653876.74	6945429.6	5834176.75	6736011.06	25169494.15
Grand Total		13272400.83	16159587.27	13754040.04	15808030.36	59085018.5

PLACING PIVOTTABLE REPORTS ON THE SAME WORKSHEET AS THE DETAILED DATA

So far, this paper has demonstrated how to place PivotTable reports on a new worksheet within a workbook. However, there might be times when you would like to see the detailed data and the PivotTable data together. For example, suppose you want to view the profit by year and by product line within the same worksheet. You might want the detailed data and the PivotTable report together in order to easily view historical trends that can help you plan for the future.

To place a PivotTable report in the same worksheet as the detailed data in Excel, you select **Existing Worksheet** in the Create PivotTable dialog box. With the TableEditor tagset, you use the PTDEST_RANGE= option to write the data to the currently opened worksheet, as shown in the following example:

Example 7

```
ods tagsets.tableeditor file="temp.html" options(sheet_name="Pivot_example"
                                                  excel_autofilter="yes"
                                                  ptdest_range="g1"
                                                  pivotrow="product_line"
                                                  pivotcol="year"
                                                  pivotdata="profit")
                                                  style=sasweb;

proc print data=sashelp.orsales;
    var product_line year quantity profit;
run;

ods tagsets.tableeditor close;
```

As you can see in the following display, the PTDEST_RANGE= option starts the PivotTable report in column G1:

Obs	Product_Line	Year	Quantity	Pr
1	Children	1999	286	4980.15
2	Children	1999	98	1479.95
3	Children	1999	588	9348.95
4	Children	1999	334	7136.8
5	Children	1999	303	7163
6	Children	1999	755	19153.05
7	Children	1999	209	1975.35
8	Children	1999	14	288.8
9	Children	1999	454	7334.7
10	Children	1999	1243	21847.85
11	Children	1999	139	3020.85
12	Clothes & Shoes	1999	2938	84982.5
13	Clothes & Shoes	1999	171	4706.85

Product_Line	1999	2000	2001
Children	541552.09	618803.21	538462.06
Clothes & Shoes	4189628.64	5009983.42	4158502.82
Outdoors	2887433.36	3585371.04	3222898.41
Sports	5653876.74	6945429.6	5834176.75
Grand Total	13272490.83	16159587.27	13754040.04

WORKING WITH DATA IN EXISTING WORKBOOKS AND WORKSHEETS

The focus of this paper has been on creating PivotTable reports that use SAS as the data source, with an intermediate step in which the data is loaded to a Web page. The Web page populates the PivotTable report. However, you can also use data directly from an Excel file (or any file that can be read by Excel) as the source for populating a report. Using data directly from an Excel file enables you to modify existing workbooks with the newly created PivotTable reports. In addition, you can use the data to update existing workbooks and worksheets. That includes updating any output that is generated via the EXCELXP, CSV, and MSOFFICE2K tagsets or the EXPORT procedure. Traditionally, you only use ODS destinations to write or create new workbooks rather updating existing workbooks. However, you can use either dynamic data exchange (DDE) or the EXPORT procedure to update the workbooks. The tagset opens the workbook or worksheet and modifies the objects that create the PivotTable reports and PivotChart reports in a selected workbook. Because you have access to all Excel objects, you have full control over the output that is you see.

UPDATING EXISTING WORKBOOKS WITH WORKSHEETS AND PIVOTTABLES REPORTS

Because workbooks can be updated, you can append Excel files or any other file that can be opened by Excel. For example, suppose that you create a workbook with the ExcelXP tagset, which uses the same source data mentioned in previous example, to perform a profitability analysis. You need to update the analysis with a PivotTable report or graphic images for your manager.

In the tagset, the UPDATE_TARGET= option specifies the Excel table that you want to update. You specify the worksheet name along with the various layout options that append both the worksheet that is generated from the first table on the Web page and the PivotTable report that uses the worksheet as the data source.

Note: Because the language in the following program is JavaScript, paths that contain backslashes (\) must have double backslashes instead of single backslashes.

Example 8

```
ods tagsets.tableeditorfile="temp.html" options(update_target="C:\\example.xls"
                                                sheet_name="Orsales"
                                                pivotrow="product_line"
                                                pivotcol="year"
                                                pivotdata="profit"
                                                pivot_format="medium5");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

This program results in the following workbook that is updated by the addition of the PivotTable report named **Orsales_pivot**:

Sum of Profit	Year	1999	2000	2001	2002	Grand Total
Children		541552.09	618803.21	538462.06	718302.42	2417119.78
Clothes & Shoes		4189628.63	5009983.42	4158502.82	4739806.53	18097921.4
Outdoors		2887433.36	3585371.04	3222898.41	3704810.34	13400513.15
Sports		5653876.74	6945429.6	5834176.75	6736011.06	25169494.15
Grand Total		13272490.82	16159587.27	13754040.04	15898930.35	59085048.48

UPDATING EXISTING WORKSHEETS

Worksheets Added

At times, you might want to add information directly to an existing worksheet. You can do that by using a combination of the UPDATE_TARGET= option and the UPDATE_SHEET= option.

In the following example, the UPDATE_TARGET= option opens the existing workbook, and the UPDATE_SHEET= option specifies the existing worksheet that you want to modify. You can also write to a specific row or column using either the WORKSHEET_LOCATION= option or the UPDATE_RANGE= option. The WORKSHEET_LOCATION= option writes to a specific location within an existing worksheet or to a new worksheet. The UPDATE_RANGE= option writes to a location on an existing sheet.

Example 9

