

Paper 003-2011

## Automatically Generating Highly Customized Excel® Reports with SAS/Macro® and Integration Technologies (DDE)

Helen Fu, Meghal Parikh, Elayne Reiss, and Sandra Archer  
University of Central Florida, Orlando, Florida

### ABSTRACT

As a fast growing large public metropolitan research institution, the Office of University Analysis and Planning Support at the University of Central Florida (UCF) is challenged with transforming very large data sets into information that will support strategic planning and decision making. This paper describes a process that utilizes SAS® macro language and integration technologies with Microsoft Excel®, to generate a series of highly customized Excel reports in an automated approach.

This methodology allows SAS to process a data set of over 400,000 records, consolidate the data, open pre-formatted Excel templates, and write the data to designated data ranges. The integration technology used, called Dynamic Data Exchange (DDE), reduces report generating time by up to 90%, compared to prior methods that used Excel PivotTables and manual copying and pasting of results, while improving report accuracy by reducing human error. Additionally, SAS macro language enables programmers to customize the report to suit the needs of a variety of end users.

### INTRODUCTION

As a rapidly growing large public metropolitan research university, the largest public institution in Florida and the second largest nationwide, the University of Central Florida (UCF) is committed to offering the best undergraduate education in Florida. In order to better understand the dynamic marketplace of collegiate enrollment, UCF performs a variety of analyses and produces many reports for institutional leadership.

For example, UCF uses the National Student Clearinghouse StudentTracker data system to track subsequent enrollment of students who were admitted but not enrolled, as well as denied students. The data system provides UCF with data records that describe subsequent terms of enrollment on all of the student records that are queried. Generating useful reports out of this rich data source requires a process that takes into consideration several key factors: the ability to process a large data set of more than 400,000 records and more than 300 variables, the ability to generate similar reports without much repetitive effort, a quick turnaround time, and good integration with Microsoft Excel products.

SAS is powerful software for large-scale data manipulation and processing. At UCF, senior management is often provided analysis in Excel so that they get the flexibility of using the data and manipulating the charts. There are a number of ways to export SAS data to Excel, including PROC EXPORT, ODS, ODBC, and OLE DB. Nevertheless, these methods do not allow data to be loaded into a report format that was as highly customized to meet our needs.

Before using the SAS DDE technique, the data were compiled by VBA in Excel. The process was not only time-consuming, but also error-prone. By using SAS macro language and Dynamic Data Exchange (DDE), our office was able to process the data and make the data available on the web within 5 business days. In this paper, we will concentrate on the design of the Excel templates and the SAS DDE technique.

### DYNAMIC DATA EXCHANGE PROCESS OVERVIEW

Introduced in 1987, the Dynamic Data Exchange (DDE) protocol has been in use since Microsoft Windows® 2.0. It is a data communication protocol which uses the Windows internal messaging layer for communicating and sharing commands and data within, as well as among, different Windows applications. A prominent place where the DDE communication protocol is utilized involves the copy, cut and paste functions in the Windows environment where multiple applications “talk” to each other using the underlying protocol of DDE.

Newer technology such as ActiveX COM (Component Object Model) has replaced DDE functionality; however, some applications still support the use of DDE as it provides easy communication from other applications including Microsoft Excel and Word®. According to Koen Vyverman, DDE “allows a SAS session to take control of the Excel application, and perform manipulation of cells, spreadsheets and entire workbooks just as if you were doing so manually.” This technique is especially useful when one has multiple data sets for which the same analysis is required and results should be presented through a consistent Excel template.

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

The DDE client/server model is presented in Figure 1. The DDE communication process encompasses a "DDE Client" and a "DDE Server". Windows applications such as Word or Excel may function as a DDE server. A SAS program can operate as a DDE client to send data and commands to any DDE server.

A DDE server application (like Excel) can be invoked using the X command within SAS, which is serving as the DDE client. The X command is a SAS-based Windows Management command. It helps SAS programmers perform Windows commands such as opening or closing a file, or to launch an application like Word or Excel. For example, this command can be used to shut down the system from the SAS environment; the command would be:

```
X C:\Windows\system32\shutdown.exe".
```

A DDE protocol uses a FILENAME statement to initiate a client/server DDE link. This statement defines the output location and helps issue commands during the "conversation." An argument called 'DDE Triplet' is also issued with the filename. The DDE triplet is application dependent and is different for every application that is run.

