Paper 250-28

Not All Fish Eat Worms: A SAS® Programmer’s Guide to MS Excel and Other Fish Stories

David Rucker, BioCor LLC, Yardley, PA

ABSTRACT

MS Excel? What does a SAS programmer need to know about a spreadsheet application? Like any good angler who wishes to catch fish, a diverse assortment of baits and lures in the tackle box is indispensable. A programmer's skills are much the same way; an array of knowledge in many applications and languages is indispensable. It is true, not all fish eat worms. Sad, but true, not all people work with the SAS® System (these species are easily identified when asking "SAS? What does S.A.S. stand for?"). The fact is, more people are hooked on using MS Excel® to collect, store, and use data, then the ocean has fish. As SAS programmers, we need to know how to work with this application to our advantage and our customer's satisfaction and profit. This paper will be beneficial to seasoned and salty old timers and novices alike, who have, at a minimum a general understanding of MS Excel. This paper will also benefit those who are learning SAS after having mastered MS Excel. (Base SAS®, SAS/ACCESS Interface to PC File Formats®, Microsoft Windows NT®, Microsoft Excel 97®, Microsoft Excel 2000®)

INTRODUCTION

As SAS programmers, we are often required to help our customers who may not be able to look at a SAS data set electronically and visa versa. This paper will assist the SAS programmer in working with MS Excel by covering topics such as comparable functions (I can do it in the SAS System, but how do I do it in MS Excel?). This paper will also expose the tricks and snags of accessing and importing data stored with MS Excel.

The various interactions a SAS programmer can expect to encounter with MS Excel, will be discussed and how to make the two work together. The purpose of this paper is to provide the reader with a document that will be meaningful in their everyday job and that can be used as a reference document. So get ready to add a new skill to your programming tackle box and let’s go fishing!

WHAT IS MS EXCEL?

MS Excel is a spreadsheet application, a newly discovered species defined in 1982 by Merriam-Webster's Collegiate® Dictionary as an accounting program for a computer; *also*: the ledger layout modeled by such a program. Much like the SAS system, MS Excel has grown into a much larger tool than just handling accounting information. The application consists of a matrix of rows (identified by a number value) and columns (identified by an alphabetical reference). SAS programmers would be more familiar with the terms observations and variables respectively. The intersection of the columns and rows, known as cells, are where the data, labels, and formulas are stored. Each matrix is known as a worksheet, and many worksheets together in the same file are known as a workbook. Beyond the physical nature of the application, MS Excel is a tool. A tool used by many who are not necessarily familiar with SAS. MS Excel is a tool that our customers, whoever they may be, will at some point want us to use.

WHAT IS SAS?

Back in 1976, when SAS was founded, SAS was an acronym for Statistical Analysis System. While statistical applications are still offered, much more is now offered to the software's users. The acronym has since been discarded as a title of an application, and is instead being used as a brand name. SAS has since become the world leader in business-intelligence software and services, enabling customers to turn raw data into usable knowledge.

Again, beyond the marketing jargon of the products’ capabilities, SAS is a tool. A tool which is quite popular, but not as commonly used as the major Microsoft Office products. The capabilities of SAS will often be called upon by our customers to process, analyze, and report data that has been stored in a MS Excel spreadsheet or to export data from SAS data sets to MS Excel.

KNOW YOUR DATA!

Anyone who has ever closely looked at a fly-fishing fly will see the careful attention that was put into the design so that it imitates an insect. The successful fly angler will know the water temperature, the hatching timetables, the behavior of the fish, and what the fish wants. The angler understands the importance of knowing how to catch a fish. The same can be true for a programmer. Knowing your data is the most important step you can take towards becoming a successful programmer. If you are interested in methods for getting to know your data, please reference a previous paper of mine which is included in the reference section and contained in the SUGI 27 conference proceedings.

