

SAS®-with-Excel Application Development Tools and Techniques, Expanded Edition

LeRoy Bessler, Assurant Health, Milwaukee, Wisconsin, USA

Abstract

The commonest non-SAS tool for data presentation, and working with data, is Microsoft Excel. Other people may want your SAS report formatted as a spreadsheet. Dynamic Data Exchange (DDE) can empower and augment your use of SAS with Excel. You can develop SAS programs to load worksheets, and to format them without point-and-click. Almost anything that you can do directly in Excel can also be done using DDE commands from your SAS program. You can access Excel data, even to the level of a specific cell, row or column, part of a row or column, or set of contiguous rows or columns, or can work with an entire worksheet. Then you can write, read, or format data. Why would you want to run Excel from SAS? You can create a hands-off production job to be automatically scheduled by the computer to access and analyze the data, load the spreadsheet, format it, and attach the report to a SAS-dispatched email message. In any case, if you find yourself preparing the same Excel report over and over, why not automate it? The intended audience is all levels of SAS users.

Introduction

This paper and its companion slide presentation are an introduction to a SAS-with-Excel self-study collection of sixty macros and twenty-six sample programs (with substantial comments), and supporting files, which are available at no charge from the author via email (at bessler@execpc.com).

To find out more about the Excel commands used here to work with DDE, or to find other commands or command options not used here, you must consult the documentation in Macrofun.hlp. To get that information, you must install Macrofun.hlp by initially downloading Macrofun.exe from the Microsoft Download Center, and then run the .exe file. If you are not experienced using the Download Center, you can find the right part of it by using google.com to search for “macrofun.exe”, or instead you could try this URL: <http://support.microsoft.com:80/support/kb/articles/Q128/1/85.ASP>.

NOTE: The Appendices contain information not included in the original version of this paper, which was limited to 20 pages for publication. See Appendix 1 for usage cautions, Appendix 2 for a ways to refresh pivot tables, Appendix 3 for ways to insert (and resize) graphs, either standalone on separate worksheets, in groups on a worksheet, or with a companion table, and Appendix 4 for two ways to do subtotals.

DDE: Dynamic Data Exchange

Client-Server Architecture

Your SAS session is a client. It opens Excel, which acts as a server. Your SAS program:

1. Starts/Executes Excel
2. Establishes a channel to talk to Excel
3. Opens a new/existing XLS file (workbook)
4. Sends commands to Excel
5. Saves, or Saves As (new or different), XLS file
6. Exits/Stops Excel

What Can You Do with DDE?

Anything that you might do in Excel with your keyboard and mouse! Here I cover only a subset of the input/output and formatting possibilities. When DDE cannot do it (rare), you can pre-record an Excel macro and run it with DDE.

Some of the Cell Formatting Options

- Font controls available include: font, style, size, color, and underline.
- Cell background can be filled with color.
- Cell sizing is possible, and column or row AutoFit and column or row Hide/Unhide.
- Cell alignment, cell merge, text wrap, etc. can be done.
- Borders can be created, e.g., to section the worksheet.
- Cell content can be indented, using two tricks shown in sample programs.

Some of the Worksheet Controls

- Select worksheet, cells, rows, columns
- Freeze/Unfreeze panes
- Turn AutoFilter On/Off
- Generate subtotals in a worksheet
- Control/Change levels for Excel subtotals (or create multiple custom sheets for various levels of totals)
- Activate a specific worksheet in a workbook
- Create additional worksheets in a workbook
- Delete unused worksheets in a workbook

Some of the Other Manipulation Possibilities

- Insert/Delete columns or rows
- Find, Find and Replace cell contents
- Copy cells, Move cells
- Use formulas to create new column(s)
- Use Zoom to shrink or enlarge a worksheet
- Control the Message text box in the lower left corner of the Excel window

The Mechanics of DDE and the Basic SAS-with-Excel Interaction

Channel for DDE Commands

DDE commands require use of special SAS filename statements:

```
filename YourFileRef dde "excel|system";
```

YourFileRef is arbitrary 1-to-8 characters. I like **DDEcmds**.

FILENAME Statement Assignments to Read or Write Data with DDE

Identification of the current selection of worksheet and a rectangle of cells for read or write is accomplished with:

```
EXCEL|YourSheet!RpppppCqqq:RsssssCttt
```

ppppp is 1-to-5-digit start/top row number

qqq is 1-to-3-digit start/left column number

sssss is end/bottom row number

ttt is end/right column number

Row/column numbers with leading zeros are permissible.

Other ways to make selections of cells are as follows:

RpppppCqqq - single cell
Rppppp:Rsssss - range of rows
Rppppp - single row
Cqqq:Cttt - range of columns
Cqqq - single column

See the section “Insert a Heading for the New Column 4” for a simple example of the FILENAME statement used for writing data with DDE. For more examples of writing data to Excel from SAS, see Reference 1.

Coding the Cell Selection

You are not required to permanently hard code it. You can use DDE select functions to operate based on criteria assigned by you (as in some examples below). Or you can use macro program code to determine the selection target dynamically (as in examples from the tools package available via email).

Excel DDE Commands

Microsoft developed DDE commands for an earlier version of Excel than the current one. I have used them with various versions of Excel and Windows. Almost all of the commands (that I have tried) work for me, but I can offer no guarantees.

The commands are documented in Macrofun.hlp, which you can download from Microsoft as mentioned above. “Macrofun” is an abbreviation for “Macro Functions”, not “Macro Fun”, but these macros functions are fun to use. They do not require use of Excel macros.

Please be aware that not every DDE command works (e.g., rename worksheets, create subtotals, . . .), and some command options may not work. Nevertheless, what *does* work is an enormous tool set!

true / false Values for DDE Commands

These are used to turn options On / Off in DDE command parameter assignments. For some commands you are required to use 1 / 0 instead.

Basic DDE Functions

1. Start Excel: provides a default empty workbook, with three worksheets
2. Open an existing workbook
3. Save As of the current workbook with a new name
4. Exit Excel

All of the examples presented in the “Tour of DDE Examples” assume that Steps 1 and 2 of Basic DDE Functions have already been completed.

Starting Excel from SAS

For other versions of Microsoft Office or Excel than what I used for my development, you may need to use a folder path in your code different from that used in this Excel start-up code:

```
x "C:\Program Files\Microsoft Office\Office\EXCEL.exe";
data _null_;
z=sleep(3); /* wait 3 seconds for Excel to start */
run;
```

The above program opens a new workbook, with three empty worksheets: Sheet1, Sheet2, and Sheet3.

Open Existing Workbook from SAS

```
data _null_; /* talk to DDE, no output data */
file DDEcmds;
put '[open("c:\YourFolder\YourWorkbook.xls")]';
x=sleep(3); /* wait 3 seconds for it to open */
run;
```

Finished Workbook Save As

```
data _null_;
file DDEcmds;
put '[error(false)]';
put '[save.as ("c:\OtherFolder\DifferentFileName.xls")]';
x=sleep(1);
run;
```

Exiting Excel from SAS

```
data _null_;
file DDEcmds;
put '[error(false)]';
put '[quit()]; /* empty parenthesis */
run;
```

The `error(false)` command tells Excel that you do not want a prompt to ask for confirmation of intention.

SAS-with-Excel Libraries of Programs and Macros (available via email)

- 26 heavily commented sample programs
- 5 empty, but specially preformatted, spreadsheets to use in sample programs
- Text files of color definition information
- 60 macros for your use, with example invocations in sample programs

Running sample programs with “`OPTIONS MPRINT;`” shows generated SAS code in the SAS log. This enables you to understand what any macro does. If you wish to avoid use of a macro, you can copy generated code from the SAS log, strip off the MPRINT prefix at the left margin, and adapt that code to other uses without the macro.

Tour of DDE Examples

All of the examples presented in this “Tour of DDE Examples” assume that Steps 1 and 2 described in the Section “Basic DDE Functions” have already been completed.

Not On the Tour: For a simple example of how to load data into Excel with DDE, rather than use a pre-existing workbook or have the data preloaded with PROC EXPORT, please see the table-load section of the third example in Appendix 3. Although Appendix 3 is ostensibly devoted to getting SAS/GRAPH output into Excel, the third example includes both a graph and a table.

We will use an Excel workbook which is loaded with data from SASHELP.CLASS and initially uses the default Excel formatting.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Alice	F	13	56.5	84										
3	Barbara	F	13	65.3	98										
4	Carol	F	14	62.8	102.5										
5	Jane	F	12	59.8	84.5										
6	Janet	F	15	62.5	112.5										
7	Joyce	F	11	51.3	50.5										
8	Judy	F	14	64.3	90										
9	Louise	F	12	56.3	77										
10	Mary	F	15	66.5	112										
11	Alfred	M	14	69	112.5										
12	Henry	M	14	63.5	102.5										
13	James	M	12	57.3	83										
14	Jeffrey	M	13	62.5	84										
15	John	M	12	59	99.5										
16	Philip	M	16	72	150										
17	Robert	M	12	64.8	128										
18	Ronald	M	15	67	133										
19	Thomas	M	11	57.5	85										
20	William	M	15	66.5	112										
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

All illustrations in this paper present screen images at full page width to maximize readability. There will be pages with lots of white space at the bottom. By limiting each page to one screen image at most, it is easier to present code and its result together on the same page.

Font Formatting – Step 1

Activate worksheet (if not already there). Select the cells to be formatted

```
data _null_;
file DDEcmds;
put '[workbook.activate("CLASS")]';
put [select("R5C3:R7C5")]';
run;
```

Cells to be formatted Selected:

The screenshot shows a Microsoft Excel window titled "TempClass.xls". The worksheet contains a table with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Alice	F	13	56.5	84										
3	Barbara	F	13	65.3	98										
4	Carol	F	14	62.8	102.5										
5	Jane	F	12	59.8	84.5										
6	Janet	F	15	62.5	112.5										
7	Joyce	F	11	51.3	50.5										
8	Judy	F	14	64.3	90										
9	Louise	F	12	56.3	77										
10	Mary	F	15	66.5	112										
11	Alfred	M	14	69	112.5										
12	Henry	M	14	63.5	102.5										
13	James	M	12	57.3	83										
14	Jeffrey	M	13	62.5	84										
15	John	M	12	59	99.5										
16	Philip	M	16	72	150										
17	Robert	M	12	64.8	128										
18	Ronald	M	15	67	133										
19	Thomas	M	11	57.5	85										
20	William	M	15	66.5	112										
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

The status bar at the bottom shows "Ready" and "Sum=459.1".