The DDE triplet argument consists of the following:

1. **Application** – The executable filename of the server application, in this case Excel
2. **Topic** – Microsoft defines the topic as 'System' to enable SAS code to send commands to Excel. System is a special topic name which helps the programmer to execute commands in the server application. By using 'System' as a topic name a SAS programmer can use the INPUT statement for input from an application and the PUT statement to issue commands to the server application from SAS.
3. **Item** – Typically the range of conversation (like specific cells in Excel) between SAS and the application

Each DDE conversation is restricted to the *application* and *topic* requested by the client at initiation. Once the client/server link is established, the PUT statement may be used to send data and commands to Excel. For example, the PUT statement may be used to execute Excel macro commands. 'Close' and 'save as' file commands in Excel can also be controlled and executed using the PUT statement, giving the user complete control of the output in the server application from the SAS environment itself. In the SAS code, the PUT statement is usually embedded in a null data step.

## DDE EXAMPLE WITH SYNTAX AND TECHNICAL DETAILS

To manually generate a report in Excel, an analyst takes the following steps: design the report template, prepare the data in SAS, open Excel, enter the data into Excel, save the file, and then close Excel. The SAS DDE process follows a similar procedure, except that the DDE process takes an additional step of initiating the link between the client and server, as demonstrated in Figure 2.