CAST & RETRIEVAL: IMPORT MS EXCEL DATA TO SAS

Have you ever seen a fish jump right into your frying pan? I would say, probably not. The same is true for “catching” data from MS Excel. Like hooking and landing a fish, certain techniques and skills must be used to ensure that the fish doesn't break the line or break loose from the hook. The same can be said about retrieving data from MS Excel. In this section, you will be presented with several techniques for making sure the data doesn't “get away”. These methods are by no means an attempt to cover every possible spreadsheet you may need to import, but instead just simple examples of code that can be utilized. Be aware that, much like fishing, you must have the proper licenses to perform some of the following methods. In all samples the following SAS products must be installed and licensed: SAS/BASE.

The following sample spreadsheet will be used for the examples that follow.

Sample Spreadsheet: c:\sugi28\temp\fishing.xls
Worksheet name: Sheet1

	A	B	C	D
1	fishname	caught	prizeamt	lbswt
2	Trout	11/1/2002	$200.00	5.2
3	Catfish	7/25/2001	$400.00	10.4
4	Sunfish	8/15/2000	$100.00	2.1

BAMBOO POLE: THE BASIC DATA STEP (EXAMPLE #1)

This method is likely the one you will use when you only have the basic SAS products installed. This method will require that the MS Excel file be saved as a comma separated file (.csv) prior to submitting the code. Much like using a bamboo pole to fish, you will be somewhat limited in your capabilities and efficiencies. The fact that the file must first be saved as a .csv file, individually list each variable, designate its type, and remove the header row(s) (column labels). Since MS Excel does not confine the user to a variable type within a column, this technique can lead to errors. Options for avoiding these errors can include selecting the appropriate SAS format for the data type you are presented. Additional options for working around data type will be presented later in the paper.

Example #1

```
LIBNAME dlib ‘c:\sugi28\temp’;
FILENAME sample1 ‘c:\sugi28\temp\fishing.csv’;

DATA dlib.sample1;
  Infile fishing dlm=’,’;
  Input fishname $
          caught    mmddyy8.
          prizeamt $
          lbswt
          ;
run;
```

FISH FINDER: PROGRAMMING WITH SAS/ACCESS TO PC FILE FORMATS (EXAMPLE #2)

This method can be utilized if the SAS/ACCESS to PC File Formats product is licensed. This method adds more technology to the process of importing data. Similar to a fish finder aiding the fisherman in determining the exact location of the fish, this option will make the import of data easier for the programmer. The ACCESS and DBLOAD procedures (the DBLOAD option is not discussed in this paper) do not support files in the Excel 97 (Version 8) format. However, you can still access Excel 97 files through the SAS Import/Export facility, which is discussed in the next option.

Example #2

```
LIBNAME dlib ‘c:\sugi28\temp’;

PROC ACCESS dbms=xls
  create dlib.sample2.access;
  path= ‘c:\sugi28\temp\fishing.xls’;
  worksheet=’sheet1’;
  scantype=yes;
  getnames=yes;
  assign=yes;
  mixed=yes;

  create dlib.sample2.view;
  select all;
run;

DATA sample2;
  set dset.sample2;
run;
```

STORE BOUGHT: SAS/ACCESS TO PC FILE FORMATS (EXAMPLE #3)

Again, this method can be utilized if the SAS/ACCESS to PC File Formats is licensed. The difference here, well, you just gave in and decided to get your fish from the store! Although I jest, many fishermen come home after investing more money on bait and tackle then it would have cost to just buy a fish. Unlike business, fishing is also a very relaxing way to spend the day and enjoy the outdoors - so catching a fish may not always be the ultimate goal. However, in business, the goal is to catch fish and to catch them in a timely manner. Therefore, do not “cast off” this method as it can be quite helpful to quickly import data. This method can also be quite helpful in that you have the option to save the code which is generated by the wizard (PROC IMPORT code will be written).

Example #3

Follow these steps:

- Create a library (example: use libname statement).
- Select FILE => IMPORT.
- Select Excel with file version.
- Click NEXT.
- Identify the file you wish to import.
- Click OPTIONS.
- Select the options that meet your example.
- Click OK.
- Click NEXT.
- Select the library to store the data set.
- Name the data set.
- Click FINISH.