```
ods tagsets.tableeditor file="temp.html" style=sasweb
  options(update_target="C:\\example1.xls"
    update_sheet="Profit"
    worksheet_location="1,6");

proc print data=sashelp.orsales(obs=10 where=(year=2002));
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor close;
```

This program adds a new PivotTable report to the example1.xls workbook. The report starts in the column 6, row 1 position, as shown in the following display:

The screenshot shows two Excel worksheets side-by-side. The left worksheet, named 'rate.xls', contains data for the year 2001. The right worksheet, named 'profit', contains data for the year 2002. Both worksheets have the same column structure: Obs, Product_Line, Year, and Profit.

Obs	Product_Line	Year	Profit
457	Children	2001	4400.95
458	Children	2001	1384.75
459	Children	2001	7524.55
460	Children	2001	6474.65
461	Children	2001	5439.6
462	Children	2001	18008.5
463	Children	2001	1897.7
464	Children	2001	209.8
465	Children	2001	6618.1
466	Children	2001	18806.5
			70765

Obs	Product_Line	Year	Profit
685	Children	2002	3799.3
686	Children	2002	1617.7
687	Children	2002	9039.05
688	Children	2002	7298.25
689	Children	2002	6247.85
690	Children	2002	19528.55
691	Children	2002	1952.8
692	Children	2002	256.2
693	Children	2002	7370.75
694	Children	2002	21553.55
			78664

USING AN EXCEL WORKBOOK AS A DATA SOURCE

When you use the TableEditor tagset, one of the benefits of using an Excel workbook as your data source is that you eliminate the intermediate step of using the Web page to load the Excel file. As a result, you can use a much larger data source than is possible when the data is loaded from the Web page. You also retain the same flexibility when you create PivotTable and PivotChart reports from CSV, HTML, and Excel files. Although a destination such as ExcelXP can generate only tables, not graphics, you can use the tagset to add PivotChart reports and PivotTable reports and update your output after it is created.

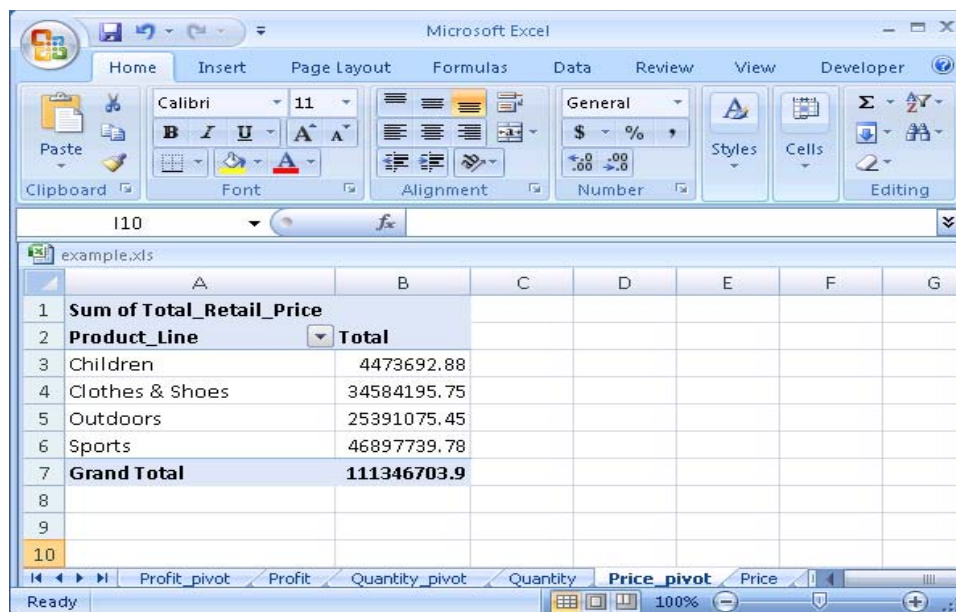
In the following example, the SHEET_NAME= option indicates the worksheets to which you want to append the PivotTable report that was created in Example 9. Notice that the sheet names are separated by commas, while the vertical bar (|)delimits each layout field in the separate sheet. The XML Spreadsheet format that is created from the ExcelXP destination does not create graphics. However, you can add graphics here and save the workbook as a native Excel file. Then, you can import this workbook back into SAS. You can also use the PIVOT_SERIES= option similar to example 3 to create multiple statistics from the same worksheet. All of these tasks normally take a lot of code to accomplish. However, you can accomplish these tasks easily using the tagset options, as shown in Example 10.

Example 10

```
ods tagsets.tableeditor file="temp.html" style=Statistical
  options(update_target="c:\\temp\\testing1.xls"
    sheet_name="Profit, Quantity, Price"
    pivotrow="Product_line |Product_line |Product_line"
    pivotdata="Profit |Quantity |Total_Retail_Price");

data _null_;
  file print;
  put "Create Pivot Tables";
run;
ods tagsets.tableeditor close;
```

The previous code adds three PivotTable reports (**Profit_pivot**, **Quantity_pivot**, and **Price_pivot**) to an existing Excel file, as shown in the following display:



WORKSHEET FORMATTING AND GENERAL OPTIONS

When you are working with PivotTable reports, it is possible to navigate between the reports and the detailed data in their respective worksheets. Therefore, you want to ensure that the detailed data is easy to read and navigate. To make the detailed data more readable, you can modify the following features:

- filters
- frozen headers
- styles
- cell formatting
- tab colors
- header and footer information
- password protection
- page orientation

Selecting a cell from the PivotTable report enables you to toggle back and forth between the PivotTable report and the detailed data on which the report is based. The TableEditor tagset provides a number of options that enable you to format that detailed data (for example, the EXCEL_FROZEN_HEADERS= and PRINT_HEADER= options).

To see a full list of options that are available to the tagset, submit the DOC="HELP" option in the ODS tagsets statement. The tagset download also contains a help file that lists these options and their syntax.

Example 11 modifies a worksheet by adding filters, a frozen header, cell formatting, tab colors, orientation, and a file format for the data. By default, Excel uses the default style unless you either specify another style explicitly or you specify the AUTO_FORMAT= option. In Example 11, the NUMBER_FORMAT= option sets the cell formatting based on columns, which are separated with a vertical bar (|).

Example 11