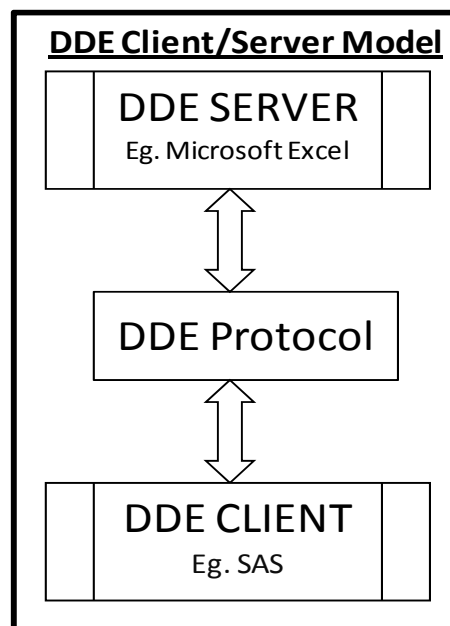


Figure 1: DDE Client/Server Model

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

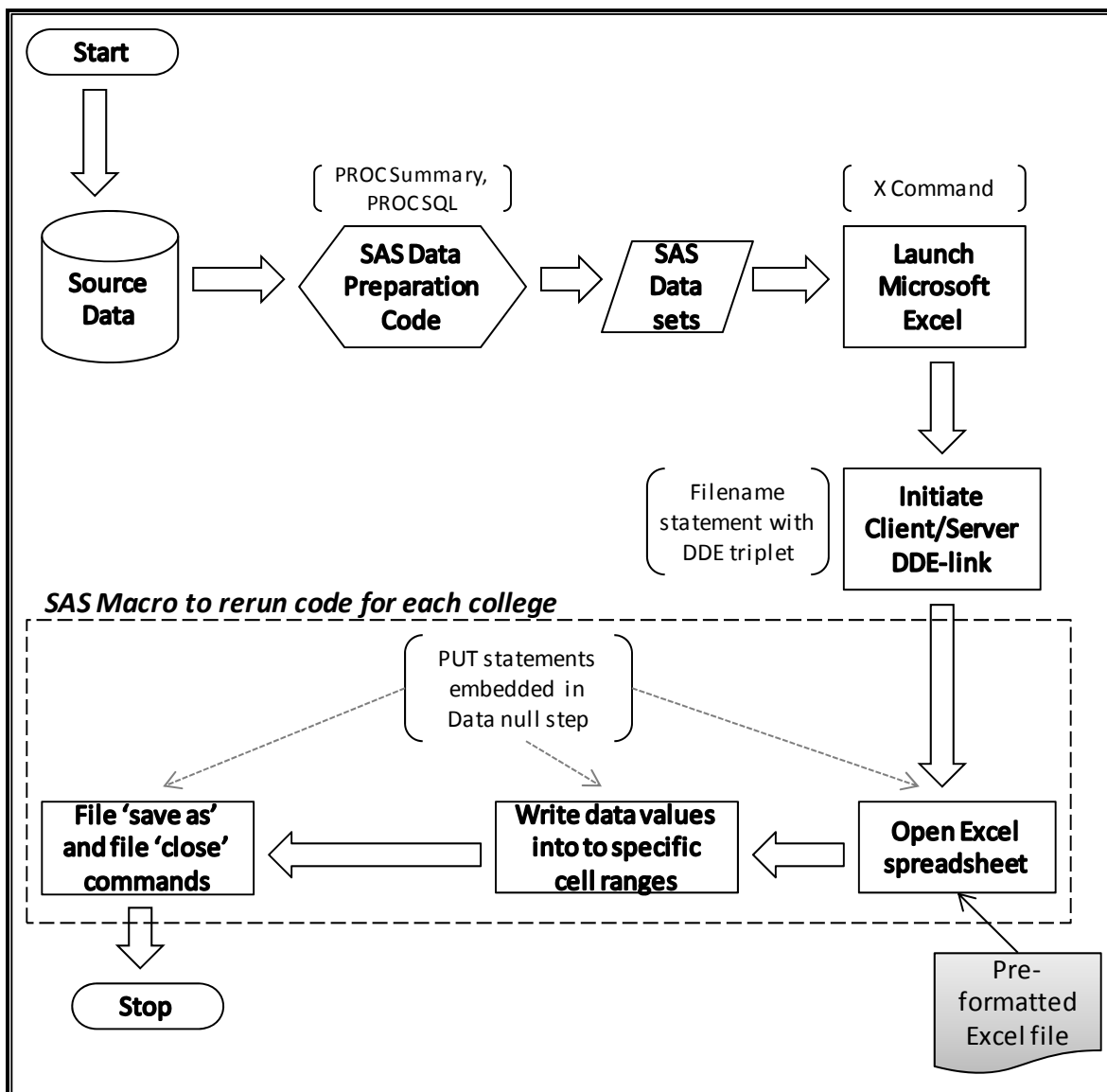


Figure 2: Process Overview

Before designing a SAS DDE process similar to this one, an analyst needs to consider the trade-offs. A few questions may be asked, to help with the decision. Does the process include a large number of similar reports that need to be generated? Do the reports share the same or similar templates? Is the process repetitive? Is the process going to be useful in future analysis? Is the project long-term in nature? If the answers are 'yes' to some of the questions above, the DDE process can be helpful.

The following sections provide a detailed description of the steps used to implement the process.

## TEMPLATE DESIGN

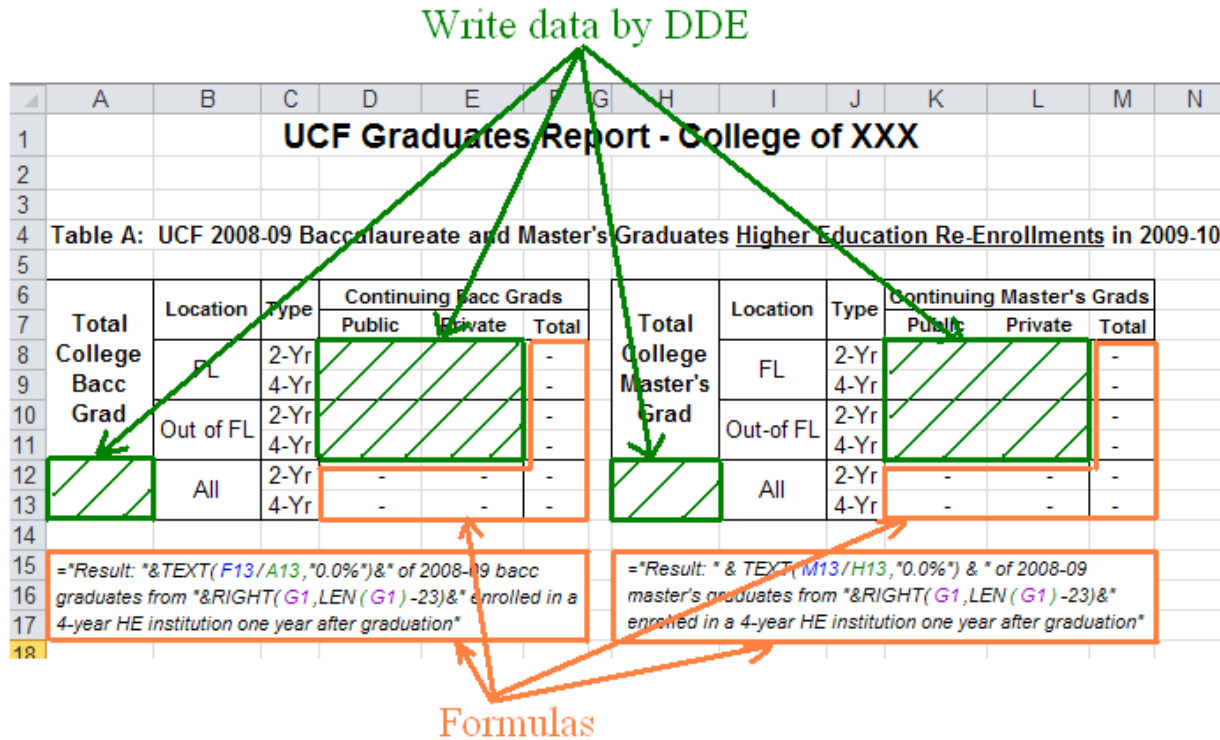
Before coding the DDE process, Excel report templates must be designed.

An example of a DDE-based process used at the University of Central Florida is the National Student Clearinghouse StudentTracker project. Every August, a report is created for each of the ten colleges at UCF. The process involves a large number of reports, all sharing the same template. The project not only requires a quick turn-around time, but also consistency and accuracy of data. Therefore, DDE process serves as a perfect tool for this particular project.

Figure 3 illustrates the first table of the template. This particular report is designed to study higher education re-enrollment among baccalaureate and master's graduates of a particular college. The DDE process automatically writes data to the green shaded areas, while the cells in the orange shaded areas contain formulas that are

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

subsequently evaluated and completed. The L-shaped areas contain formulas, which automatically calculate the subtotals and totals. The two rectangular shaped formula areas contain a concatenation of plain text and calculated figures, serving as notes for the information in the tables above. This table helps to demonstrate the powerful combination of SAS-driven DDE to process the heavier figures and the capability in Excel to create meticulously spaced and formatted layouts, along with the calculation of consistently-used formulas.



**Figure 3: DDE Report Template Design – Formula Writing**

In addition to integrating formula calculations, Excel features conditional formatting that can be used to improve the readability of the data tables. Figure 4 illustrates how conditional formatting is used in the template design.

In the template, columns A-C contain the major for UCF graduates, columns D-F contain a major track where applicable, columns H-M display the number of students in each track, and among those graduates, the number of those who re-enroll in higher education institutions within one year after graduation.

In order to increase the readability, the lines are gray/white shaded; in addition, there are borders to separate each major. Therefore, the rows in the table may have four different statuses: white row without border, gray row without border, white row with border, gray row without border. In order to automatically format the area based on the value of the columns, four different values may be assigned to a row, and the conditional formatting in Excel is based on the assigned value.

Column O decides the color shade, and is hard-coded. The gray shaded rows are assigned a value of 1, and the white rows are assigned a value of 0.

Column P decides the border by using a formula. If the major in column A in the current row is different than the previous row, then the row value is assigned 4; otherwise, the row value is assigned 2. The larger value represents that there will be a border; while the smaller value represents that there will not be a border.

Column Q is the sum of column O and P, and serves as the rule value for conditional formatting. As shown in Figure 4, the possible values for column Q are 2, 3, 4, and 5, and each value represents the row format for columns A-M.



Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

**Syntax:** `filename <fileref> dde '<server app>|system';`

**Example:** `filename commands DDE 'Excel|SYSTEM';`

### Open Excel Template

Once Excel has been launched and the DDE client/server link established between SAS and Excel, commands may be written from SAS to Excel through the fileref established in the DDE triplet. In our example, the fileref is "commands." In the null data step, the PUT statement is used to send the file open command to the Excel system.

**Syntax:**

```
data _null_;
  file <fileref>;
  put "[open("<file path and name>")]";
run;
```

**Example:**

```
data _null_;
  file commands;
  put "[open C:\template.xlsx]";
run;
```

### Initiate Another Client/Server DDE Link

To write data from SAS to Excel, another DDE triplet is written. Here, the server application is "Excel", the topic is the desired sheet name, and the item is the data range to which we would like to write data from SAS. In this example, the file name is "template", the sheet name is "GradReport", and the data range is r8c4:r11c5.

	A	B	C	D	E	F
5						
6				Continuing Bacc Grads		
7	Total	Location	Type	Public	Private	Total
8	College Bacc Grad	FL	2-Yr			-
9			4-Yr			-
10		Out of FL	2-Yr			-
11			4-Yr			-
12		All	2-Yr	-	-	-
13			4-Yr	-	-	-
14						

Figure 5: Data Range

**Syntax:**

```
filename <fileref> dde '<server app>|<topic>!<item>';
```

**Example:**

```
filename mysheet DDE "Excel|[template.xlsx]GradReport!R8C4:R11C5" NOTAB LRECL=3000
```

The "notab" option prevents SAS from converting tabs in Excel to blanks. The "LRECL =" option specifies the record length (in bytes). In our case, it was given a large value, so that all the actual length of our characters is less than the value. With these two DDE triplets executed, "commands" now refers to the Excel system, and "mysheet" refers to the cells in the specified sheet and tab.

### Write information to Excel sheets

After establishing the connection between SAS and Excel, data are ready to be written. The code below uses a PUT statement in a null data step to write data directly to the specific Excel worksheet cell range. In this example, "BS" is associated with a backspace (-1), and "T" with a tab ('09'x is its hexadecimal code).

**Syntax:**

```
data _NULL_;
  retain BS (-1) T '09'X;
  set <dataset>;
  file <fileref>;
  put
    <variable name1> T+BS
    <variable name2>;
run;
```

**Example:**

```
data _NULL_;
  retain BS (-1) T '09'X;
  set work.data;
  file mysheet
  put
    public T+BS
    private
```

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

## Close Excel

In this step, the new file is generated and closed. If the file already exists, Excel will first ask the user whether to overwrite the file.

### Syntax:

```
data _null_;
  file commands;
  put "[Save.as("<file path\name>")]";
  put "[File.Close()]";
run;
```

### Example:

```
data _null_;
  file commands;
  put "[Save.as("C:\report.xlsx")]";
  put "[File.Close()]";
run;
```

## Results

Figure 6 illustrates the result for the spreadsheet presented in Figure 3. When SAS DDE writes the data to the template, all the pre-existing data, formulas and cell formats on the template remain unchanged. The column width, row height, data font and color are all kept the same as in the template. As soon as the data are written, all the formulas on the template automatically evaluate. Sometimes, the length of the data or character may exceed the row height or column width. It is important to check the final contents of the report to ensure that the results fit properly within the template.

SAS DDE automatically writes the data

UCF Graduates Report - College of Sciences												
Table A: UCF 2008-09 Baccalaureate and Master's Graduates Higher Education Re-Enrollments <sup>(1)</sup> in 2009-10												
Total College Bacc Grad	Location	Type	Continuing Bacc Grads			Total	Total College Master's Grad	Location	Type	Continuing Master's Grads		
			Public	Private	Total					Public	Private	Total
1,867	FL	2-Yr	76	-	76	165	FL	2-Yr	6	-	6	
		4-Yr	278	69	347			4-Yr	32	2	34	
	Out of FL	2-Yr	7	1	8		Out of FL	2-Yr	-	-	-	
		4-Yr	53	60	113			4-Yr	8	-	8	
1,867	All	2-Yr	85	1	84	165	All	2-Yr	6	-	6	
		4-Yr	331	129	460			4-Yr	40	5	45	
Result: 24.6% of 2008-09 bacc graduates from College of Sciences enrolled in a 4-year HE institution one year after graduation						Result: 27.3% of 2008-09 master's graduates from College of Sciences enrolled in a 4-year HE institution one year after graduation						