BOAT FISHING: SAS/ACCESS TO ODBC (EXAMPLE #4)

This method can be utilized if the SAS/ACCESS Interface to ODBC is licensed, as well as an installed and configured Microsoft Excel ODBC driver. Like using a boat to get you to where the fish are, in this method you are using the ODBC driver to get you to the data.

Example #4

```
LIBNAME dlib ‘c:\sugi28\temp’;

PROC SQL;
  connect to odbc (‘c:\sugi28\temp\fishing.xls’);
  create table dlib.sample4
  As select * from connection to odbc
  (select * from ‘Sheet1$’);
  disconnect from odbc;
quit;
```

USING ARTIFICIAL BAIT: DYNAMIC DATA EXCHANGE (DDE) (EXAMPLE #5)

This method is similar to the DATA step procedure, instead DDE is used. If you only have SAS/BASE licensed, you will be able to use this method for importing MS Excel spreadsheets. One of the key differences with this method is that the file can (and is required) to be open prior to running the code. This method can also be quite useful in that you may invoke Visual Basic for Application (VBA) code which can be used to perform a variety of effects on the data while in MS Excel (which is beyond the scope of this paper).

Example #5

```
FILENAME sample5 dde ‘Excel|Sheet1!R1C1:R4C4’ notab;

DATA sample5;
  Infile sample5 dlm=’09’x;
  Input fishname   $
          caught      mmddyy8.
          prizeamt   $
          lbswt
          ;
Run;
```

SNAGS: WYSAAWYG

Fishermen are known for their fabrication of stories, so I will not disappoint you. Many of you may be familiar with the species WYSIWYG (What You See Is What You Get). Well when I was last using MS Excel, I landed the biggest WYSAAWYG (What You See Ain't Always What You Get) East of the Pacific Ocean. I put up a great fight before the WYSAAWYG finally broke my line and disappeared into the depths. Let me tell you, the WYSAAWYG is one tough creature. These files can be a great challenge, and if not careful will leave you with nothing but a great story about the one that got away.

Now that you have been introduced to several methods for importing the data, it's time to introduce you to some of the issues that you may be presented in your data which would cause unwanted results. So why is this an issue? It all leads back to the fact that MS Excel is not a database. The fact that each cell within a column can be formatted or entered different can cause many snags.

FORMAT CHARACTERS (EXAMPLE #6)

One of the methods presented above involved importing a .csv file. One of the issues with this method is that the .csv file is essentially a text file. The snag with using this file type, involves the inclusion of formatted value characters which are carried over into the .csv file (example '$'). In many cases, this issue can be handled using SAS formats. Another way of handling formatting issues is to remove the format from the data in MS Excel before importing the data into SAS. This can be done in MS Excel by following the instruction in Example #6.

Example #6

Follow these steps:

- Open the MS Excel file and worksheet
- Using the left mouse button, select the column of interest.
- Click FORMAT from the main menu.
- Click CELLS.
- Select the category: Number
- Change decimal places if needed.
- Click OK.

DATES

SAS and MS Excel both store dates as numeric values, however, they use different relative dates as numeric value of 0. SAS uses 1/1/1960 and Excel uses 1/1/1900. This will create a +60 year difference in the SAS date value if you remove the format from a value. One problem with dates is that you are not always sure what type of value you are viewing. It may look like a date, however if the date was entered into MS Excel as a character value, don't expect to get a character value in SAS during the import procedure. You may not know you have a date character date value just by looking at the data. One way to check what kind of data you are looking at is to change the format of the values. For date values, change the format of the column in MS Excel to a Numeric value.

Before Re-format
Sample Spreadsheet: c:\sugi28\temp\fishing.xls
Worksheet name: Sheet1

	A	B	C	D
1	fishname	caught	prizeamt	lbswt
2	Trout	11/1/2002	$200.00	5.2
3	Catfish	7/25/2001	$400.00	10.4
4	Sunfish	8/15/2000	$100.00	2.1

If by chance you have a date value which was entered as a character value, it will not convert into a numeric value. You will want to reference the XLS Chapter of the SAS OnlineDocs for more information concerning the differences between SAS and Excel for Date and Time values. As a general rule, it may be a good idea to check all numeric values in this manner to see if you have any character values mixed in with the data.