```
ods tagsets.tableeditor file="temp.html" style=Statistical
    options(excel_autofilter="yes"
           excel_frozen_headers="yes"
           excel_orientation="landscape"
           print_header="Test header"
           auto_format="light13"
           sheet_name="Worksheet"
           number_format="###|@|@@"
           macro="'c:\\worksheetF.xlsm'!Format"
           file_format=xls");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

The example code generates the following worksheet, which adds a table style (Light 13) to the report. Autofilters are added to the column headings, and the headings are frozen. In addition, the code calls a macro from another worksheet to provide formatting to the **Profit**, **Quantity**, and **Total_Retail_Price** columns, and dashboards are added to those columns as well. The number format is added to prevent Excel from using the General format for the data. Finally, the header and page orientation are modified.

The screenshot shows an Excel spreadsheet with the following data (rows 7-28):

Year	Quarter	Product Line	Product Category	Product Group	Quantity	Price	Total Retail Price
1999	1999Q1	Children	Children Sports	N.D. Gear, Kids	755	19153.05	34250.5
1999	1999Q1	Children	Children Sports	Olssons, Kids	209	1975.35	3339.3
1999	1999Q1	Children	Children Sports	Orion Kid's Clothes	14	288.8	580.4
1999	1999Q1	Children	Children Sports	Osprey, Kids	454	7334.7	13219.6
1999	1999Q1	Children	Children Sports	Tracker Kid's Clothes	1243	21847.85	40049.5
1999	1999Q1	Children	Children Sports	Ypsilon, Kids	139	3020.85	5354.7
1999	1999Q1	Clothes & Shoes	Clothes	Eclipse Clothing	2938	84982.5	170206.1
1999	1999Q1	Clothes & Shoes	Clothes	Green Tomato	171	4706.85	7846.2
1999	1999Q1	Clothes & Shoes	Clothes	Knitwear	1554	79951.69	140077.94
1999	1999Q1	Clothes & Shoes	Clothes	LSF	335	16878	32535.5
1999	1999Q1	Clothes & Shoes	Clothes	Leisure	312	14394.3	26047.2
1999	1999Q1	Clothes & Shoes	Clothes	Massif	25	3767.9	7192.1
1999	1999Q1	Clothes & Shoes	Clothes	Orion	793	38066.35	72236.5
1999	1999Q1	Clothes & Shoes	Clothes	Orion Clothing	460	11650.25	22294.3
1999	1999Q1	Clothes & Shoes	Clothes	Osprey	266	9218.75	15712.8
1999	1999Q1	Clothes & Shoes	Clothes	Shorts	150	2278.65	4063
1999	1999Q1	Clothes & Shoes	Clothes	Stockings & Socks	237	2099.55	3564.9
1999	1999Q1	Clothes & Shoes	Clothes	Street Wear	408	15325.3	26391.8
1999	1999Q1	Clothes & Shoes	Clothes	T-Shirts	362	5829.1	10497.1
1999	1999Q1	Clothes & Shoes	Clothes	Tracker Clothes	2624	69039.2	127046.8
1999	1999Q1	Clothes & Shoes	Clothes	Twain	71	9522.58	18878.3
1999	1999Q1	Clothes & Shoes	Clothes	Eclipse Shoes	4121	205449.95	413796.4

POSITIONING WORKSHEETS IN A WORKBOOK

In previous versions of this tagset, it was necessary to request that the tagset generate the multiple worksheets per workbook when multiple tables existed on the source Web page. However, the latest version of the tagset creates multiple worksheets per workbook, by default. If you specify any of the Excel options except SHEET_INTERVAL="NONE", each table on the page is created as a separate worksheet. If you specify SHEET_INTERVAL="NONE", the table is placed on the same sheet and is positioned beneath the prior table. However, you can use the WORKSHEET_LOCATION= option to specify a specific row and cell in which the worksheet should start, as shown in the following example:

Example 12

```
ods tagsets.tableeditor file="temp.html"
                        options(excel_orientation="landscape"
                               worksheet_location="3,2");

proc print data=sashelp.orsales( obs=5 where=(year=1999)) noobs;
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor options(sheet_interval="none" worksheet_location="3,6");

proc print data=sashelp.orsales(obs=5 where=(year=2000)) noobs;
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor options(sheet_interval="none" worksheet_location="11,2");

proc print data=sashelp.orsales(obs=5 where=(year=2001)) noobs;
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor options(sheet_interval="none" worksheet_location="11,6");

proc print data=sashelp.orsales(obs=5 where=(year=2002)) noobs;
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor close;
```


As you can see in the following display, the worksheets are positioned according to the locations that are specified in the WORKSHEET_LOCATION= option. The first worksheet is positioned in column 2, row 3, while the second worksheet is located in column 7, row 3.