The comments automatically populate as SAS writes the data

Figure 6: Result (1) - Template with Formula

Figure 7 illustrates the result of the conditionally formatted Excel template depicted in Figure 4. For example, column A indicates that row 78 and 79 both address psychology majors, while row 80 addresses social sciences majors. Therefore, based on the formulas, the values for cells Q78, Q79 and Q80 are 2, 3, and 4, respectively. Based on the conditional formatting described in the process depicted by Figure 4, the background colors of the cells are white, gray and white, and there is a cell border above row 80 to separate the different majors. Column O, P and Q, used strictly for conditional formatting indicators, may be hidden in the final report before distribution.

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

UCF Major	UCF Track	Baccalaureate Graduates			Master's Graduates			color	border	code
		Total Graduates	Continuing Students	%	Total Graduates	Continuing Students	%			
Bus Mgmt & Admin Serv	Actuarial Sci	11	-	0.0%	-	-		1	2	3
Communications	Advertising	119	15	12.6%	-	-		0	4	4
Communications	Journalism	37	6	16.2%	-	-		1	2	3
Communications	Mass Comm	265	24	9.1%	34	5	14.7%	0	2	2
Communications	Radio & TV Broadcast	86	2	2.3%	-	-		1	2	3
Letters	Rhet Speech & Comm	13	1	7.7%	-	-		0	4	4
Life Sciences	Biology, General	133	43	32.3%	17	6	35.3%	1	4	5
Life Sciences	Biomolecular Science	-	-		-	-		0	2	2
Life Sciences	Conservation Biology	-	-		-	-		1	2	3
Mathematics	Appl Math/Math Sci	-	-		8	6	75.0%	0	4	4
Mathematics	Math, General	17	8	47.1%	-	-		1	2	3
Mathematics	Statistics	3	3	100.0%	18	2	11.1%	0	2	2
Multi/Interdisc. Studies	Behavioral Sciences	-	-		-	-		1	4	5
Multi/Interdisc. Studies	Modeling & Sim	-	-		13	3	23.1%	0	2	2
Physical Sciences	Chem Sci/Indus Chem	-	-		2	-	0.0%	1	4	5
Physical Sciences	Chemistry	9	7	77.8%	-	-		0	2	2
Physical Sciences	Physics	8	5	62.5%	4	1	25.0%	1	2	3
Protective Services	Forensic Science	36	12	33.3%	2	1	50.0%	0	4	4
Psychology	Clinical Psychology	-	-		14	2	14.3%	1	4	5
Psychology	Indus/Org Psych	-	-		11	1	9.1%	0	2	2
Psychology	Psychology, General	761	232	30.5%	3	3	100.0%	1	2	3
Social Sciences	Anthropology	72	14	19.4%	11	3	27.3%	0	4	4
Social Sciences	Poli Sci & Govt	249	76	30.5%	12	4	33.3%	1	2	3
Social Sciences	Social Sciences, General	14	4	28.6%	-	-		0	2	2
Social Sciences	Sociology	34	8	23.5%	16	8	50.0%	1	2	3
<b>Total Cont. Students</b>		<b>1,867</b>	<b>460</b>	<b>24.6%</b>	<b>165</b>	<b>45</b>	<b>27.3%</b>	<b>0</b>	<b>4</b>	<b>4</b>

The row format is based on the value of column Q

Automatically based on formulas

Figure 7: Result (2) – Conditional Formatting

SAS Macro Language and SAS DDE

As discussed previously, one of the motivations for developing a SAS DDE process is that a large number of similar reports need to be generated. This section gives an example on how SAS macro language is coded in conjunction with SAS DDE to automatically generate several similar reports.

Suppose that the Excel template is designed as in Figure 4, and the data are summarized in a data set "temp\_sum", as shown in Figure 8. The data set contains the following information: student major, major track, number of baccalaureate graduates, number of baccalaureate graduates who re-enroll within one year after graduation, number of master's graduates, and the number of master's graduates who re-enroll within one year after graduation.



Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

VIEWTABLE: Work.Temp_sum							
	Major	Track	College	B_grad	B_cont	M_grad	M_cont
1							
2	Area and Ethnic Studies	Women's Studies	CAH				
3	Bus Mgmt & Admin Serv	Accounting	CBA	318	112	88	4
4	Bus Mgmt & Admin Serv	Actuarial Sci	COB	11			
5	Bus Mgmt & Admin Serv	Bus Adm & Mgmt	CBA	343	43	2	
6	Bus Mgmt & Admin Serv	Bus Manag Eco	CBA	49	8	5	2
7	Bus Mgmt & Admin Serv	Bus Market Mgmt	CBA	355	23		
8	Bus Mgmt & Admin Serv	Business, General	CBA	225	22	246	9
9	Bus Mgmt & Admin Serv	Finance, General	CBA	402	59		
10	Bus Mgmt & Admin Serv	Hosp Adm/Mgmt	HSPMG	476	30	24	2
11	Bus Mgmt & Admin Serv	MIS/Bus Data Proc	CBA	98	13	25	2
12	Bus Mgmt & Admin Serv	Non-Prof Mgmt	COHPA			37	7
13	Bus Mgmt & Admin Serv	Real Estate	CBA	26	5		
14	Bus Mgmt & Admin Serv	Taxation	CBA			20	
15	Bus Mgmt & Admin Serv	Travel & Tour Mgmt	HSPMG	105	5		
...	...	...	...	...	...	...	...
121	Visual & Perf Arts	Photography	CAH	17	1	.	.
122	Visual & Perf Arts	Studio/Fine Art	CAH	70	10	10	.
123	Visual & Perf Arts	Visual Art, General	CAH	42	7	.	.

Figure 8: Example Data Set

In the following sample code, we can see the DDE code is called by the SAS macro procedure to produce multiple reports, one for each college.

```

/*****Sample code to create multiple report using SAS DDE and MACRO*****/

%let AY = 2009; *specifying the year of report for global variable AY;

%let template="C:\SASGF\DDE\GradReport"; *specifying the directory path and file name
                                         for the Excel template;

%let output= C:\SASGF\DDE; *specifying the directory path where the output Excel
                           reports will be stored;

libname DDEPaper 'C:\SASGF\DDE';

/*launch Microsoft Excel*/
options noxwait noxsync obs=max;
X "C:\Program Files\Microsoft Office\Office11\Excel.exe";

/*initiating client/server DDE link using the filename commands in the DDE Triplet*/
FILENAME commands DDE 'Excel|SYSTEM';

/*creating a macro named report which will be called for each college */
%macro report(college,title,filename);

    data _null_; *null data step to open pre-formatted Excel template;
        file commands;
        ddeopen="[open("&template..xlsx")]";
        PUT ddeopen;
    run;

    data Table_C; *data step to select data specific to this college;
        set DDEPaper.temp_sum;
        option missing=0;
        where College=&college.;
    run;

```

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

```

*initiate DDE link to cells in the Excel template using file name mysheet;
FILENAME mysheet DDE "Excel|[GradReport.xlsx]GOI!R59C1:R81C9" NOTAB LRECL=3000;

data out_2; *null data step to write data into columns A, D, H, I;
  retain BS (-1) T '09'X;
  set Table_C;
  file mysheet;
  put    Major T+BS T+BS T+BS
        Track T+BS T+BS T+BS T+BS
        B_Grad T+BS
        B_cont T+BS;
run;

*initiate DDE link to a different set of cells in the Excel template;
FILENAME mysheet DDE "Excel|[GradReport.xlsx]GOI!R59C11:R81C12" NOTAB LRECL=3000;

data out_3; *null data step to write data into columns H and I;
  retain BS (-1) T '09'X;
  set third;
  file mysheet;
  put    M_Grad T+BS
        M_cont ;
run;

data _null_; *save the Excel file with the name of the college and year;
  file commands;
  put "[Save.as("&output.&filename.&AY")]";
  put "[File.Close()]" ;*close Excel;
run;

/*end the macro */
%mend report;

/*Call the SAS macro "report" for each college passing the parameters of college name,
report title and file name */
%report('COS', 'UCF Graduates Report - College of Sciences', GradReport_COS);
%report('COM', 'UCF Graduates Report - College of Medicine', GradReport_COM);
%report('CON', 'UCF Graduates Report - College of Nursing', GradReport_CON);
%report('EDUC', 'UCF Graduates Report - College of Education', GradReport_COE);
%report('UGST', 'UCF Graduates Report - Undergraduate Studies', GradReport_UGST);

