SAS® and Microsoft Office: Tales From the Trenches
John W. Davison, Jr., Factotum, Inc., Leesburg, VA

ABSTRACT
The popularity of Microsoft Office and especially Excel has stimulated interest in methods for moving data between the two environments. This in turn has given rise to papers on the topic at recent SUGI/SGF and regional SUG meetings. These have treated various approaches in different contexts, and rely in general on the SAS® Output Delivery System for reporting and SAS/Access® to access Excel workbooks as SAS® libraries to move data in either direction. This paper reports the author's experience with these over the past six years; particularly in situations where the governing consideration was to use new SAS® features to simplify applications developed using DDE and OLE while matching the appearance of existing results. It does not pretend to offer a unified approach that meets all needs but presents several approaches that have worked together with the direction the author is taking, and considerations for improving the methodology.

INTRODUCTION
The SAS Output Delivery System, SAS/Access®, and SAS Add-In for Microsoft Office make it easy to direct the output of SAS procedures to Microsoft Office documents, and even to use Excel workbooks as SAS data libraries. These facilities provide much the best approach for such purposes when developing a new application; however, the situation often arises where an existing application already generates such output, either because it was developed using older SAS features or because it is being converted to SAS from another source., and one of the design constraints is that the new application must conform to the look and feel of the existing one. It seems more usual than not that the such application output has been laid out in a more or less free hand manner in the target application, usually Excel or Word. In such a case it is usually difficult at best to write SAS code or ODS templates to reproduce the desired appearance and the most practical approach then appears to be to create a model of the target document and to insert the elements generated by SAS into their appointed places. This has been done most often through DDE or OLE. This paper presents cases illustrating the author’s attempts using different approaches to move data in larger blocks with some success.

The examples that follow do not pretend to be authoritative guides to the best approach. Each addresses a particular report and was limited by the need to stop when it was good enough for the needs of the moment, although lessons drawn from each were applied to the next where possible. These lessons constituted first efforts at a systematic approach. It is hoped that this paper will stimulate others to consider the issues presented, seek answers to their own questions and thus contribute to developing a discipline for such reports.

CASE I: OUTPUT TO MICROSOFT WORD
A few years ago the author was responsible for producing operational and financial reports for PSINet, Inc. as the company struggled to contain costs, manage its assets and in particular improve collections in an effort to recover from bankruptcy proceedings. Of these reports one of the most important was a list of all the accounts in arrears, produced weekly as a Microsoft Word document that was torn apart and distributed to the collections agents. The content was just a SAS dataset of delinquent accounts. Since each agent was compensated in part on the basis of the number and dollar amount of collections the rows were grouped by means of a SAS DATA step so that each agent had a comparable number of accounts to handle and to equalize as much as possible both the number of large and small accounts and the total amount due assigned to each agent for collection. The grouping was done by assigning serial codes and the report produced by a simple PROC PRINT using PAGE BY the group code. The practical problem in reproducing the report was that the quantity of information so crowded the margins of a 9 by 14 inch page that the columns folded. This made the report difficult to read and so unacceptable to the collections manager. On the other hand the report could be edited by hand to adjust the margins so they fit exactly in the space allowed by the printer but doing so took longer than the manager would tolerate. The fix turned out to be very simple, so much so that it would hardly merit presenting here except to demonstrate that simple solutions should never be ignored because one seeks a clever one. It turned out that all that was necessary was to direct the SAS ODS output to a Microsoft Word document and open it; highlight and copy (CTRL-C) the new output from the beginning of the first line to the end of the last, some 60 or 70 pages worth; then open a copy of the previous week’s report already formatted to fit the printer margins and highlight there the corresponding text; and Paste Special the unformatted text into the formatted space. While it might have been elegant to create a Word macro to would make the manual step unnecessary this procedure accomplished in five minutes as much as was required.
CASE II: SIMPLE OUTPUT TO A MICROSOFT EXCEL WORKBOOK

The Office of the Comptroller of the Currency (OCC) charters all national banks and shares oversight responsibility for the banking system with the Board of Governors of the Federal Reserve System (FRB) and the Federal Deposit Insurance Corporation (FDIC). One of the duties of the Comptroller is to report to the Congress on the health of the system; most of the information in these reports comes from the quarterly Report of Condition or Call Report, required of all banks doing business in the United States and collected by the FRB for use by all three agencies. From this data the OCC analysts derive a wide variety of reports.