Product Line	Year	Profit
Children	1999	4980.15
Children	1999	1479.95
Children	1999	9348.95
Children	1999	7136.8
Children	1999	7163
		30108.85
Children	2001	4400.95
Children	2001	1384.75
Children	2001	7524.55
Children	2001	6474.65
Children	2001	5439.6
		25224.5

Product Line	Year	Profit
Children	2000	5627.45
Children	2000	1618.5
Children	2000	9501.15
Children	2000	7508.55
Children	2000	7193.2
		31448.85
Children	2002	3799.3
Children	2002	1617.7
Children	2002	9039.05
Children	2002	7298.25
Children	2002	6247.85
		28002.15

PROVIDING WORKSHEET TEMPLATES FOR FORMATTING

You can use Excel templates to provide consistent formatting for Excel worksheets. For example, suppose you have a template that is used company-wide for formatting. You specify that template with the WORKSHEET_TEMPLATE= option. As illustrated in Example 13, you can also use this option in conjunction with the WORKSHEET_LOCATION= option to specify where the tagset should begin writing the data. The data can originate from one or more data sources that enable you to write to the template. The data can contain embedded formulas, corporate images, or any other formatting that is necessary to generate the look that you want.

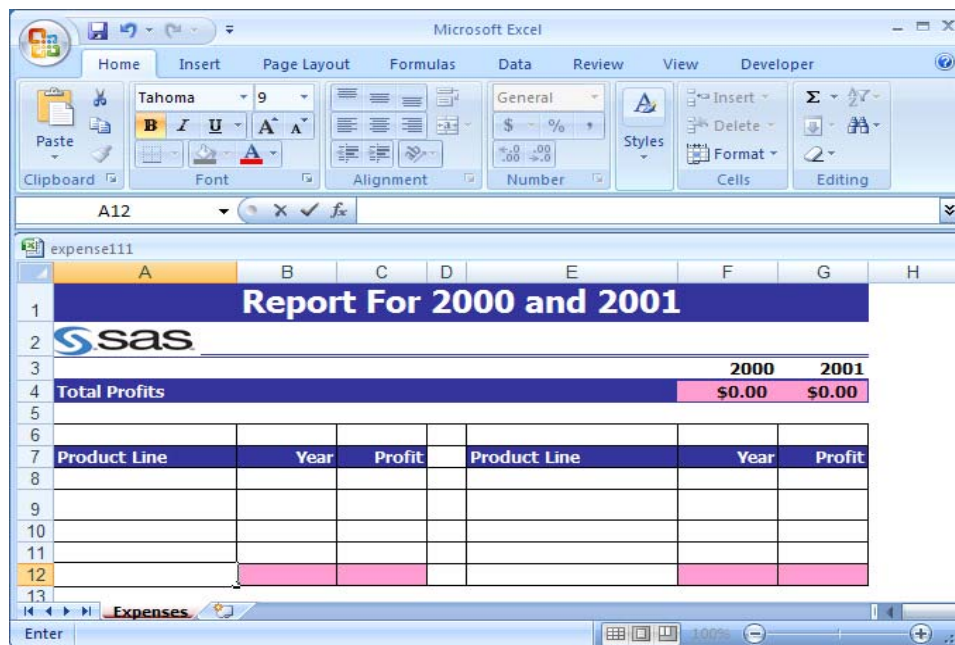
Example 13

```
ods tagsets.tableeditor file="temp.html"
                        options(worksheet_location="7,1"
                               worksheet_template="c:\\expense.xltx");

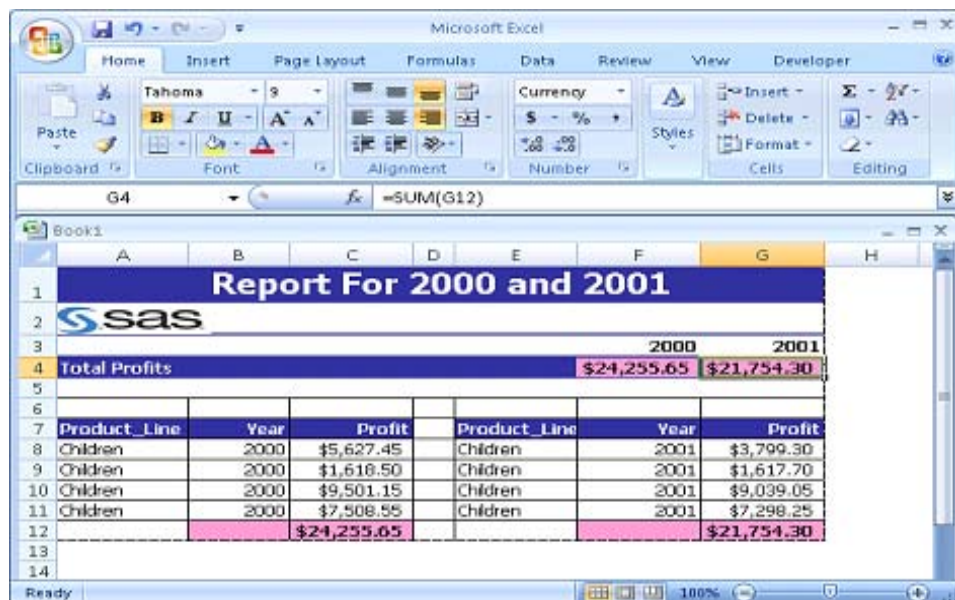
proc print data=sashelp.orsales(obs=4 where=(year=2000)) noobs;
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor options(sheet_interval="none" worksheet_location="7,5");

proc print data=sashelp.orsales(obs=4 where=(year=2001) noobs;
  var product_line year profit;
  sum profit;
run;
ods tagsets.tableeditor close;
```

The following display illustrates how to use the tagset to incorporate a worksheet template when you add a new worksheet. In this case, the tagset uses the **Expenses** worksheet template, which does not contain any data.