```

## CONCLUSION

Using the power of SAS macro language, the challenge for creating recurring, uniform Excel-based reports with required multiple data sets and tables can be overcome. This paper demonstrated how a wide array of commands, ranging from open and save functions, to writing data to specific cells in Excel can be performed with straightforward DDE commands using PUT statements written in SAS data steps. Similarly, DDE can be used to pull data from specific cells in Excel into the SAS system through INFILE and INPUT statements, or run more complex system commands such as creating Excel tabs or calling VBA for Excel macros. In this example project presented in this paper, analysts at the University of Central Florida are able to provide a series of highly customized Excel reports to institutional leadership in a very quick timeframe. Through the use of SAS, analysts can focus on sharing and drawing conclusions from the data, rather than spending an excessive amount of time exporting and formatting reports.

Automatically Generating Highly Customized Excel Reports with SAS/Macro and Integration Technologies (DDE), continued

## REFERENCES

- Chen, Chienchung. 2010. "Generating Reports With DDE and VBA, the Concept - Alternative to ODS to Produce Complex Reports". *Proceedings of the SAS Global Forum 2010, Seattle, WA*. Available at <http://support.sas.com/resources/papers/proceedings10/074-2010.pdf>
- Derby, Nathaniel. 2007. "Revisiting DDE: An Updated Macro for Exporting SAS Data into Custom-Formatted Excel Spreadsheets Part I - Usage and Examples". *Proceedings of the WUSS Fifteenth Annual Conference, San Francisco, CA*. Available at [http://www.wuss.org/proceedings07/DataPresentations&BusIntell/DPBI\\_Derby\\_RevisitingDDEI.pdf](http://www.wuss.org/proceedings07/DataPresentations&BusIntell/DPBI_Derby_RevisitingDDEI.pdf)
- Helbers, Lawrence and Vinokurov, Alex. 2007. "SAS Output to Excel: DDE and Beyond". *Proceedings of the Northeast SAS Users Group, Portland, ME*. Available at <http://www.nesug.org/proceedings/nesug02/ad/ad007.pdf>
- Li, M, Archer, S., Denslow, R. 2003. "Using DDE and SAS/MACRO® for Automated Excel Report Consolidation and Generation". *Proceedings of the Southeast SAS Users Group, St. Pete Beach, FL.* Available at <http://analytics.ncsu.edu/sesug/2003/DM08-Li.pdf>
- SAS Institute, Inc. May 2010. *SAS 9.2 Companion for Windows, Second Edition*. Page 275 – 284. Cary, NC. SAS Publishing.
- Symes, Rebecca. 2005. "Using DDE to connect SAS results with MS-Excel". *Proceedings of the Northeast SAS Users Group, Baltimore, MD*. Available at <http://www.nesug.org/proceedings/nesug05/pos/pos15.pdf>
- Vyverman , Koen. 2002. "Creating Custom Excel Workbooks from Base SAS with Dynamic Data Exchange: a Complete Walkthrough." *Proceedings of the 27th SAS Users Group International Conference, 2002*. Available at <http://www2.sas.com/proceedings/sugi27/p190-27.pdf>

## CONTACT INFORMATION

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

Helen Fu, Meghal Parikh, Elayne Reiss, and Sandra Archer  
University of Central Florida  
12424 Research Parkway, Suite 215  
Orlando, FL 32826  
Phone: (407)882-0285  
Fax: (407)882-0288  
[uaps@ucf.edu](mailto:uaps@ucf.edu)  
<http://uaps.ucf.edu>

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.