After Re-format
Sample Spreadsheet: c:\sugi28\temp\fishing.xls
Worksheet name: Sheet1

	A	B	C	D
1	fishname	caught	prizeamt	lbswt
2	Trout	37561	$200.00	5.2
3	Catfish	37097	$400.00	10.4
4	Sunfish	8/15/2000	$100.00	2.1

SINGLE QUOTES

So how is it possible that you can have a character value where it appears to be the same as all the other values? The dreaded single quote is one method. In MS Excel, if you first enter a single quote into a cell, and then type a numeric value, the value will actually be stored within MS Excel as a character value. The single quote can cause you more headaches than trying to untangle fishing line. It is an invisible character which exists in MS Excel. The one method of exposing this type is to use the procedure described above.

LEADING BLANKS

Another snag which you may find yourself agonizing over are values which may have leading blanks. Again, depending on how the data appears, these records may not be visible. This is especially true if the data is right justified within the column or cell. One technique that I like to use to find these nasty snags is to simply sort the column. Since MS Excel will want to sort numeric values and then character values, you quickly find that your sort is not accurate (at least not how you would have expected the data to sort).

MAKE MULTIPLE VARIABLES FROM ONE (EXAMPLE #7)

Maybe your data isn't exactly in the format that you want to import. A great example is how names are stored in a database. If for example, you want to import data from Excel and append a SAS data set with the new data. The challenge is the SAS data set has two separate fields for first name and last name, while the MS Excel spreadsheet has both values stored in the same cell. One possible technique for avoiding problems is to "break" the column into two columns such that it matches your data set by using the menu option of Text-to-Columns. This is a process where you will be able to separate the first name and the last name by either a fixed width or a delimited method.

Example #7

Follow these steps:

- Open the MS Excel file and worksheet
- Insert blank column(s) to the right of the value you are going to parse.
- Using the left mouse button, select the column of interest.
- Click DATA from the main menu.
- Click Text to columns
- Follow the instructions on the wizard

Note: This method may take some time to master, especially with difficult data, but can be quite helpful when you realize its power. This can be used for many other tasks which are outside this scope, but let your imagination run wild with the possibilities.

I DON'T WANT TO TOUCH SLIMY FISH (EXAMPLE #8)

So you don't want to touch the data. This is understandable, especially in a regulated environment you may not want to touch or modify the original data. If this is the case, then another option within MS Excel called filtering may be useful. How filtering works is quite simple, once the filter is applied, clicking on the drop down arrow will produce a sorted list of unique values that are stored in that column. Because the list of values is sorted, you will be able to spot some of the irregularities within the data. This method is also great when you have many rows of data to review.

Example #8

Follow these steps:

- Open the MS Excel file and worksheet
- Make sure that each of the columns has a label or title.
- Select the entire row which contains the titles.
- Click DATA from the main menu.
- Click FILTER
- Click Autofilter
- Click on the drop down box in each column that you want to view.

FISH ON!! EXPORT SAS DATA SETS TO MS EXCEL DATA

A famous line often heard on successful party boats is the ever exciting, “Fish On! Fish On!”. Now that you have the data stored as a SAS data set you can do whatever you want within the capabilities of SAS. The next logical step in working with MS Excel is to export the data into MS Excel. This process is a much easier and cleaner process than importing simply because you are now forcing MS Excel into having columns with consistent formats.

CATCH AND RELEASE: METHODS FOR EXPORTING

In all cases, the techniques presented for importing the data can be duplicated for exporting the data. The one major difference is that you are redirecting the data to a MS Excel Workbook (file) and Worksheet. Again, the technique used for exporting will be dependent on the licenses which you have available at your site.

Your only limitations may be the size of your data. Depending on the MS Excel version you are using, you will have different number of rows and columns to export the data. The need to export the data may be many, but the leading method in my experience is so that non-SAS users can use data which has been processed by SAS.