The code uses the WORKSHEET_TEMPLATE= option to write the contents of two separate procedures to a new workbook. The SHEET_INTERVAL= option enables you to place the data from the second procedure on the new worksheet as well. The template contains formulas that are used for calculations in the new worksheet.



CREATING PIVOTCHART REPORTS

PivotChart reports are the graphical representations of PivotTable data. Like the PivotTable report, a PivotChart report enables you to visually inspect the data, from which you can make better-informed decisions. Using Excel or the tagset, you can rearrange the layout, change the chart type, and add or remove data from a PivotChart report. Creating PivotChart reports is easy, because the data is already in a format that lends itself to charting. Some features, such as the ability to filter a chart based on the layout fields, is inherent to PivotChart reports. It is also possible to modify the underlying PivotTable report by modifying the chart.

With the TableEditor tagset, you can use the CHART_TYPE= and the PIVOTCHARTS= options to create charts. Other options are available to make your charts more meaningful, but these two particular options, along with the layout fields in the PivotTable reports, generate default charts.

If you use the option `PIVOTCHARTS="YES"`, the application assumes that you have already created a PivotTable report, and it creates a PivotChart report based on that table. By default, the naming convention for PivotChart reports is to start with the name of the worksheet (where your detailed data is located) and append `_Chart` (for example `Orsales_Chart`).

Note: If you modify the layout of a PivotTable report, the layout of the corresponding PivotChart report is also modified.

A PivotChart report comprises the following parts:

- **plot area**—an area where the actual data is displayed in a chart.
- **data series**—an element that corresponds to one related group of numbers in the PivotChart report. For example, it can refer to a column of numbers.
- **category axis**—the axis that lists the values of the data categories.
- **data series axis**—the axis that identifies the individual data series.
- **value axis**—the axis that displays a scale of values for the data points.
- **chart title**—the title of the chart.
- **axis labels**—labels for the individual axes.
- **legend**—the key that identifies a data series by color or pattern.

ENHANCING YOUR PIVOTCHART REPORTS

Excel provides many chart selections from which you can choose. The TableEditor tagset offers the same charts for creating PivotChart reports, including charts that contain both single and multiple parameters. To see examples of all charts that are available from the tagset, specify the `DOC="HELP"` option in the ODS statement. All the valid chart types are output to the SAS log. The Help file that is available when you download the tagset also includes the available charts types.

As shown in Example 14, the `PIVOTCHARTS="YES"` option creates the PivotChart report. If you omit this option, the tagset adds a regular chart. By default, a new PivotChart report is added to a new chart sheet.

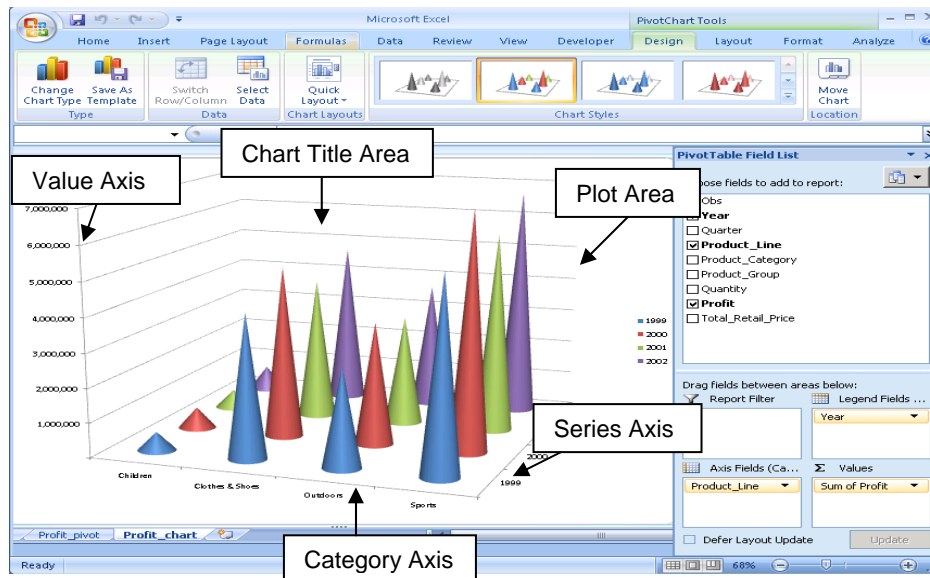
The following example code creates a PivotChart report that is based on the Profit data:

Example 14

```
ods tagsets.tableeditor file="temp.html" options(pivotrow="product_line"
                                              pivotcol="year"
                                              pivotdata="profit"
                                              pivotdata_fmt="#,###"
                                              pivotcharts="yes"
                                              chart_type="conecol");

proc print ata=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

The `Profit_Chart` chart that is generated uses cones for the columns (`CHART_TYPE="CONECOL"`) and shows some of the parts that make up a PivotChart report:



CHANGING CHART AND AXIS TITLES

With PivotChart reports, you can display a title for the overall chart and for each of the axes within the chart. Chart and axes titles make the chart more descriptive. Excel also provides options that modify the color and size of chart and axes titles. To access these options in Excel, you select **Layouts ► Axis Titles** from the menu bar.

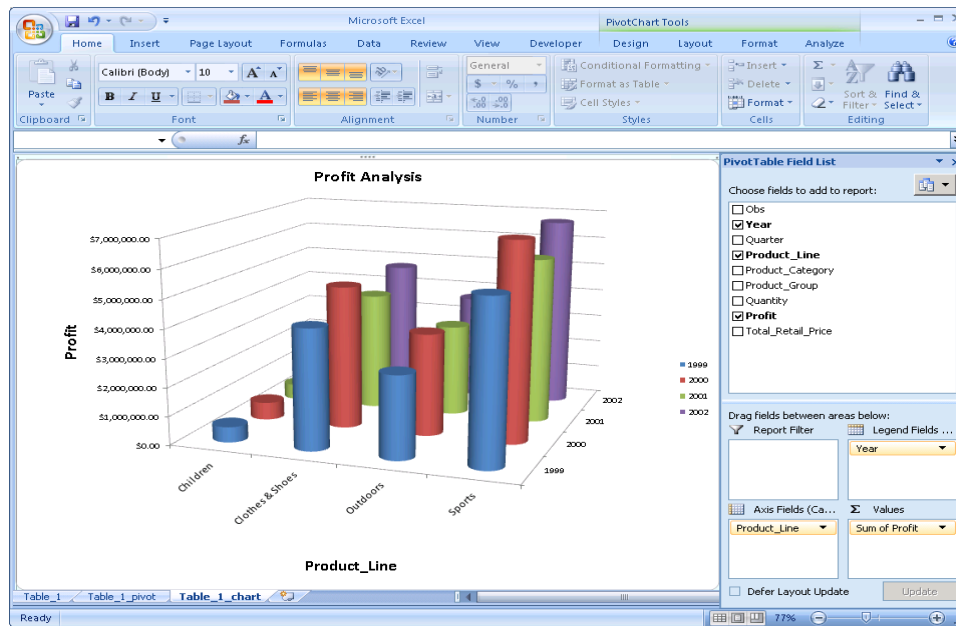
In the tagset, you modify chart and axes titles with the `CHART_TITLE=`, `CHART_XAXES_TITLE=`, and the `CHART_YAXES_TITLE=` options, as shown in Example 15. The tagset also provides options for modifying the color and size of the titles. In addition, you can modify the orientation and number format by using the `CHART_YAXES_ORIENTATION=` and the `CHART_YAXES_NUMBERFORMAT=` options. Other options are also available in the tagset that enable you to modify the scale of axes.

Example 15

```
ods tagsets.tableeditor file="temp.html"
  options(pivotrow="product_line"
         pivotcol="year"
         pivotdata="profit"
         pivotdata_fmt="#,###"
         pivotcharts="yes"
         chart_type="cylindercol"
         chart_title="Profit Analysis"
         chart_yaxes_title="Profit"
         chart_xaxes_title="Product_line"
         chart_xaxes_orientation="45"
         chart_yaxes_numberformat="currency");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

The PivotChart report that is generated by this code uses cylinders as the column indicator (`CHART_TYPE="CYLINDERCOL"`), and it adds a chart title (**Profit Analysis**) in addition to the X-axis and Y-axis titles (**Profit** and **Product_Line**, respectively).



MODIFYING THE APPEARANCE OF PIVOTCHART REPORTS

Both Excel and the tagset enable you to modify the appearance of a PivotChart report. For example, you can add data labels to the chart and reposition the legend. Within Excel, you select **Layout ► Data Labels** to modify the appearance of the display.

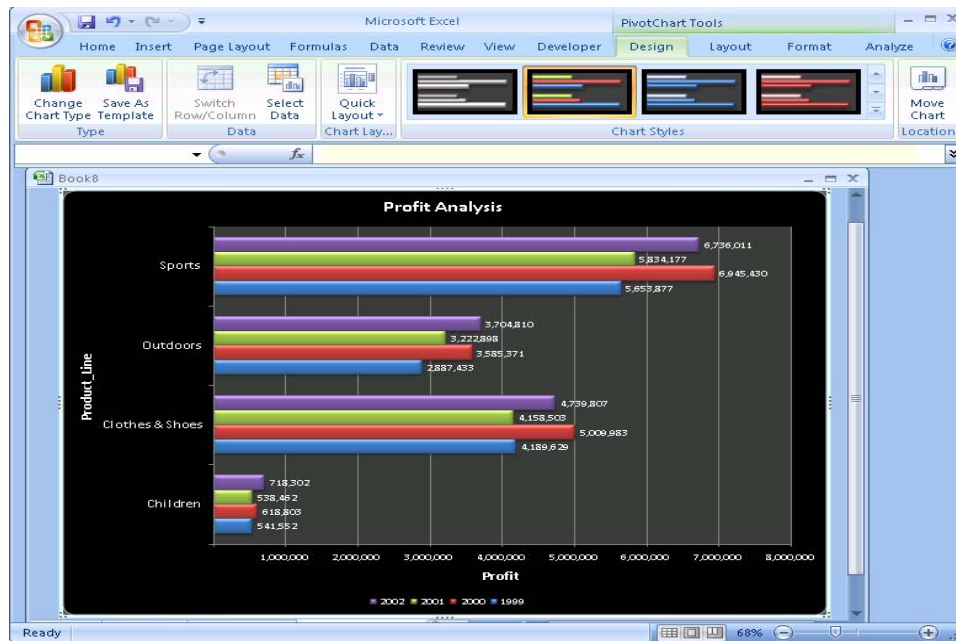
To modify the appearance of the display using the tagset, you use the chart options. For example, to add data labels, you use the `CHART_DATALABELS=` option, as shown in Example 16. Valid values for the `CHART_DATALABELS=` option include `VALUE`, `PERCENT`, `LABELANDPERCENT`, `SHOWBUBBLESIZES`, and `YES` (which has the same result as `VALUE`). Excel 2007 now enables you to select styles from the ribbon. You can add styles with the tagset using the `CHART_STYLE=` option. The values range from 1–48. In the following example, notice that the chart legend has been repositioned using the `CHART_LEGEND=` option.