Font Formatting – Step 2

Let's use:

- font Verdana
- style Bold
- size 16 point
- (Excel) color 3 - red

Here's how (those commas are very important, and represent unused but available options):

```
data _null_;
file DDEcmds;
put '[font.properties("Verdana","Bold",16,,,,,3)]';
run;
```

The font is changed, but is obscured by the cell selection box and shading:

The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Alice	F	13	56.5	84										
3	Barbara	F	13	65.3	98										
4	Carol	F	14	62.8	102.5										
5	Jane	F	12	59.8	84.5										
6	Janet	F	15	62.5	112.5										
7	Joyce	F	11	51.3	50.5										
8	Judy	F	14	64.3	90										
9	Louise	F	12	56.3	77										
10	Mery	F	15	66.5	112										
11	Alfred	M	14	69	112.5										
12	Henry	M	14	63.5	102.5										
13	James	M	12	57.3	83										
14	Jeffrey	M	13	62.5	84										
15	John	M	12	59	99.5										
16	Philip	M	16	72	150										
17	Robert	M	12	64.8	128										
18	Ronald	M	15	67	133										
19	Thomas	M	11	57.5	85										
20	William	M	15	66.5	112										
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															

The cells for Jane (row 5), Janet (row 6), and Joyce (row 7) are highlighted with a blue selection box and red shading. The font for these cells is Verdana, Bold, 16 point, and red. The status bar at the bottom shows 'Sum=459.1'.

Font Formatting – Step 3

After the block of cells has been formatted, the necessary predecessor `select("R5C3:R7C5")` command leaves them highlighted (just as a mouse would). But I want to see the new font color. So, move the focus with:

```
data _null_;
file DDEcmds;
put '[select("R1C1")]';
run;
```

“I can see clearly now.”

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Alice	F	13	56.5	84										
3	Barbara	F	13	65.3	98										
4	Carol	F	14	62.8	102.5										
5	Jane	F	12	59.8	84.5										
6	Janet	F	15	62.5	112.5										
7	Joyce	F	11	51.3	50.5										
8	Judy	F	14	64.3	90										
9	Louise	F	12	56.3	77										
10	Mary	F	15	66.5	112										
11	Alfred	M	14	69	112.5										
12	Henry	M	14	63.5	102.5										
13	James	M	12	57.3	83										
14	Jeffrey	M	13	62.5	84										
15	John	M	12	59	99.5										
16	Philip	M	16	72	150										
17	Robert	M	12	64.8	128										
18	Ronald	M	15	67	133										
19	Thomas	M	11	57.5	85										
20	William	M	15	66.5	112										
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															

Font Formatting *Can* Be Done All In One Step

```
data _null_;
File DDEcmds;
put '[workbook.activate("Class")]';
put '[select("R5C3:R7C5")]';
put '[font.properties("Verdana","Bold Italic",16,,,,,,,,3)]';
put '[select("R1C1")]';
run;
```

DDE Command Format

In a command of the form

```
put '[CommandName(arg1,arg2,...,argN)]';
```

the **CommandName** may be multi-word, dot-separated.

For an example with several (in this case, four) command arguments, there are three ways to indicate omission of one argument—

- if at start of list: `,arg2,arg3,arg4`
- if in middle of list: `arg1,,arg3,arg4`
- if at end of list: `arg1,arg2,arg3` (i.e., just omit)

Color the Cell Background

```
data _null_;
file DDEcmds;
put '[select.special(11)]'; /* select the last cell of worksheet */
put '[patterns(1,,3,true)]'; /* color it red */
put '[select("R1C1")]'; /* move the highlight to the first cell
                           to see the color with no halo */
run;
```

The screenshot shows a Microsoft Excel window titled 'TempClass.xls'. The spreadsheet contains a table with the following data:

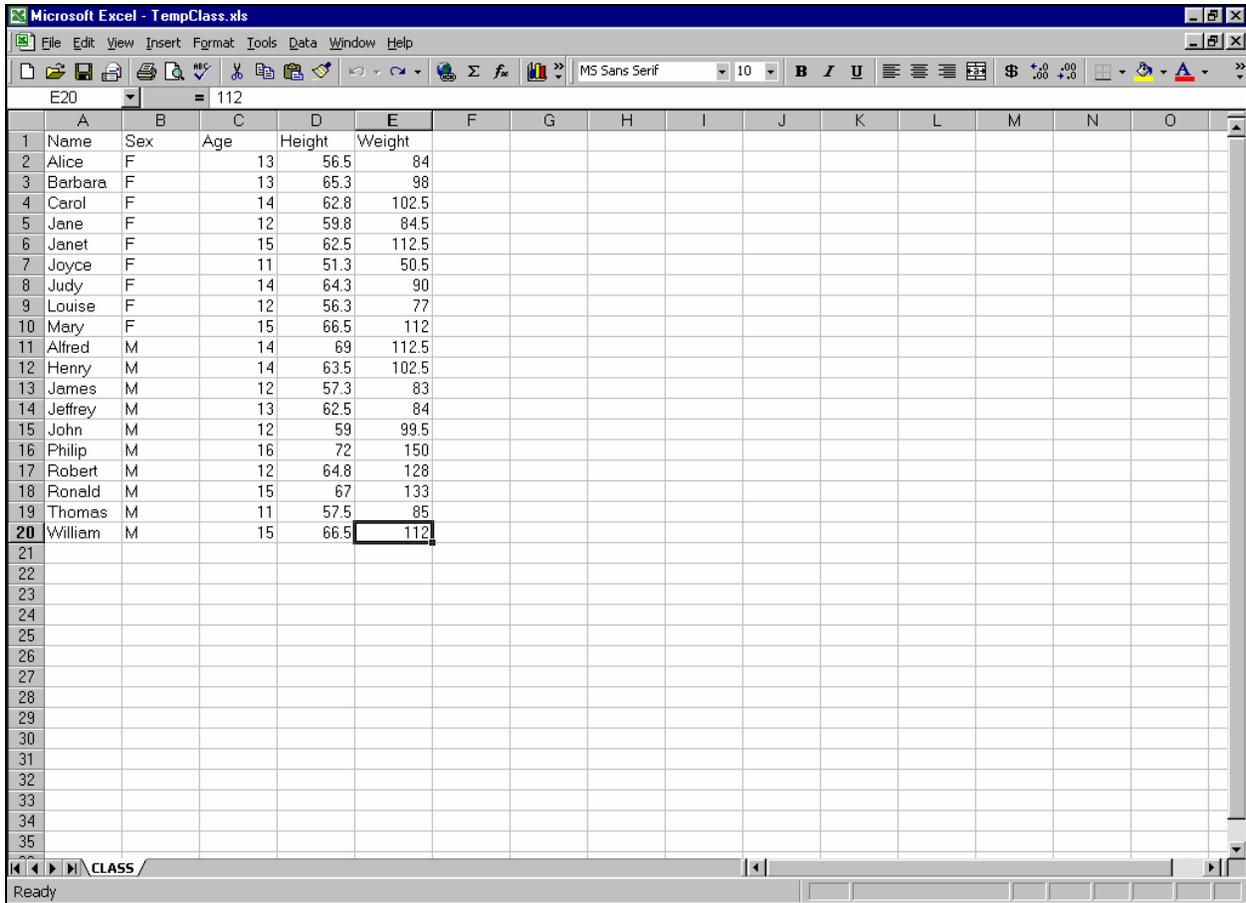
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Alice	F		13	56.5	84									
3	Barbara	F		13	65.3	98									
4	Carol	F		14	62.8	102.5									
5	Jane	F		12	59.8	84.5									
6	Janet	F		15	62.5	112.5									
7	Joyce	F		11	51.3	50.5									
8	Judy	F		14	64.3	90									
9	Louise	F		12	56.3	77									
10	Mary	F		15	66.5	112									
11	Alfred	M		14	69	112.5									
12	Henry	M		14	63.5	102.5									
13	James	M		12	57.3	83									
14	Jeffrey	M		13	62.5	84									
15	John	M		12	59	99.5									
16	Philip	M		16	72	150									
17	Robert	M		12	64.8	128									
18	Ronald	M		15	67	133									
19	Thomas	M		11	57.5	85									
20	William	M		15	66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

The cell at row 20, column E (containing the value 112) is highlighted with a red background. The status bar at the bottom of the window shows 'Ready' and the active cell is 'CLASS'.

**Remove the Background Color
(or reverse whatever was the last command)**

```
data _null_;
file DDEcmds;
put '[undo()]';
run;
```

Color Undone (but focus returns to the cell for which the prior action is undone):



Using Formulas in Excel

Normally, one could create all the Excel report data in SAS pre-spreadsheet-load processing. Suppose, however, we want columns derived from Excel subtotals. For demo only, we will work with simple spreadsheet content, without subtotals. With subtotals present, the use of Excel formulas works the same.

