

Paper 022-31

SAS®-with-Excel Application Development: Tools and Techniques

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 the 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](http://support.microsoft.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>.

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 sequence 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.

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 "Microsoft Excel - 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 window titled "Microsoft Excel - TempClass.xls". The spreadsheet contains a table with 20 rows and 16 columns (A-O). The first five columns (A-E) contain data. The data is as follows:

	A	B	C	D	E
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

The cells containing the values 12, 59.8, 84.5, 15, 62.5, 112.5, 11, 51.3, and 50.5 (rows 5-7, columns C-E) are highlighted with a blue selection box and have a light blue background shading. The font for these cells is Verdana, Bold, size 16, and color 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

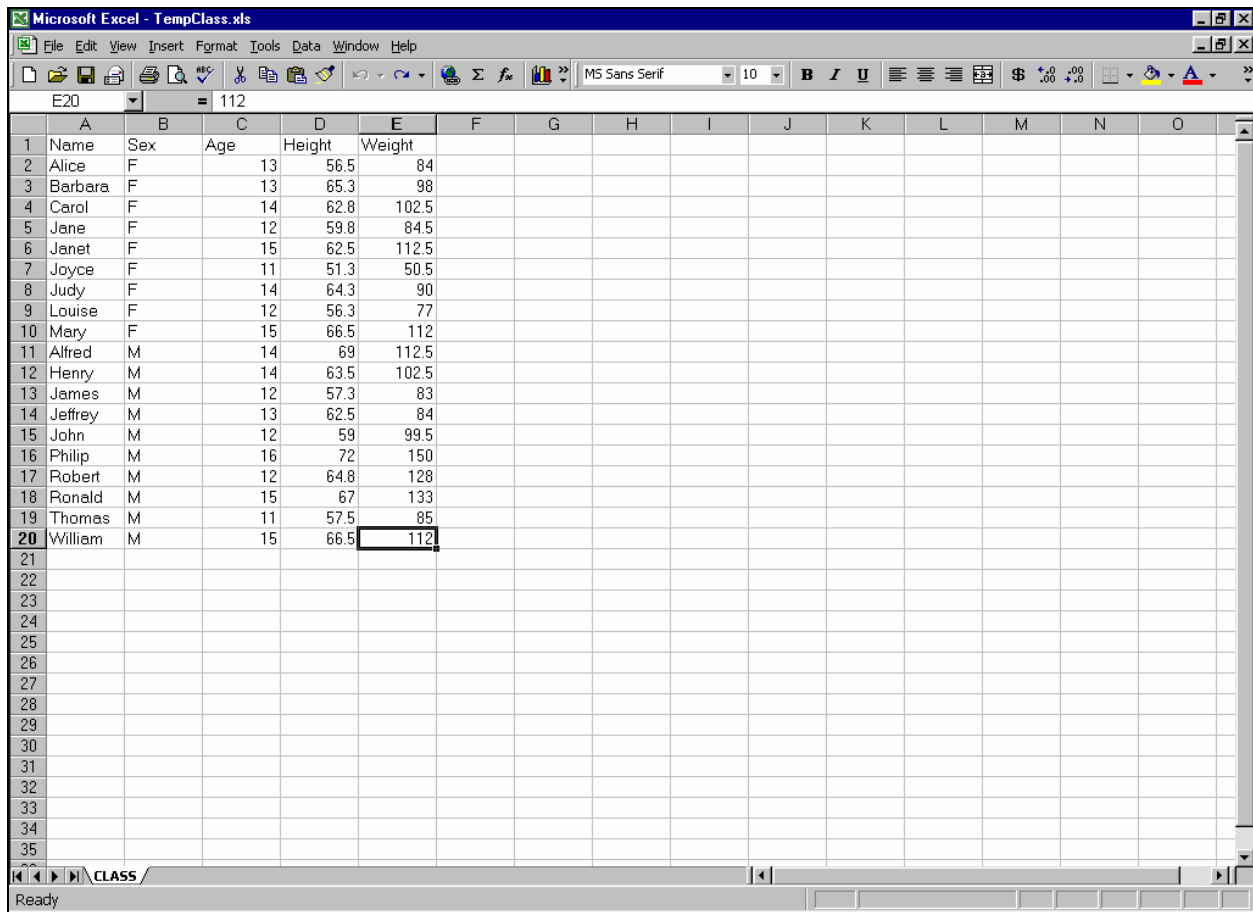
```
data _null_;
file DDEcmds;
put '[select.special(11)]'; /* select the last cell of worksheet */
put '[patterns(1,,3,true)]'; /* color it red */
put '[select("R1C1")]'; /* move the highlight to the first cell
                           to see the color with no halo */
run;
```

	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															

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):



Microsoft Excel - TempClass.xls

File Edit View Insert Format Tools Data Window Help

MS Sans Serif 10 B I U

E20 = 112

	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															

Ready

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 window titled "Microsoft Excel - TempClass.xls". The formula bar displays "= 'Lbs./Yr.'". The worksheet contains a table with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Name	Sex	Age	Lbs./Yr.	Height	Weight									
1	Alice	F	13		56.5	84									
2	Barbara	F	13		65.3	98									
3	Carol	F	14		62.8	102.5									
4	Jane	F	12		59.8	84.5									
5	Janet	F	15		62.5	112.5									
6	Joyce	F	11		51.3	50.5									
7	Judy	F	14		64.3	90									
8	Louise	F	12		56.3	77									
9	Mary	F	15		66.5	112									
10	Alfred	M	14		69	112.5									
11	Henry	M	14		63.5	102.5									
12	James	M	12		57.3	83									
13	Jeffrey	M	13		62.5	84									
14	John	M	12		59	99.5									
15	Philip	M	16		72	150									
16	Robert	M	12		64.8	128									
17	Ronald	M	15		67	133									
18	Thomas	M	11		57.5	85									
19	William	M	15		66.5	112									
20															
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
	Name	Sex	Age	Lbs./Yr.	Height	Weight									
1															
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.fill1("=RC[+2]/RC[-1]","R2C4:R20C4")]';
run;
```

Some (Expected) Error Messages Appear:

The screenshot shows a Microsoft Excel spreadsheet titled 'TempClass.xls'. The formula bar displays '=F2/C2'. The spreadsheet contains 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	#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:

	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			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															

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:

	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		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	Q
	Name	Sex	Age	Lbs./Yr.	Height	Weight											
1	Alice	F	13	6.5	56.5	84											
2	Barbara	F	13	7.5	65.3	98											
3	Carol	F	14	7.3	62.8	102.5											
4	Jane	F	0	N/A	59.8	84.5											
5	Janet	F	15	7.5	62.5	112.5											
6	Joyce	F	11	4.6	51.3	50.5											
7	Judy	F	14	6.4	64.3	90											
8	Louise	F	0	N/A	56.3	77											
9	Mary	F	15	7.5	66.5	112											
10	Alfred	M	14	8.0	69	112.5											
11	Henry	M	14	7.3	63.5	102.5											
12	James	M	0	N/A	57.3	83											
13	Jeffrey	M	13	6.5	62.5	84											
14	John	M	0	N/A	59	99.5											
15	Philip	M		N/A	72	150											
16	Robert	M	0	N/A	64.8	128											
17	Ronald	M	15	8.9	67	133											
18	Thomas	M	11	7.7	57.5	85											
19	William	M	15	7.5	66.5	112											
20																	
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	Q
	Name	Sex	Age	Lbs./Yr.	Height	Weight											
1																	
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:

The screenshot shows a Microsoft Excel spreadsheet titled "TempClass.xls". The spreadsheet contains a table with 6 columns: Name, Sex, Age, Lbs./Yr., Height, and Weight. The data is organized into rows, with the first row (row 1) serving as the header. The table lists 20 students, with their names, genders, ages, and measurements. The data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	Name	Sex	Age	Lbs./Yr.	Height	Weight											
1																	
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 David Johnson, the SUGI Application Development Section Chair, 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