Example 16

```
ods tagsets.tableeditor file="temp.html" options(pivotrow="product_line"
                                              pivotcol="year"
                                              pivotdata="profit"
                                              pivotdata_fmt="#,###"
                                              pivotcharts="yes"
                                              chart_type="barclustered"
                                              chart_title="Profit Analysis"
                                              chart_yaxes_title="Profit"
                                              chart_xaxes_title="Product_Line"
                                              chart_style="42"
                                              chart_legend="bottom"
                                              chart_datalabels="value" );

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

As you can see in the following report, adding data labels enhances your report even further:



EMBEDDING PIVOTCHART REPORTS IN A WORKSHEET

You might find it helpful to see both your PivotChart report and your PivotTable report together on the same worksheet. In Excel, you can use the Move Chart dialog box (available on the **Design** tab) to embed PivotChart reports in a worksheet rather than creating a new chart from the PivotTable report.

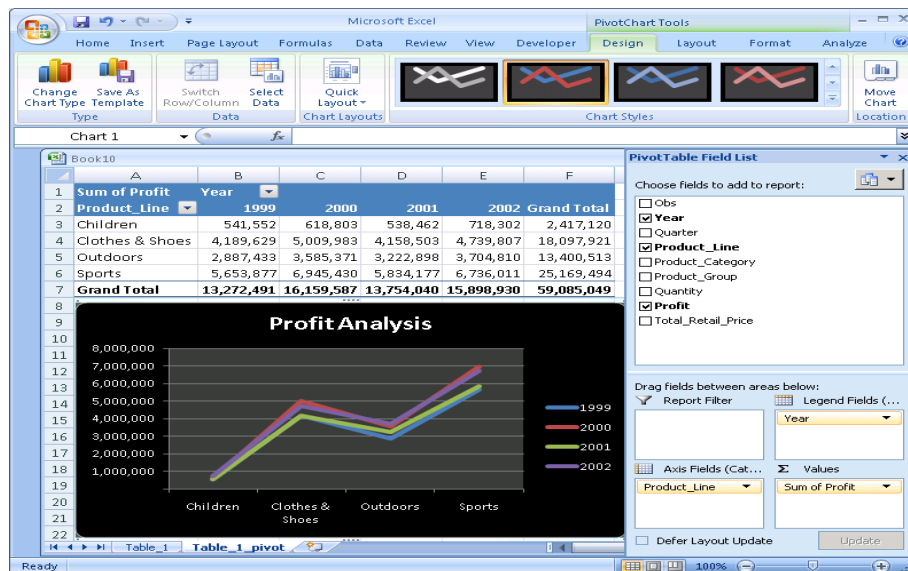
As shown in Example 17, you can use the tagset to embed a PivotChart report in a worksheet by using the CHART_LOCATION="SAME_SHEET" option. This value can be the name of the worksheet.

Example 17

```
ods tagsets.tableeditor file="temp.html"
                        options(pivotrow="product_line"  pivotcol="year"
                                pivotdata="profit"        pivotdata_fmt="#,###"
                                pivotcharts="yes"          chart_type="line"
                                chart_style="42"
                                chart_location="same_sheet");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

The PivotChart report is embedded in the worksheet along with the corresponding PivotTable report, as shown in the following display:



POSITIONING PIVOTCHART REPORTS IN A WORKSHEET

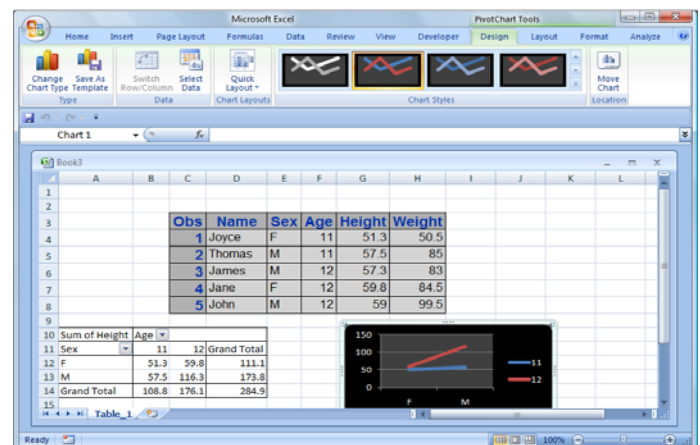
If you add the PivotTable report and the PivotChart reports on the same worksheet, you might want to position both the report and the chart to the location of your choice. By adding the PDEST_RANGE= option, you can define the location that the PivotTable report should begin writing. Using the CHART_POSITION= option, you can specify the left, top, height and width of the embedded image. This gives you full control over the placement of the data.

As shown in Example 18, you can use the tagset to embed a PivotChart report in a worksheet as well as position both the image and the PivotTable report.