Insert a New Column to the Left of Column 4

```
data _null_;
file DDEcmds;
put '[select("C4")]';
put '[insert(4)]'; /* 4 specifies type of INSERT */
run;
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age		Height	Weight									
2	Alice	F	13		56.5	84									
3	Barbara	F	13		65.3	98									
4	Carol	F	14		62.8	102.5									
5	Jane	F	12		59.8	84.5									
6	Janet	F	15		62.5	112.5									
7	Joyce	F	11		51.3	50.5									
8	Judy	F	14		64.3	90									
9	Louise	F	12		56.3	77									
10	Mary	F	15		66.5	112									
11	Alfred	M	14		69	112.5									
12	Henry	M	14		63.5	102.5									
13	James	M	12		57.3	83									
14	Jeffrey	M	13		62.5	84									
15	John	M	12		59	99.5									
16	Philip	M	16		72	150									
17	Robert	M	12		64.8	128									
18	Ronald	M	15		67	133									
19	Thomas	M	11		57.5	85									
20	William	M	15		66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Excel Format the (Still Empty) Cells in Column 4 Like SAS Format 4.1

```

/* no need to re-select the column, because focus is still there */
data _null_;
file DDEcmds;
put '[Format.Number("#0.0")]';
run;

```

Insert a Heading for the New Column 4

```

filename CELL1 DDE "EXCEL|Class!R1C4";
data _null_;
file CELL1;
put 'Lbs./Yr.';
run;

```

The screenshot shows a Microsoft Excel spreadsheet titled 'TempClass.xls'. The spreadsheet has the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Lbs./Yr.	Height	Weight									
2	Alice	F	13		56.5	84									
3	Barbara	F	13		65.3	98									
4	Carol	F	14		62.8	102.5									
5	Jane	F	12		59.8	84.5									
6	Janet	F	15		62.5	112.5									
7	Joyce	F	11		51.3	50.5									
8	Judy	F	14		64.3	90									
9	Louise	F	12		56.3	77									
10	Mary	F	15		66.5	112									
11	Alfred	M	14		69	112.5									
12	Henry	M	14		63.5	102.5									
13	James	M	12		57.3	83									
14	Jeffrey	M	13		62.5	84									
15	John	M	12		59	99.5									
16	Philip	M	16		72	150									
17	Robert	M	12		64.8	128									
18	Ronald	M	15		67	133									
19	Thomas	M	11		57.5	85									
20	William	M	15		66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Prepare for a Division Operation

Column 3 is to be the divisor. For this demo, set some of its cells to null (missing) values, and some other cells to zero values.

```
data _null_;
file DDEcmds;
put '[select("C3")]';
put '[formula.replace("16","",1,,false,false)]';
put '[formula.replace("12","0",1,,false,false)]';
put '[select("R1C3")]'; /* remove highlight */
run;
```

Note the Presence Now of Some Zero Values and a Null in the Age Column:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Lbs./Yr.	Height	Weight									
2	Alice	F	13		56.5	84									
3	Barbara	F	13		65.3	98									
4	Carol	F	14		62.8	102.5									
5	Jane	F	0		59.8	84.5									
6	Janet	F	15		62.5	112.5									
7	Joyce	F	11		51.3	50.5									
8	Judy	F	14		64.3	90									
9	Louise	F	0		56.3	77									
10	Mary	F	15		66.5	112									
11	Alfred	M	14		69	112.5									
12	Henry	M	14		63.5	102.5									
13	James	M	0		57.3	83									
14	Jeffrey	M	13		62.5	84									
15	John	M	0		59	99.5									
16	Philip	M			72	150									
17	Robert	M	0		64.8	128									
18	Ronald	M	15		67	133									
19	Thomas	M	11		57.5	85									
20	William	M	15		66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Divide Weight By Age

```
data _null_;
file DDEcmds;
put '[select("R2C4")]';
put '[formula.fill(="RC[+2]/RC[-1]","R2C4:R20C4")]';
run;
```

Some (Expected) Error Messages Appear:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Lbs./Yr.	Height	Weight									
2	Alice	F	13	6.5	56.5	84									
3	Barbara	F	13	7.5	65.3	98									
4	Carol	F	14	7.3	62.8	102.5									
5	Jane	F	0	#DIV/0!	59.8	84.5									
6	Janet	F	15	7.5	62.5	112.5									
7	Joyce	F	11	4.6	51.3	50.5									
8	Judy	F	14	6.4	64.3	90									
9	Louise	F	0	#DIV/0!	56.3	77									
10	Mary	F	15	7.5	66.5	112									
11	Alfred	M	14	8.0	69	112.5									
12	Henry	M	14	7.3	63.5	102.5									
13	James	M	0	#DIV/0!	57.3	83									
14	Jeffrey	M	13	6.5	62.5	84									
15	John	M	0	#DIV/0!	59	99.5									
16	Philip	M	0	#DIV/0!	72	150									
17	Robert	M	0	#DIV/0!	64.8	128									
18	Ronald	M	15	8.9	67	133									
19	Thomas	M	11	7.7	57.5	85									
20	William	M	15	7.5	66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Get Rid of the Error Message #DIV/0!

```
data _null_;
file DDEcmds;
put '[select.special(3,16)]'; /* select cells with error messages */
put '[clear]';
run;
```

Cleared Cells Remain Selected:

The screenshot shows a Microsoft Excel spreadsheet titled "TempClass.xls". The spreadsheet has columns labeled A through O and rows numbered 1 through 35. The data is as follows:

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Lbs./Yr.	Height	Weight									
2	Alice	F	13	6.5	56.5	84									
3	Barbara	F	13	7.5	65.3	98									
4	Carol	F	14	7.3	62.8	102.5									
5	Jane	F	0		59.8	84.5									
6	Janet	F	15	7.5	62.5	112.5									
7	Joyce	F	11	4.6	51.3	50.5									
8	Judy	F	14	6.4	64.3	90									
9	Louise	F	0		56.3	77									
10	Mary	F	15	7.5	66.5	112									
11	Alfred	M	14	8.0	69	112.5									
12	Henry	M	14	7.3	63.5	102.5									
13	James	M	0		57.3	83									
14	Jeffrey	M	13	6.5	62.5	84									
15	John	M	0		59	99.5									
16	Philip	M	0		72	150									
17	Robert	M	0		64.8	128									
18	Ronald	M	15	8.9	67	133									
19	Thomas	M	11	7.7	57.5	85									
20	William	M	15	7.5	66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

The error message "#DIV/0!" is visible in the formula bar and cell D5. The spreadsheet is titled "TempClass.xls" and the active cell is D5. The status bar at the bottom shows "Ready".

Replace Nulls with "N/A"

```
data _null_;
file DDEcmds;
put '[formula.replace("", "N/A", 1, , false, false)]';
put '[select("R1C4")]'; /* remove the highlight */
run;
```

Almost Finished:

The screenshot shows a Microsoft Excel spreadsheet titled 'TempClass.xls'. The spreadsheet has the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Lbs./Yr.	Height	Weight									
2	Alice	F	13	6.5	56.5	84									
3	Barbara	F	13	7.5	65.3	98									
4	Carol	F	14	7.3	62.8	102.5									
5	Jane	F	0	N/A	59.8	84.5									
6	Janet	F	15	7.5	62.5	112.5									
7	Joyce	F	11	4.6	51.3	50.5									
8	Judy	F	14	6.4	64.3	90									
9	Louise	F	0	N/A	56.3	77									
10	Mary	F	15	7.5	66.5	112									
11	Alfred	M	14	8.0	69	112.5									
12	Henry	M	14	7.3	63.5	102.5									
13	James	M	0	N/A	57.3	83									
14	Jeffrey	M	13	6.5	62.5	84									
15	John	M	0	N/A	59	99.5									
16	Philip	M	0	N/A	72	150									
17	Robert	M	0	N/A	64.8	128									
18	Ronald	M	15	8.9	67	133									
19	Thomas	M	11	7.7	57.5	85									
20	William	M	15	7.5	66.5	112									
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Make the Headings Bold, and AutoFit the Columns

```
data _null_;
file DDEcmds;
put '[select("R1")]';
put '[font.properties("", "Bold")]';
/* "", = font name unchanged */
put '[select("C1:C6")]';
put '[column.width(,,,3)]'; /* commas required */
/* 3 = AutoFit */
put '[select("R1C1")]'; /* remove the selection highlight */
run;
```

Better, But Still Not Finished:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Name	Sex	Age	Lbs./Yr.	Height	Weight										
2	Alice	F	13	6.5	56.5	84										
3	Barbara	F	13	7.5	65.3	98										
4	Carol	F	14	7.3	62.8	102.5										
5	Jane	F	0	N/A	59.8	84.5										
6	Janet	F	15	7.5	62.5	112.5										
7	Joyce	F	11	4.6	51.3	50.5										
8	Judy	F	14	6.4	64.3	90										
9	Louise	F	0	N/A	56.3	77										
10	Mary	F	15	7.5	66.5	112										
11	Alfred	M	14	8.0	69	112.5										
12	Henry	M	14	7.3	63.5	102.5										
13	James	M	0	N/A	57.3	83										
14	Jeffrey	M	13	6.5	62.5	84										
15	John	M	0	N/A	59	99.5										
16	Philip	M		N/A	72	150										
17	Robert	M	0	N/A	64.8	128										
18	Ronald	M	15	8.9	67	133										
19	Thomas	M	11	7.7	57.5	85										
20	William	M	15	7.5	66.5	112										
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																

Align Columns 2 and 3

```
data _null_;
file DDEcmds;
put '[select("C2:C3")]';          /* Sex & Age Columns */
put '[alignment(3,false,3,0)]'; /* 3 = Center */
put '[select("C4")]';          /* Lbs./Yr. Column */
put '[alignment(4, false,3,0)]'; /* 4 = Right */
put '[select("R1C1")]';
run;
```

But Need To Fix the Format of Height and Weight:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Name	Sex	Age	Lbs./Yr.	Height	Weight										
2	Alice	F	13	6.5	56.5	84										
3	Barbara	F	13	7.5	65.3	98										
4	Carol	F	14	7.3	62.8	102.5										
5	Jane	F	0	N/A	59.8	84.5										
6	Janet	F	15	7.5	62.5	112.5										
7	Joyce	F	11	4.6	51.3	50.5										
8	Judy	F	14	6.4	64.3	90										
9	Louise	F	0	N/A	56.3	77										
10	Mary	F	15	7.5	66.5	112										
11	Alfred	M	14	8.0	69	112.5										
12	Henry	M	14	7.3	63.5	102.5										
13	James	M	0	N/A	57.3	83										
14	Jeffrey	M	13	6.5	62.5	84										
15	John	M	0	N/A	59	99.5										
16	Philip	M		N/A	72	150										
17	Robert	M	0	N/A	64.8	128										
18	Ronald	M	15	8.9	67	133										
19	Thomas	M	11	7.7	57.5	85										
20	William	M	15	7.5	66.5	112										
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																

Excel Format the Height and Weight Data Like SAS Format 4.1

```
data _null_;
file DDEcmds;
put '[select("C5:C6")]';
put '[format.number("#0.0")]';
put '[select("R1C1")]';          /* remove the column selection
highlight */
run;
```

Finished:

1	Name	Sex	Age	Lbs./Yr.	Height	Weight
2	Alice	F	13	6.5	56.5	84.0
3	Barbara	F	13	7.5	65.3	98.0
4	Carol	F	14	7.3	62.8	102.5
5	Jane	F	0	N/A	59.8	84.5
6	Janet	F	15	7.5	62.5	112.5
7	Joyce	F	11	4.6	51.3	50.5
8	Judy	F	14	6.4	64.3	90.0
9	Louise	F	0	N/A	56.3	77.0
10	Mary	F	15	7.5	66.5	112.0
11	Alfred	M	14	8.0	69.0	112.5
12	Henry	M	14	7.3	63.5	102.5
13	James	M	0	N/A	57.3	83.0
14	Jeffrey	M	13	6.5	62.5	84.0
15	John	M	0	N/A	59.0	99.5
16	Philip	M		N/A	72.0	150.0
17	Robert	M	0	N/A	64.8	128.0
18	Ronald	M	15	8.9	67.0	133.0
19	Thomas	M	11	7.7	57.5	85.0
20	William	M	15	7.5	66.5	112.0
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						

Conclusion

There are other ways to generate Excel reports with SAS, but DDE is the method I prefer. Its astonishing power has impressed me. The examples above are only a small subset of what you can do with DDE. Though still just a larger subset of DDE's power, available at no charge via email at the address below are my 60 macros, 26 sample programs (commented), and supporting files, to implement the above and other functions, including:

- Cell sizing
- Column or row Hide/Unhide
- Cell merge, text wrap, etc.
- Create borders to section the worksheet
- Indent cell content
- Freeze/Unfreeze panes
- Turn AutoFilter On/Off
- Generate subtotals in a worksheet
- Control/Change levels for Excel subtotals (or create custom worksheets for various levels of totals)
- Create additional worksheets in a workbook, Delete unused worksheets in a workbook
- Delete columns or rows
- Copy cells, Move cells
- Use Zoom to shrink or enlarge a worksheet
- Control the Message text box in the lower left corner of the Excel window
- Run Excel macros

Bibliography (Including a Paper on Use of DDE for Microsoft Word)

1. Koen Vyverman, "Using Dynamic Data Exchange to Export Your SAS Data to MS Excel – Against All ODS, Part I", *Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference*, SAS Institute Inc. (Cary, NC, USA), 2001.
2. Koen Vyverman, "Creating Custom Excel Workbooks from Base SAS with Dynamic Data Exchange", *Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference*, SAS Institute Inc. (Cary, NC, USA), 2002.
3. Phil Mason, "Customising ODS Output for Microsoft Word Using DDE", *VIEWS 2003 Proceedings*, London, UK, 2003. Available at http://www.views-uk.demon.co.uk/Conference/paper_requests.htm.

Acknowledgments

My thanks to John Xu and Delayne Stokke, MWSUG 2007 Conference Co-chairs, for providing me an opportunity to share some of the power and fun of using DDE for SAS reporting via Excel.

Notices

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other product and brand names are trademarks or registered trademarks of their respective owners.

Author Information

Your requests, questions, comments, suggestions, and other DDE solutions are welcome.

LeRoy Bessler PhD, Fox Point, PO Box 96, Milwaukee, WI 53201, USA, bessler@execpc.com

Appendix 1: Important Information about Using DDE for SAS-with-Excel

If you use DDE for SAS-with-Excel on your desktop/laptop, you may not have any problems. You have control of all Excel options, and you can see everything that is happening with SAS and Excel.

If you have to use SAS and Excel on a remote server, there are some things to be aware of.

1. Possible Excel Situation

If your code has a malfunction, an Excel session may be left open on the server. If it had an Excel file open, the file will remain open. If you do not have Remote Desktop or some sort of terminal access to the server, you will need to have a Windows System Administrator close the file and terminate the Excel session for you.

2. Possible SAS Situation

If you are using SAS Enterprise Guide as a client for that remote server and your code has a malfunction, a SAS process may remain running or "hung" on the server. If your version of SAS EG permits you to kill the process, you may not have a problem. If your version of SAS EG does not permit you to kill the process, and you do not have Remote Desktop or some sort of terminal access to the server, you will need to have a Windows System Administrator kill it for you.

3. Possible Excel Situation

Only two of the examples involve the use of a custom Excel macro. Excel on your server may be set to inhibit the use of such Excel macros, or may put certain conditions on their use. If so, you need to work with your Windows System Administrator either to get the inhibition removed or to find out what you need to do get your macro(s) authorized to run.

Appendix 2: Using Dynamic Data Exchange to Load a Pivot Table from SAS with Your SAS-prepared Data

Abstract

This coding tidbit shows you how to load a pivot table with SAS by running a pre-recorded custom Excel macro via DDE.

Introduction

Knowing that pivot tables are a very common destination of SAS-originated data, whether prepared with PROC SUMMARY or some other method, and being a DDE enthusiast, I long wanted to use DDE to create pivot tables. There appears to be some machinery available, if you read MacroFun.HLP, but I could not get it to work. When I asked Cynthia Zender from SAS Institute about my problem, she suggested pre-recording an Excel macro to handle the load, and then running the Excel macro with DDE. I had a little bit of experience with that, and so this solution is the result.

For a different way to create pivot tables from SAS, please see Conway's work in Reference 1.

Solution

The code below is meant to be self-explanatory, and it assumes that you have already worked through other information, including the main body of the paper and the more basic sample programs that are in the BesslerSASandExcel zip file package available via email. It uses SAS macros from that package.

```
* UseCustomPreRecordedPivotTableMacro.sas *;

* NOTE:
  To use this sample code,
  you can choose to highlight specific code blocks
  and submit them individually.
  This sample program need not be submitted and run en masse.
  Once Excel has been started and a spreadsheet has been opened,
  you can toggle over to the Excel window
  after each SAS code execution step
  to inspect the results. *;

* These folders can, of course, be named anything you prefer,
  but the expected macros, input data, or output data
  must be stored there. *;
%let FolderForMacros      = C:\BesslerSASandExcel\macros;
%let FolderForXLsfiles   = C:\BesslerSASandExcel\spreadsheets;
%let FolderForXLsoutput  = C:\BesslerSASandExcel\output;

* tell SAS where the macros are stored *;
filename SASandXL "&FolderForMacros";
* search for macros in the order listed *;
options sasautos=(SASandXL sasautos);

options mprint; * list the resolved translation of any macros
                in the SAS log *;

* Specifying the location of the Excel executable *;
%XLStart(ExcelPgm=C:\Program Files\Microsoft Office\Office10\EXCEL.exe);

%XLDDDecmdsFileName;
```

```
* open the workbook that contains the macro(s) *;
%XLOpen(WorkBook=&FolderForXLSfiles.\PivotTableMacroForSASHELPdotCLASSweightVar.xls);

* open the workbook to be processed with the macro(s) *;
%XLOpen(WorkBook=&FolderForXLSfiles.\TempClass.xls);

* run the macro *;
* The DDE Pivot function refused to work for this situation.
  that's why I use the pre-recorded macro instead. *;
%XLRunMacro(WorkBook=PivotTableMacroForSASHELPdotCLASSweightVar.xls,
            XLMacro=Macro1);

%XLDeleteWorkSheet(WorkSheet=CLASS);

* save the result with a new XLS file name *;
%XLSaveAs(WorkBook=&FolderForXLSoutput.\sasHelpClassWeightVarInPivotTable.xls);

* Close that WorkBook *;
%XLClose;

* Close the Excel Macro WorkBook *;
%XLClose;

* Exit *;
%XLExit;
```

Reference (Non-DDE Solution)

1. Ted Conway, Sur La Table: Creating Microsoft Excel PivotTables in a Jiffy from SAS Data, *Proceedings of the Thirty-First Annual SAS Users Group International Conference*, 2006. Cary, NC: SAS Institute Inc.

Appendix 3: Using DDE to Insert and Re-Size SAS-Created Graphs

Abstract

These sample programs show you how to use DDE to insert (and, if desired, re-size) graphs, either standalone on separate worksheets, in groups on a worksheet, or with a companion table.

Introduction

I have often been asked how to use DDE to create Excel worksheets that contain SAS/GRAPH output, and always had to say that I did not know. Since in other papers I have been exploring and explaining how to create SAS graphs in a wide variety of different delivery contexts, I finally decided to investigate Excel via DDE. Here no use is made of my SAS-DDE macros. All of the programming is explicit and hard-coded. The new DDE tools used are the `INSERT.PICTURE` and `FORMAT.SIZE` functions.

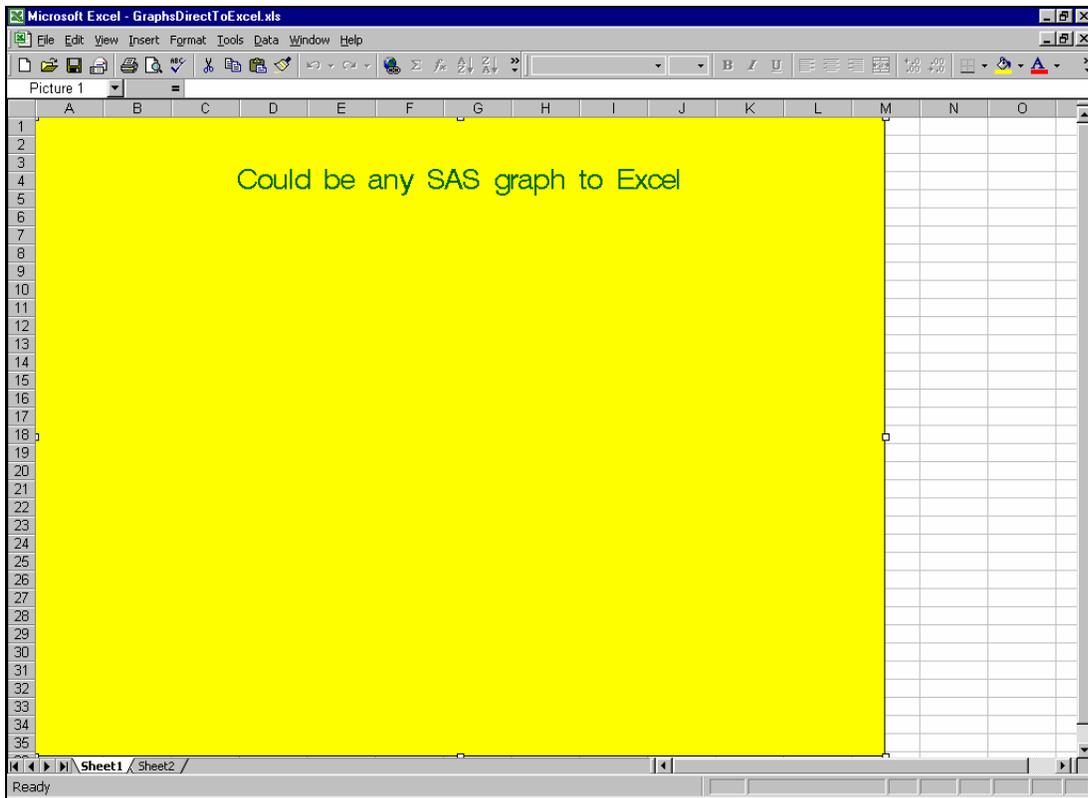
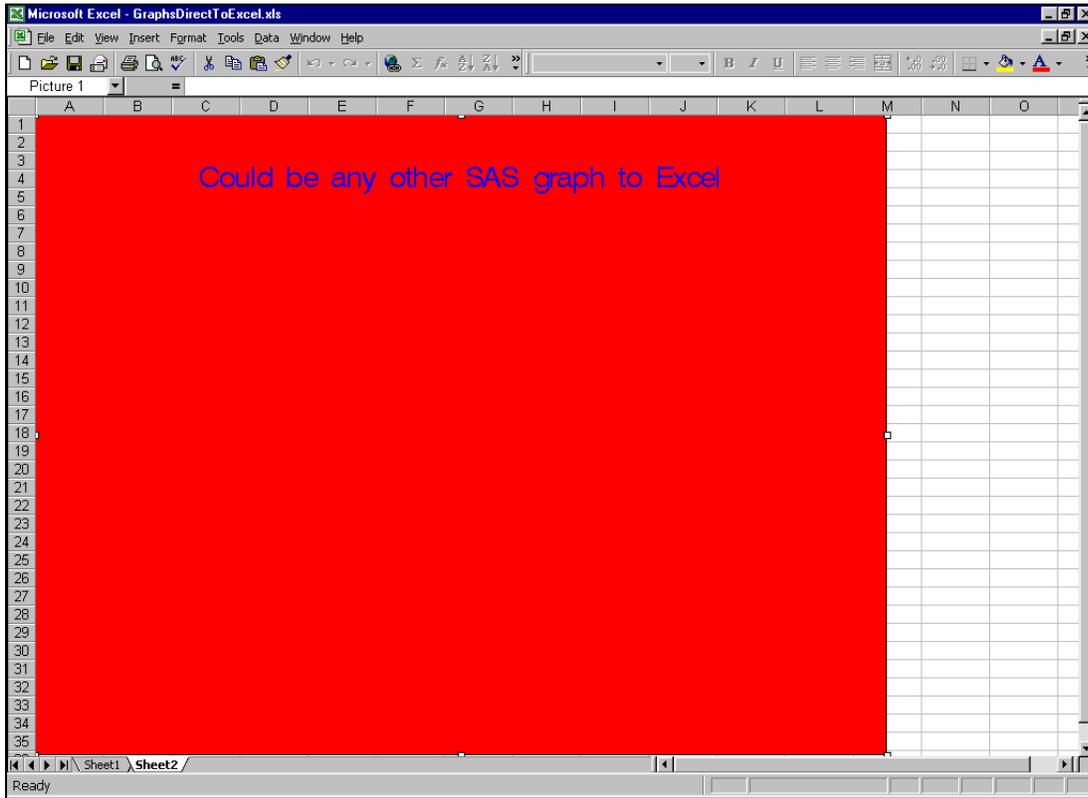
Examples of SAS/GRAPH to Excel via DDE

The code below is meant to be understandable without further explanation, and assumes that you already understand basic DDE functions.

In all three examples, a new Excel workbook is opened, one worksheet is customized, and the remaining two sheets present by default are deleted. Instead, a pre-existing workbook could be used and customized, by modifying a pre-existing worksheet or a newly inserted one.

Since these examples are meant to demonstrate how to get SAS/GRAPH output into Excel, and not to teach you how to use SAS/GRAPH, the “graphs” are merely text slides that are trivial to create.

Two Graphs to Two Separate Worksheets



Here is the code used to create the Excel workbook above:

```

goptions reset=all;
goptions device=GIF;
goptions border; /* Put the graph in a box. */
goptions gsfmode=replace gsfname=anyname;
proc gslide;
filename anyname "C:\image1.GIF";
goptions cback=CXFFFF00; /* Use RGB Yellow for background. */
/* Use RGB Medium Dark Green for title. */
title1 ' ' justify=CENTER height=2 color=CX006600 'Could be any SAS graph to Excel';
run;
filename anyname "C:\image2.GIF";
goptions cback=CXFF0000; /* Use RGB Red for background. */
/* Use RGB Blue for text. */
title1 ' ' justify=CENTER height=2 color=CX0000FF 'Could be any other SAS graph to
Excel';
run;
quit;

options noxwait noxsync;

x "C:\Program Files\Microsoft Office\Office\EXCEL.exe";

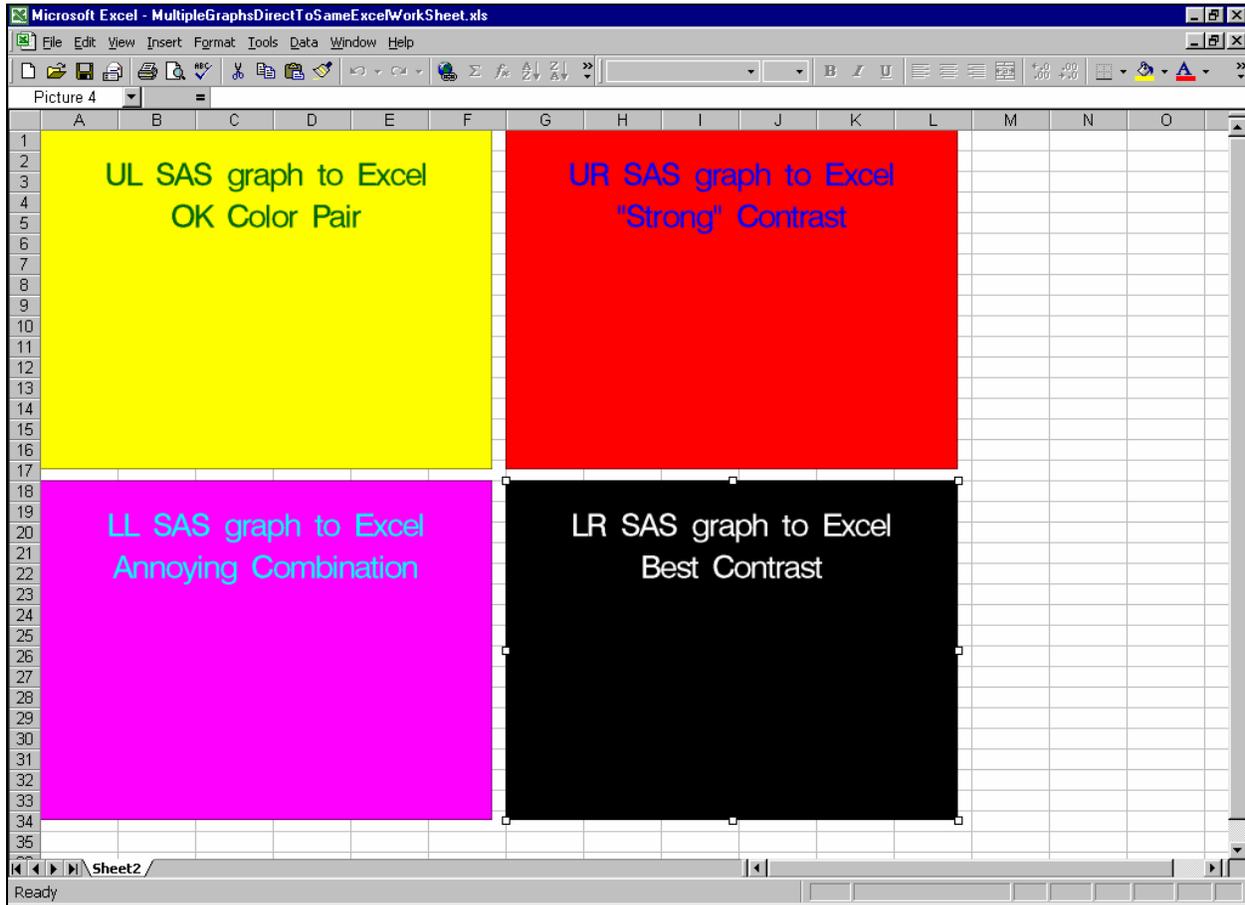
data _null_;
z=sleep(3); /* wait 3 seconds for Excel to start */
run;

filename DDEcmds dde "excel|system";

data _null_;
file DDEcmds;
put '[workbook.activate("Sheet1")]';
put '[insert.picture("C:\image1.GIF")]';
put '[format.size(600,450)]'; /* this really keeps it unchanged */
put '[workbook.activate("Sheet2")]';
put '[insert.picture("C:\image2.GIF")]';
put '[error(false)]';
put '[workbook.delete("Sheet3")]';
put '[error(false)]';
put '[save.as("C:\GraphsDirectToExcel.xls")]';
z=sleep(3); /* wait 3 seconds for Save to end */
put '[error(false)]';
put '[quit()]'; /* empty parenthesis */
run;

```

Four Graphs to the Same WorkSheet



Here is the code used to create the Excel worksheet above:

```

goptions reset=all;
goptions device=GIF;
goptions border;
goptions gsfmode=replace gsfname=anyname;
proc gslide;
filename anyname "C:\image1.GIF";
goptions cback=CXFFFF00;
title ' ' justify=CENTER height=4 color=CX006600
'UL SAS graph to Excel'
justify=CENTER
'OK Color Pair';
run;
filename anyname "C:\image2.GIF";
goptions cback=CXFF0000;
title ' ' justify=CENTER height=4 color=CX0000FF
'UR SAS graph to Excel'
justify=CENTER
'"Strong" Contrast';
run;
proc gslide;
filename anyname "C:\image3.GIF";
goptions cback=CXFF00FF; /* Use RGB Magenta for background. */
/* Use RGB Cyan (Turquoise) for text. */

```

```

title1 ' ' justify=CENTER height=4 color=CX00FFFF
'LL SAS graph to Excel'
justify=CENTER
'Annoying Combination';
run;
filename anyname "C:\image4.GIF";
goptions cback=CX000000; /* Use RGB Black for background. */
/* Use RGB White for text. */
title1 ' ' justify=CENTER height=4 color=CXFFFFFF
'LR SAS graph to Excel'
justify=CENTER
'Best Contrast';
run;
quit;

options noxwait noxsync;

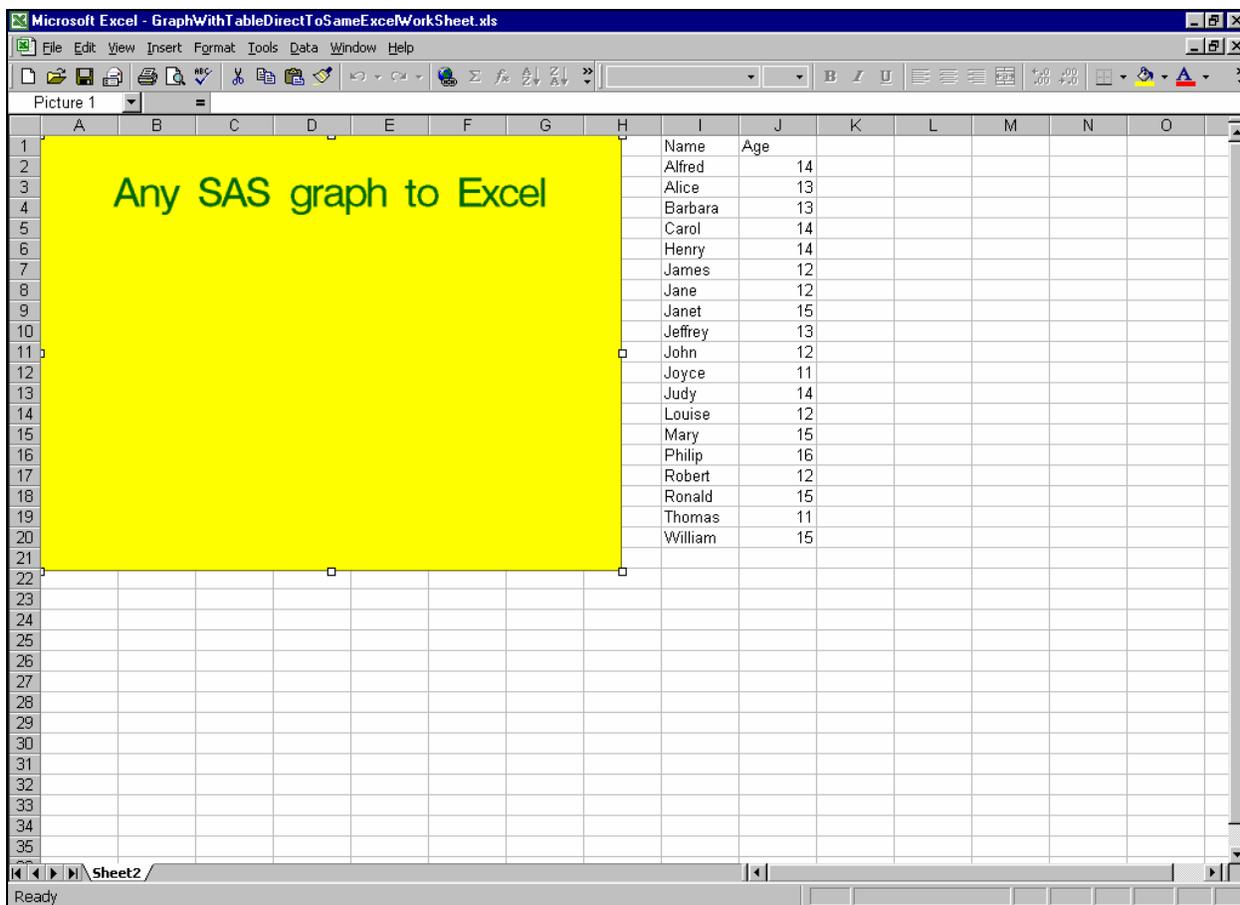
x "C:\Program Files\Microsoft Office\Office\EXCEL.exe";

data _null_;
z=sleep(3);
run;

filename DDEcmds dde "excel|system";

data _null_;
file DDEcmds;
put '[workbook.activate("Sheet2")]';
put '[select("R1C1")]'; /* really the default position after activate */
put '[insert.picture("C:\image1.GIF")]';
put '[format.size(280,210)]';
put '[select("R1C7")]';
put '[insert.picture("C:\image2.GIF")]';
put '[format.size(280,210)]';
put '[select("R18C1")]';
put '[insert.picture("C:\image3.GIF")]';
put '[format.size(280,210)]';
put '[select("R18C7")]';
put '[insert.picture("C:\image4.GIF")]';
put '[format.size(280,210)]';
put '[error(false)]';
put '[workbook.delete("Sheet1")]';
put '[error(false)]';
put '[workbook.delete("Sheet3")]';
put '[error(false)]';
put '[save.as("C:\MultipleGraphsDirectToSameExcelWorkSheet.xls")]';
z=sleep(3);
put '[error(false)]';
put '[quit()]';
run;

```



Here is the code used to create the Excel worksheet above:

```

options reset=all;
options device=GIF;
options border;
options gsfmode=replace gsfname=anyname;
proc gslide;
filename anyname "C:\image1.GIF";
options cback=CXFFFFF0;
title1 ' ' justify=CENTER height=4 color=CX006600 'Any SAS graph to Excel';
run; quit;

```

```
options noxwait noxsync;
```

```

/* for your version of the Microsoft Office suite,
you may need to use a different folder path below. */

```

```
x "C:\Program Files\Microsoft Office\Office\EXCEL.exe";
```

```

data _null_;
z=sleep(3);
run;

```

```
filename DDEcmds dde "excel|system";
```

```

data _null_;
file DDEcmds;
put '[workbook.activate("Sheet2")]';

```

```

put '[insert.picture("C:\image1.GIF")]';
put '[format.size(360,270)]';
run;

/* insert labels for the table */

filename Labels DDE 'excel|Sheet2!r1c9:r1c10';

data _null_;
file Labels;
LABEL1 = 'Name';
LABEL2 = 'Age';
put LABEL1 LABEL2;
run;

/* load the table data */

filename TblData DDE 'excel|Sheet2!r2c9:r20c10';

data _null_;
file TblData;
set sashelp.class;
put Name Age;
run;

/* At this point, DDE could be used to customize the appearance of the table. */
/* However, it will be left As Is. */

data _null_;
file DDEcmds;
put '[error(false)]';
put '[workbook.delete("Sheet1")]';
put '[error(false)]';
put '[workbook.delete("Sheet3")]';
put '[error(false)]';
put '[save.as("C:\GraphWithTableDirectToSameExcelWorkSheet.xls")]';
z=sleep(3);
put '[error(false)]';
put '[quit()]';
run;

```

Appendix 4: SAS Creation of Excel SubTotals

This information was not included in the original version of the paper. It includes: (a) text copied from the extra slides in the presentation; and (b) captioned slides of Excel screen print images.

Excel Subtotals: Two Solutions

1. Separate Custom Worksheets for each level
2. DDE use of Excel's subtotals (requires a pre-built Excel macro for one step)

Excel Subtotals Solution 1: Separate Custom Worksheets for Each Level

An Alternative to Excel Subtotals

- Custom Design: Really “Have It Your Way”
- Prepare multiple pre-formatted empty WorkSheets
- Load them via DDE
- May have to use Software Intelligence for post-load formatting
- User moves from “level” to “level via Tab Navigation, not Excel “Level Buttons”
- Here showing only results, not code

Overview WorkSheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Sex	Height	Weight												
2	F Total	545.3	811.0												
3	M Total	639.1	1089.5												
4	Grand Total	1184.4	1900.5												
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Summary WorkSheet

Age	Sex	Height	Weight
F 11 Total		51.3	50.5
F 12 Total		116.1	161.5
F 13 Total		121.8	182.0
F 14 Total		127.1	192.5
F 15 Total		129.0	224.5
F Total		545.3	811.0
M 11 Total		57.5	85.0
M 12 Total		181.1	310.5
M 13 Total		62.5	84.0
M 14 Total		132.5	215.0
M 15 Total		133.5	245.0
M 16 Total		72.0	150.0
M Total		639.1	1089.5
Grand Total		1184.4	1900.5

Detail + Summary WorkSheet

Age	Sex	Name	Height	Weight
F 11		Joyce	51.3	50.5
F 11 Total			51.3	50.5
F 12		Jane	59.8	84.5
F 12		Louise	56.3	77.0
F 12 Total			116.1	161.5
F 13		Alice	56.5	84.0
F 13		Barbara	65.3	98.0
F 13 Total			121.8	182.0
F 14		Carol	62.8	102.5
F 14		Judy	64.3	90.0
F 14 Total			127.1	192.5
F 15		Janet	62.5	112.5
F 15		Mary	66.5	112.0
F 15 Total			129.0	224.5
F Total			545.3	811.0
M 11		Thomas	57.5	85.0
M 11 Total			57.5	85.0
M 12		James	57.3	83.0
M 12		John	59.0	99.5
M 12		Robert	64.8	128.0
M 12 Total			181.1	310.5
M 13		Jeffrey	62.5	84.0
M 13 Total			62.5	84.0
M 14		Alfred	69.0	112.5
M 14		Henry	63.5	102.5
M 14 Total			132.5	215.0
M 15		Ronald	67.0	133.0
M 15		William	66.5	112.0
M 15 Total			133.5	245.0
M 16		Philip	72.0	150.0
M 16 Total			72.0	150.0
M Total			639.1	1089.5
Grand Total			1184.4	1900.5

Detail Only WorkSheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Name	Sex	Age	Height	Weight												
2	Alfred	M	14	69.0	112.5												
3	Alice	F	13	56.5	84.0												
4	Barbara	F	13	65.3	98.0												
5	Carol	F	14	62.8	102.5												
6	Henry	M	14	63.5	102.5												
7	James	M	12	57.3	83.0												
8	Jane	F	12	59.8	84.5												
9	Janet	F	15	62.5	112.5												
10	Jeffrey	M	13	62.5	84.0												
11	John	M	12	59.0	99.5												
12	Joyce	F	11	51.3	50.5												
13	Judy	F	14	64.3	90.0												
14	Louise	F	12	56.3	77.0												
15	Mary	F	15	66.5	112.0												
16	Philip	M	16	72.0	150.0												
17	Robert	M	12	64.8	128.0												
18	Ronald	M	15	67.0	133.0												
19	Thomas	M	11	57.5	85.0												
20	William	M	15	66.5	112.0												
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	
35																	

Excel Subtotals Solution 2: DDE Use of Excel's Subtotals

Loaded WorkSheet (e.g., via Export)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Alfred	M	14	69	112.5										
3	Alice	F	13	56.5	84										
4	Barbara	F	13	65.3	98										
5	Carol	F	14	62.8	102.5										
6	Henry	M	14	63.5	102.5										
7	James	M	12	57.3	83										
8	Jane	F	12	59.8	84.5										
9	Janet	F	15	62.5	112.5										
10	Jeffrey	M	13	62.5	84										
11	John	M	12	59	99.5										
12	Joyce	F	11	51.3	50.5										
13	Judy	F	14	64.3	90										
14	Louise	F	12	56.3	77										
15	Mary	F	15	66.5	112										
16	Philip	M	16	72	150										
17	Robert	M	12	64.8	128										
18	Ronald	M	15	67	133										
19	Thomas	M	11	57.5	85										
20	William	M	15	66.5	112										
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

Sort the Data

```

data _null_;
file DDEcmds;
put '[select("R1C1:R65536C256")]'; /* all cells */
put '[sort(1,"Sex",1,"Age",1,"Name",1,1,1,0)]';
put '[select("R2C1")]';
run;
/* "Key",1 = sort on Key ascending */
/* 1 = sort by rows */
/* 1,1,0 = Normal sort, Headers present, Not case sensitive */

```

By Name Within Age Within Sex

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Sex	Age	Height	Weight										
2	Joyce	F	11	51.3	50.5										
3	Jane	F	12	59.8	84.5										
4	Louise	F	12	56.3	77										
5	Alice	F	13	56.5	84										
6	Barbera	F	13	85.3	98										
7	Carol	F	14	62.8	102.5										
8	Judy	F	14	84.3	90										
9	Janet	F	15	62.5	112.5										
10	Mary	F	15	66.5	112										
11	Thomas	M	11	57.5	85										
12	James	M	12	57.3	83										
13	John	M	12	59	99.5										
14	Robert	M	12	64.8	128										
15	Jeffrey	M	13	62.5	84										
16	Alfred	M	14	69	112.5										
17	Henry	M	14	63.5	102.5										
18	Ronald	M	15	67	133										
19	William	M	15	66.5	112										
20	Philip	M	16	72	150										
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															

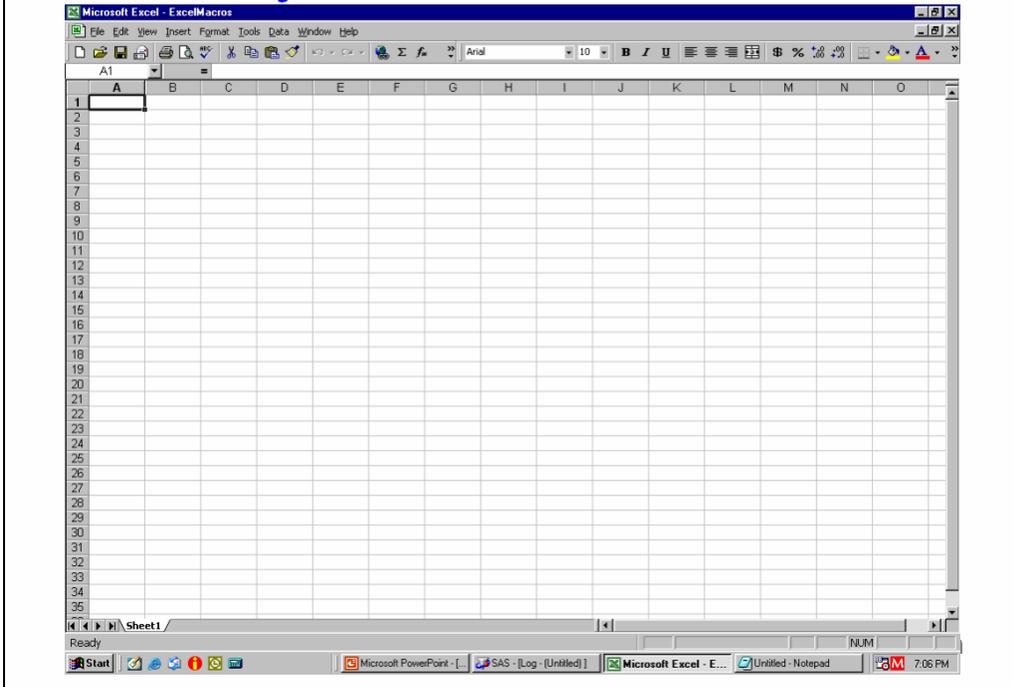
Close Data Workbook, Open a Workbook of Only Excel Macros

```

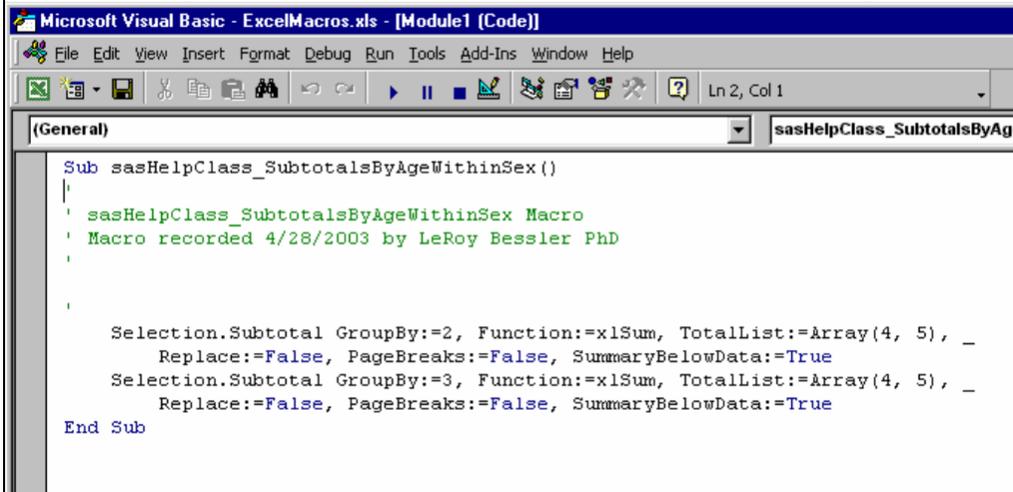
data _null_;
file DDEcmds;
put '[close(true)]';
x=sleep(1);
put '[open("C:\Folder\ExcelMacros.xls")]';
x=sleep(3);
run;

```

Only Excel Macros Here



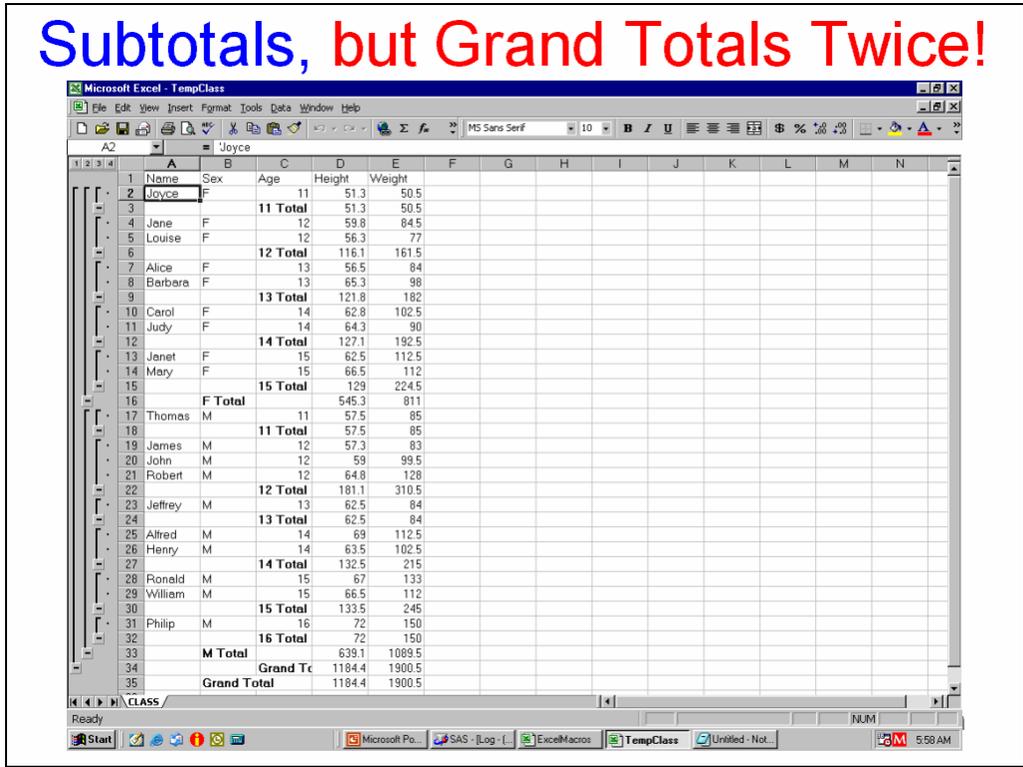
Open Macro for Demo Only



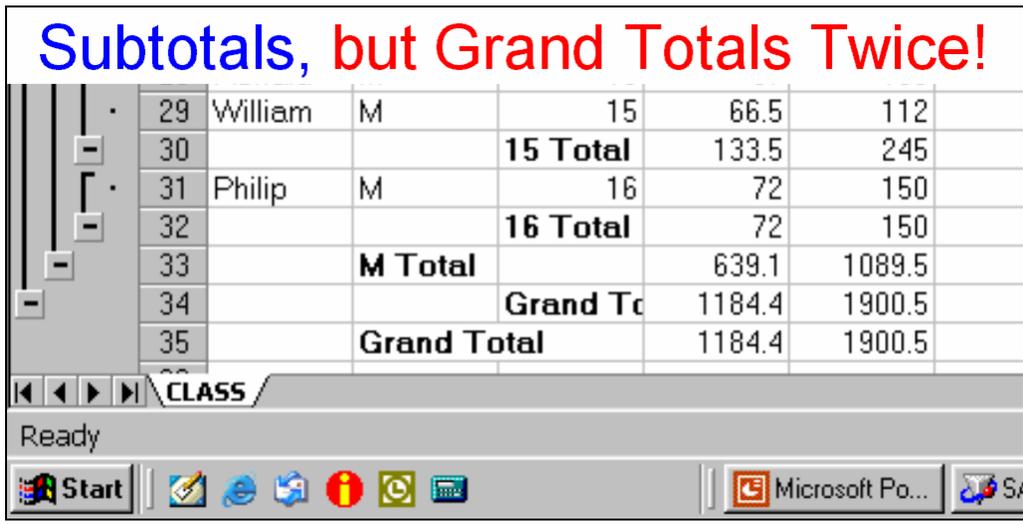
Open Data Workbook, and Run the Subtotals Excel Macro

```
data _null_;
file DDEcmds;
put '[open("C:\Folder\TempClass.xls")]';
x=sleep(3);
put '[run("ExcelMacros.xls!DoSubtotals",false)]';
run;
```

Subtotals, but Grand Totals Twice!



Subtotals, but Grand Totals Twice!



Removing the Extra Grand Total

First, Reformat & Save

```
data _null_;
file DDEcmds;
put '[select("R1")]';
put '[font.properties("", "Bold")]';
put '[select("C1:C5")]';
put '[column.width(,,,3)]';
put '[error(false)]';
put '[save.as("C:\Folder\DataWithSubtotals.xls")]';
x=sleep(1);
run;
```

Remove Extra Grand Total, Part 1:

Find Second from Last Row & Put Row Number in Symbol Table

```
filename Column DDE "EXCEL|CLASS!R1C1:R65536C1" notab;
data _null_;
length FirstCharInColumnCell $ 1;
infile Column trunccover end=LastObs;
input FirstCharInColumnCell;
if LastObs;
call symput("RowToRemove",trim(left(_N_-1)));
run;
```

Remove Extra Grand Total, Part 2:

Select Second from Last Row & Delete It

```
data _null_; file DDEcmds;
put '[select("R' "&RowToRemove" '")]';
/* resolves to: put '[select("R34")]'; */
put '[edit.delete(3)]';
run;
```

Only One Grand Total

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Sex	Age	Height	Weight									
2	Joyce	F	11	51.3	50.5									
3			11 Total	51.3	50.5									
4	Jene	F	12	59.8	84.5									
5	Louise	F	12	56.3	77									
6			12 Total	116.1	161.5									
7	Alice	F	13	56.5	84									
8	Barbara	F	13	65.3	98									
9			13 Total	121.8	182									
10	Carol	F	14	62.8	102.5									
11	Judy	F	14	64.3	90									
12			14 Total	127.1	192.5									
13	Janet	F	15	62.5	112.5									
14	Mary	F	15	66.5	112									
15			15 Total	129	224.5									
16		F Total		645.3	811									
17	Thomas	M	11	57.5	85									
18			11 Total	57.5	85									
19	James	M	12	57.3	83									
20	John	M	12	59	99.5									
21	Robert	M	12	64.8	128									
22			12 Total	181.1	310.5									
23	Jeffrey	M	13	62.5	84									
24			13 Total	62.5	84									
25	Alfred	M	14	69	112.5									
26	Henry	M	14	63.5	102.5									
27			14 Total	132.5	215									
28	Ronald	M	15	67	133									
29	William	M	15	66.5	112									
30			15 Total	133.5	245									
31	Philip	M	16	72	150									
32			16 Total	72	150									
33		M Total		639.1	1089.5									
34		Grand Total		1184.4	1900.5									
35														

Changing & Indenting Values and Changing Excel Subtotal Labels

Pseudo-indent Cell Content

```

data _null_;
file DDECmnds;
put '[select("R2C2:R15C2")]';
put '[formula.replace("F","__Female",1,,false,false)]';

/* change color of three__ to White to match cell background: */
put '[font.properties(,,,,,,,,,2,,,1,3)]';
/* 2 = White , StartAt = 1, ForLengthOf = 3 */

run;

```

Indent Work In Progress

	Name	Sex	Age	Height	Weight
2	Joyce	Female	11	51.3	50.5
3			11 Total	51.3	50.5
4	Jane	Female	12	59.8	84.5
5	Louise	Female	12	56.3	77
6			12 Total	116.1	161.5
7	Alice	Female	13	56.5	84
8	Barbara	Female	13	65.3	98
9			13 Total	121.8	182
10	Carol	Female	14	62.8	102.5
11	Judy	Female	14	64.3	90
12			14 Total	127.1	192.5
13	Janet	Female	15	62.5	112.5
14	Mary	Female	15	66.5	112
15			15 Total	129	224.5
16		F Total		545.3	811
17	Thomas	M	11	57.5	85
18			11 Total	57.5	85
19	James	M	12	57.3	83
20	John	M	12	59	99.5
21	Robert	M	12	64.8	128
22			12 Total	181.1	310.5
23	Jeffrey	M	13	62.5	84
24			13 Total	62.5	84

Indents Complete

	Name	Sex	Age	Height	Weight
2	Joyce	Female	11	51.3	50.5
3			11 Total	51.3	50.5
4	Jane	Female	12	59.8	84.5
5	Louise	Female	12	56.3	77
6			12 Total	116.1	161.5
7	Alice	Female	13	56.5	84
8	Barbara	Female	13	65.3	98
9			13 Total	121.8	182
10	Carol	Female	14	62.8	102.5
11	Judy	Female	14	64.3	90
12			14 Total	127.1	192.5
13	Janet	Female	15	62.5	112.5
14	Mary	Female	15	66.5	112
15			15 Total	129	224.5
16		F Total		545.3	811
17	Thomas	Male	11	57.5	85
18			11 Total	57.5	85
19	James	Male	12	57.3	83
20	John	Male	12	59	99.5
21	Robert	Male	12	64.8	128
22			12 Total	181.1	310.5
23	Jeffrey	Male	13	62.5	84
24			13 Total	62.5	84

New Labels for Sex Subtotals

	Name	Sex	Age	Height	Weight
2	Joyce	Female	11	51.3	50.5
3			11 Total	51.3	50.5
4	Jane	Female	12	59.8	84.5
5	Louise	Female	12	56.3	77
6			12 Total	116.1	161.5
7	Alice	Female	13	56.5	84
8	Barbara	Female	13	65.3	98
9			13 Total	121.8	182
10	Carol	Female	14	62.8	102.5
11	Judy	Female	14	64.3	90
12			14 Total	127.1	192.5
13	Janet	Female	15	62.5	112.5
14	Mary	Female	15	66.5	112
15			15 Total	129	224.5
16		All Females		545.3	811
17	Thomas	Male	11	57.5	85
18			11 Total	57.5	85
19	James	Male	12	57.3	83
20	John	Male	12	59	99.5
21	Robert	Male	12	64.8	128
22			12 Total	181.1	310.5
23	Jeffrey	Male	13	62.5	84
24			13 Total	62.5	84

Controlling Subtotal Display

Excel Subtotals, Showing to Level 4

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Sex	Age	Height	Weight									
2	Joyce	Female	11	51.3	50.5									
3			11 Total	51.3	50.5									
4	Jane	Female	12	59.8	84.5									
5	Louise	Female	12	56.3	77									
6			12 Total	116.1	161.5									
7	Alice	Female	13	56.5	84									
8	Barbara	Female	13	65.3	98									
9			13 Total	121.8	182									
10	Carol	Female	14	62.8	102.5									
11	Judy	Female	14	64.3	90									
12			14 Total	127.1	192.5									
13	Janet	Female	15	62.5	112.5									
14	Mary	Female	15	66.5	112									
15			15 Total	129	224.5									
16		All Females		545.3	811									
17	Thomas	Male	11	57.5	85									
18			11 Total	57.5	85									
19	James	Male	12	57.3	83									
20	John	Male	12	59	99.5									
21	Robert	Male	12	64.8	128									
22			12 Total	181.1	310.5									
23	Jeffrey	Male	13	62.5	84									
24			13 Total	62.5	84									
25	Alfred	Male	14	69	112.5									
26	Henry	Male	14	63.5	102.5									
27			14 Total	132.5	215									
28	Ronald	Male	15	67	133									
29	William	Male	15	66.5	112									
30			15 Total	133.5	245									
31	Philip	Male	16	72	150									
32			16 Total	72	150									
33		All Males		639.1	1089.5									
34		Grand Total		1184.4	1900.5									
35														

Show Subtotals Only to Level 3:

```
data _null_; file DDEcmds; put '[show.levels(3,0)]'; run;
```

Excel Subtotals, Showing to Level 3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Sex	Age	Height	Weight									
3			11 Total	51.3	50.5									
6			12 Total	116.1	161.5									
9			13 Total	121.8	182									
12			14 Total	127.1	192.5									
15			15 Total	129	224.5									
16		All Females		545.3	811									
18			11 Total	57.5	85									
22			12 Total	181.1	310.5									
24			13 Total	62.5	84									
27			14 Total	132.5	215									
30			15 Total	133.5	245									
32			16 Total	72	150									
33		All Males		639.1	1089.5									
34		Grand Total		1184.4	1900.5									
35														

Excel Subtotals, Showing to Level 2

Microsoft Excel - sasHelpClassWithSubtotals

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Sex	Age	Height	Weight									
16		All Females		545.3	611									
33		All Males		639.1	1089.5									
34		Grand Total		1184.4	1900.5									

Excel Subtotals, Showing to Level 1

Microsoft Excel - sasHelpClassWithSubtotals

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
34		Grand Total		1184.4	1900.5									