EXPORT Methods (all require SAS/CORE and SAS/BASE)	
Method	**Additional Software**
Data Step	None.
Export Wizard	SAS/ACCESS to PC File Formats.
ODBC	SAS ODBC driver (configured).
PROC EXPORT	SAS/ACCESS to PC File Formats.
PROC DBLOAD	SAS/ACCESS to PC File Formats.
DDE	None, however Excel must be open.
ODS	Version 8

For coding samples, reference Technical Specification TS598F: Export SAS Datasets into EXCEL (XLS) Format.

I DON’T LIKE TO CATCH FISH, BUT I’LL EAT ‘EM: USING ODS TO EXPORT FOR VERSION 8 USERS (EXAMPLE #9)

You know the type, can't stand wiggling or smell of the bait, don't want to touch the fish, or clean the daily catch, but they are the first to order the fillet of flounder. Sure you do, they are your non-SAS using customers. They don't care what you have to do to get them data or reports, they just want it.

One method which is extremely powerful for exporting to MS Excel is the use of ODS to create output which non-SAS users can use. The power of this method is that you can produce output from SAS procedures (such as Proc Tabulate, Proc Report), for import into MS Excel. You can open any HTML file, MTHML file, or well-formed XML file in Excel by using the Open command on the File menu.

Example #9

```
LIBNAME dlib ‘c:\sugi28\temp’;

Ods html file ‘c:\sugi28\temp\fishing.html’;
 Proc report data=slib.fishing nowd;
    Column fishname caught prizeamt lbswt;
    Define fishname  / ;
    Define caught      / ;
    Define prizeamt   / ;
    Define lbswt         /;
Run;
Ods HTML close;
```

The power of this process is that the output from the procedures can be analyzed by the non-SAS user. One of the great capabilities, especially for quality review, includes the ability to use auto filters to look at the data results. It is also possible to apply formats to the data which translate quite well in MS Excel, such as font weight, font color, and background colors to highlight specific values for the end user.

Another benefit for users of DDE, are that they can dynamically run VBA macro code from within a SAS program to perform an intended function on the data once it resides in MS Excel. Although this topic is beyond the scope of this paper, it is one that should be explored by SAS programmers who extensively use MS Excel. Reference SAS Institute Inc. TS325: The SAS System and DDE for additional information.

COMPARING FISH STORIES: SAS FUNCTIONS AND THEIR EQUIVALENT IN MS EXCEL

We have all heard the fish stories...comparing weight, length, and difficulty of the landing. The stories are compared back and forth between fishing buddies. Well here is the SAS / Excel version of that conversation of some of the more popular functions and procedures that you may encounter.

The purpose of the conversion chart is to provide users with a summarized version of capabilities within each product and its partner in the other product. To find out the proper syntax or procedure for each product, please reference the appropriate user manual or reference material. The syntax for the MS Excel functions is included in the Insert Function option.

(f) = Function (p) = Process or Procedure.

Description	SAS system	MS Excel
Absolute value	ABS (f)	ABS (f)
Remove blanks, unwanted characters	COMPRESS (f)	CLEAN (f)
Searches for a pattern of characters	INDEX (f)	SEARCH (f)
Finds first character pattern	INDEXC (f)	FIND (f)
Description	**SAS system**	**MS Excel**
Convert all values to lower case	LOWCASE (f)	LOWER (f)
Convert all values to upper case	UPPERCASE (f)	UPPER (f)

Convert text values to proper case	--NONE--	PROPER (f)
Return length of text	LENGTH (f)	LEN (f)
Extract a substring	SUBSTR (f)	LEFT (f), RIGHT (f), MID (f), TEXT TO COLUMN (p)
Change characters	TRANSLATE (f)	SUBSTITUTE (f), REPLACE (f)
Remove training blanks	TRIM (f)	TRIM (f)
Return Today's Date	DATE (f) TODAY (f)	TODAY (f)
Return current date and time of day	DATETIME (f)	NOW (f)
Return a date from a month, day, year	MDY(f)	DATE (f)
Return the largest value	MAX (f)	MAX (f)
Return the smallest value	MIN (f)	MIN (f)
Return the number of non-missing values	N (f)	COUNT (f) COUNTA (f)
Return the number of missing values	NMISS (f)	COUNTBLANK (f)
Compute the Mean	MEAN (f)	AVERAGE (f)
Return the largest value in part of a list	ORDINAL (f)	MAX (f)
Calculate the standard deviation	STD (f)	STDEV (f)
Calculate the sum of values	SUM (f)	SUM (f)
Return smallest integer >= to value	Ceil (f)	CEILING (f)
Return largest integer <= to value	FLOOR (f)	FLOOR (f)
Return Integer Value	INT (f)	INT (f)
Rounds unit to nearest unit	ROUND (f)	ROUND (f)
Count observations	FREQ (p)	pivot table (p)
Summarize data	TABULATE (p), UNIVARIATE (p), MEANS (p)	pivot table (p)
Merging Data	DATA STEP PROC SQL	Vlookup (f) Hlookup (f)
Numeric to Character conversion	PUT (f)	Insert single quote or Change format
Character to Numeric conversion	INPUT (f)	Remove single quote and change format
Join character values together	Use ||	Concatenate (f)
Sorting data	SORT (p)	Menu option
Subset data	Where statemet	Auto Filters
Transposition of variables	Transpose	--none--

CONCLUSION

The SAS system allows a programmer to perform many functions, excluding the ability to electronically provide data to a non-SAS user who is interested in reviewing the data. Therefore, each programmer should be aware of the options that are available to present the data in a different form.

Having the correct tools for the task at hand is important. When your goal is to catch fish, you will want to make sure that you have all the right equipment. In some cases, you will be able to substitute equipment and still bring home the fish.

The same can be said of the SAS system and its interactions with MS Excel. Making sure that you have the right SAS products, especially when importing data can make the difference between a simple task and a difficult task.

Understanding the behavior of the fish is also important to the successful fisherman. Again, the same is true for SAS programmers and MS Excel users. Knowing the behaviors of each during the import and export processes will make the task at hand easier. No not all fish eat worms, nor will all your work ever be in the SAS system, but you are now much better prepared to handle the challenge.

REFERENCES

SAS Institute Inc. SAS Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990.

SAS Institute Inc. SAS Procedures Guide: Reference, Version 6, Third Edition, Cary, NC: SAS Institute Inc., 1990.

New Features in Version 8 of the SAS System Course Notes, Book code 56963, Course Code NASH, Prepared Date 08Mar00, Cary, NC: SAS Institute Inc., 2000.

SAS Institute Inc. TS598B: Importing Excel Files to SAS Datasets. SAS Institute Inc., Cary NC

SAS Institute Inc. TS598F: Export SAS Datasets into EXCEL(XLS) Format. SAS Institute Inc., Cary NC

SAS Institute Inc. TS325: The SAS System and DDE. SAS Institute Inc., Cary NC

SAS Institute Inc. SAS OnlineDoc Version 8, February 2000, SAS/ACCESS to PC File Formats, Part 3: XLS Chapter, Cary, NC: SAS Institute Inc., 2000.

SAS Institute Inc., Proceedings of the Twenty-Seventy Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc., 2002. "Haven't We Met Somewhere Before?" and Other Useful Lines (of code) to Get to Know Their Data"; David Rucker, Covance Periapproval Services, SAS Institute Inc., Cary NC

ACKNOWLEDGMENTS

I would first like to thank my Grandfather who was instrumental in my fascination (not my lack of skill) with fishing. I would also like to thank all the individuals who have been a part of my career such that it made writing this paper possible. These individuals are: Lucas Makris, Carol Matthews, and the entire BioCor team, Alisa Monteiro, Ellen Brookstein, Jim Copp, Jim Grudzinski, the CSTAT team, Lloyd Yates, and Maryrose Sheppard.

CONTACT INFORMATION

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

David Rucker
BioCor LLC.

301 Oxford Valley Road
Suite 1725
Yardley, PA 19067
Work Phone: 215.321.9877 ext. 241
Fax: 215.321.9877
Email: drucker@biocor.com
Web: www.biocor.com