One such report, published regularly, shows aggregates of assets, liabilities, income, expenses and various performance indicators of national banks, produced as an Excel workbook of over a dozen sheets. Another sheet contains control buttons to select pages for display and submit them for printing. Each data page is simple, with some 50 or 60 rows and several columns.

The first three sheets show the assets and liabilities for the current quarter and income and expenses for both the current quarter and the year to date, in each case compared to the same period for the previous year. The first column is the row labels containing text with indentations, type faces and sizes that were specified when the report was designed years ago, all interspersed with blank spacer rows. The second and third columns are the values for the current and previous periods, the third and fourth columns are the difference and the percentage change between the current and the previous period, and the last column contains the names assigned the items in the Interactive Banking Information System (IBIS), developed at the OCC for analysts working with Call Report and other banking data. The remaining sheets show details of the various aggregated indicators summarized for all national and commercial banks, grouped by size in terms of total deposits and also for all national and all commercial banks.

Most of the content of this report such as the row labels, the IBIS variable names, and the print control sheet does not change from month to month and the page headings change only in the date of the quarter reported, as do the column headings in the first three sheets. In the latter two of the columns contain formulas for the difference and the percent change that are the same every month. Much of the fixed information, especially the controls, could not be created as SAS output; therefore an existing copy of the report was edited to create an Excel template, with all of the fixed information edited in its proper place and format and with blank cells for the data. The data for each sheet was created by a PROC TABULATE step in which a CLASS variable forced the proper sequence of the rows and dummy variables with zero values were inserted into the sequence to create blank rows for spacing as required by the original. The result was output to HTML by ODS since the author was as yet unaware of the XML tagset to create Excel workbooks. Once the output had been created both the HTML pages and the template were opened in Excel, the columns of the data sheets copied from the HTML, pasted using Paste Special (values only) into the template and saved as the new report. The cut and paste operations could be done in two multipage steps because the first three sheets each have two columns in the same positions of variable data and the remaining ten sheets each have six such columns. In the final version the date of the current quarter was edited manually in its place on the third sheet since that cell was the first that had the complete date in a convenient Excel format and copied by reference from there into its proper places on the others in various formats conforming to the original, with the date of the previous period derived by formula.

In this case again work was carried only to the point of good enough and no further effort was made to make it elegant by writing macros to replace the manual operations.

CASE III: EXTENSIVE SIMPLE OUTPUT TO A MICROSOFT EXCEL WORKBOOK

The previous example is conceptually almost as simple as the first, relying as it does on copying and pasting the data from an output file containing the desired values without the desired format into an existing file with formatted space to receive them. The next example is almost as simple in concept but takes better exploits the capacity of the computer to bring the unformatted result together with the formatted display.

The report in this example, one of many, displays indicators for groups of banks classified by common attributes such as size or business concentration. They are essentially all the same except for the classification attribute, so a single model could be used for each case. The report aggregates Call Report data for all banks in the class both quarterly and annually since 1984 producing one page per quarter and per year for each of two characteristic groups, Key Indicators and Loan Performance, together with a page of controls to select the quarter and attributes to be displayed. Thus the complete report is quite large and only parts will be exhibited to illustrate key points.

The report itself is easily generated in two PROC TABULATE steps, one BY YEAR and the other BY QUARTER, with two TABLE statements for each, one for Key Indicators and one for Loan Performance and ODS output to HTML. In its original form these were copied to the Excel workbook sheets by the same copy and Paste Special
device as in the first example. This would have been simply done by one or two very large multipage operations but for the fact that the HTML output from the PROC TABULATE steps did not leave the pages in the sequence desired in the report. The report could be made available to analysts without the formatting details of the original without concern for this through use of the navigation buttons of the control page.