Example 18

```
ods tagsets.tableeditor
file="temp.html"
options(pivotrow="sex"
       pivotcol="age"
       pivotdata="height"
       pivotcharts="yes"
       worksheet_location="3,3"
       ptdest_range="a10"
       chart_style="42"
       chart_source="a10"
       chart_position="150,275,100"
       chart_location="Table_1"
       chart_type="line");
```

```
proc print data=class(obs=5);
run;
ods tagsets.tableeditor close;
```



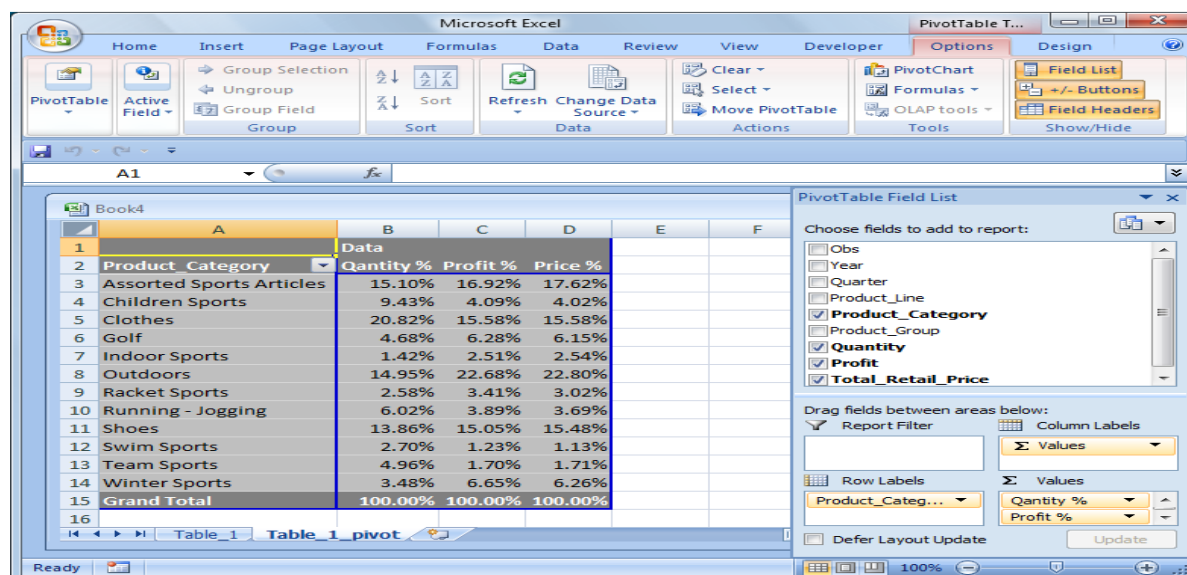
ADVANCED AND CUSTOMIZED LAYOUTS

We can generate advanced layouts by labeling columns in the data area to be more descriptive and meaningful. Labeling the values in the data area is done using the PIVOTDATA_CAPTION= option. The PIVOTDATA_STATS= option can be used to supply a function. We can also remove subtotals and grand totals from the report using the PIVOT_SUBTOTALS= and the PIVOT_GRANDTOTAL= options. Other things that we might want to do is to move multiple data values from the row to the column. This is done using the PIVOTDATA_TOCOLUMNS= option. What we can do in Excel, we should have the ability to do here. Example 19 highlights some of the options that will modify the default layouts.

Example 19

```
ods tagsets.tableeditor file="temp.html"
options(pivotrow="product_category"
       pivotdata="quantity,profit,total_retail_price"
       pivotdata_caption="Quantity %,Profit %, Price %"
       pivotdata_stats="sum,sum,sum"
       pivotcalc="column"
       pivot_format="dark1"
       pivot_subtotals="no"
       pivotdata_tocolumns="yes");

proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```



Product Category	Quantity %	Profit %	Price %
Assorted Sports Articles	15.10%	16.92%	17.62%
Children Sports	9.43%	4.09%	4.02%
Clothes	20.82%	15.58%	15.58%
Golf	4.68%	6.28%	6.15%
Indoor Sports	1.42%	2.51%	2.54%
Outdoors	14.95%	22.68%	22.80%
Racket Sports	2.58%	3.41%	3.02%
Running - Jogging	6.02%	3.89%	3.69%
Shoes	13.86%	15.05%	15.48%
Swim Sports	2.70%	1.23%	1.13%
Team Sports	4.96%	1.70%	1.71%
Winter Sports	3.48%	6.65%	6.26%
Grand Total	100.00%	100.00%	100.00%

CONCLUSION

Creating customized PivotTable and PivotChart reports is easy with SAS ODS markup language. Whether you need a quick report or a heavily styled and formatted report, you can do it all with minimal effort using the TableEditor tagset that was created with ODS markup language. This paper provided only a glimpse of the formatting capabilities that are available. However, the tips offered here should provide a "pivotal" turning point in how you customize your future applications!

REFERENCES

Parker, Chevell. 2008. "Creating That Perfect Data Grid Using the SAS® Output Delivery System." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/forum2008/258-2008.pdf.

Parker, Chevell. 2007. "Creating a Data Grid Like VB.NET" (includes the TableEditor tagset download). Cary, NC: SAS Institute Inc. Available at support.sas.com/rnd/base/ods/odsmarkup/tableeditor/index.html.

SAS Institute Inc. 2008 "ODS MARKUP Resources". Cary, NC: SAS Institute Inc. Available at support.sas.com/rnd/base/ods/odsmarkup/.

SAS Institute Inc. 2006. SAS Note 17890. "An error may occur when there is a long list of options specified for the ODS statement." Cary, NC: SAS Institute Inc. Available at support.sas.com/kb/17/890.html.

ACKNOWLEDGMENTS

I would like to thank the ODS development group for continually producing the great software that was discussed in this paper. I would also like to thank the Base and Graph group in SAS Technical Support as well as Susan Berry, SAS Technical Support Writing and Editing.

CONTACT INFORMATION

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

Chevell Parker
SAS Institute
SAS Campus Drive
Cary, NC 27513
E-mail: chevell.parker@sas.com
Web: support.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.