The Excel macros that implement these selection buttons are somewhat more complex than those of the previous case, including among other things provision for specifying the path used by the application. The motivation for replacing the existing reports, which were adequate for their purpose, was replacement of the server on which they executed with one based on a different operating system. The Control page with its buttons programmed for the HTML access is shown as the example workbook Case_3_Map.xls together with representative pages of the HTML output from SAS.

The next section introduces an improved approach to populating the formatted report pages using the SAS output as an Excel data source, illustrated with this example revisited.

**CASE IV: COMPLEX OUTPUT TO A MICROSOFT EXCEL SPREADSHEET**

The comparatively modest success achieved in the two previous cases led to an effort to use the same methods to display a report of a model developed to test fairness in lending. The workbook contains specific banking information and details of policy that preclude its actual presentation but it can be described in general terms. The report compared the combination of descriptive statistics and SAS statistical procedures applied to loans made by a bank to all and identified groups of borrowers. It quickly became apparent that the methods described so far would require almost as much manual effort in each instance as did the cutting and pasting already in practice, since the desired cells were scattered over several pages of the output. Values were taken from a few cells in each page and inserted into cells of the report laid out to present the analysts’ key points; in many places these values were in fact edited into text strings that combined them into more complex expressions. It is possible that with more time the text manipulation needed could have been done in SAS data steps but it was clear that the project was drifting away from the goal of making the presentation easier and into the area of trying to force the result to demonstrate that it could be done.

Fortunately, at this point a conversation with Vincent del Gobbo of SAS Institute led to his suggestion of directing the SAS output to an Excel workbook using the XML-Excel ODS tagset and using that as a data source for the end result. A principal advantage of the Excel workbook was that the pages were packaged together rather than a collection of individual files that requiring manual effort to organize conveniently for editing. The editing effort itself required a considerable amount of manual effort and some imagination to create the desired report but it only had to be done once for each variant of the report, which was in fact used in a fairly large number of cases.

Apart from the sensitivity of the content of this report, its presentation as an example would tend more to show off how Excel functions can be used to derive complex results from text and cell contents. The previous example, however, is amenable to the same approach and illustrates it more simply. Once again a formatted version of the desired result is saved as a template but this time the SAS output is produced directly as an Excel workbook with its pages laid out similarly. The actual report had some 80 pages representing two criteria for each of 20 bank categories presented both annually and quarterly from 1984 to the current period. The underlying result was done with PROC TABULATE, which of course appended each new quarter automatically.

The example shown uses only two pages of the report. One workbook contains the pages as created by PROC TABULATE and the other has been saved as a template with titles, row and column headings, and cell formats in place but with formulas in the individual cells that assign them the values of the corresponding cells in the output pages. In this instance the cells match exactly so the example is quite similar to those above; however, it should be evident that if necessary they could pick and choose, as it were, from cells in pages laid out quite differently. Preparation of the template would require more effort but once done the machine takes over and does the work as often as required. The one additional bit of manual effort needed in this case is to add a column to the template each quarter, which is done on one sheet and then copied in one step to all others. One point to remember is that a workbook created in this fashion is dependent upon the continued existence of the data source. When the data source will not be changed the result can be made permanent by highlighting from beginning to end, copying it and then using Paste Special, Values Only, to put it back into the same location.

**CONCLUSIONS**

When the constraints of an existing application preclude creating SAS output in a specified format an alternative approach is to capture it into a predefined formatted template or file. This can be simply done although with some effort using ODS to create a model to serve as a data source for the import.
REFERENCES
The work described in this paper depends almost entirely on Excel rather than SAS operations. There are several excellent references for Excel; the principal one used here was:


ACKNOWLEDGMENTS
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 registered trademarks or trademarks of their respective companies.

I am particularly indebted to Vincent del Gobbo of SAS Institute for directing my attention to the use of a SAS created Excel workbook as an external data source.

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

John W. Davison, Jr.
Factotum, Inc.
15265 Montresor Road
Leesburg, VA 20176-5811
Work Phone: 703-777-7164
Email: johndavison@earthlink.net

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *