

Paper 082-29

A Practical Approach to Transferring Data from Microsoft Excel® to SAS® in Pharmaceutical Research

Hong Qi, Merck & Co., Inc., Blue Bell, PA

ABSTRACT

The need to transfer data from MS Excel into SAS is frequently encountered in pharmaceutical research. Multiple approaches could be taken in the data transfer, such as IMPORT WIZARD, SAS/ACCESS, ODBC, DDE and DBMS/COPY. However, these methods may not handle the data transfer correctly or efficiently, especially when data format inconsistencies are present or multiple Excel sheets are involved. This paper proposes a practical, user-friendly macro solution to this problem that provides many flexible features in the data transfer process. In analyzing the components of the macro's design, this paper also demonstrates how to incorporate the principles of accuracy, efficiency and convenience in the transfer of data from Excel to SAS. Finally, a real example is used to illustrate how the macro's features are applied in practice.

INTRODUCTION

In pharmaceutical research, Excel spreadsheets are commonly used for temporary display and storage as well as preliminary analysis of experimental data, which is important for early scientific and technical decision making. However, formal statistical analyses, tabulation and graphic display of the data are usually performed using more sophisticated software such as SAS, which requires transferring the data from Excel into that software format. Unfortunately, these Excel files are usually created without using systemically controlled methods, by different people or divisions, such as scientists from different laboratories, or medical program coordinators from various study sites. Therefore, inconsistency among data format is always an issue during the data transfer. Moreover, it is often desired to automate the data transfer process after some simple user initiation.

Over the years, many SUG papers have discussed a variety of approaches to performing data transfer, such as IMPORT WIZARD, SAS/ACCESS, ODBC, DDE and DBMS/COPY. Each of these methods has its unique features. However, some of them cannot handle the data inconsistencies, and thus, do not always transfer the data correctly. For example, if both text and numeric values appear in the same column of an Excel file, the corresponding SAS data set imported by IMPORT WIZARD, SAS/ACCESS, or ODBC would have missing values for that column (variable).

Among all the currently available approaches, Dynamic Data Exchange (DDE) has been widely accepted as one of the best means of transferring data with accuracy. This paper focuses on the use of DDE to design the macro %EXDDE to import data from Excel into SAS. In addition to the unique features of DDE in handling format inconsistencies, the macro also provides many flexible features in the data importing process, such as opening the input Excel file either automatically by the macro, or manually by the user; reading multiple Excel sub-sheets with a single macro call; customizing the data format for each variable to be imported; removing blank records; flagging the data origin of the sub-sheet; and more. The macro can be used to generate either temporary or permanent SAS data sets.

THE DESIGN OF MACRO %EXDDE

Macro %EXDDE was designed to be both powerful and user-friendly. It is particularly practical because of its ability to combine the advantages of DDE with features that cover other issues important to data transfer. To ensure that data could be transferred successfully with this macro, the following factors were considered during its development: the features of Excel files in pharmaceutical research, the components of the data transfer process and the contents in the output SAS data set. The design process was also guided by the principles of accuracy, efficiency and convenience. The macro's ability to incorporate each of these principles is described below.

ACCURACY

DDE is well known for its accuracy in data transfer. However, the execution of data transfer can be stopped when this approach is used inappropriately. The accuracy of DDE relies on factors including correct data path, file name, proper reading order, sufficient buffering space, and appropriate formats for the input variables. Therefore, to ensure that data is transported correctly, macro %EXDDE defines corresponding logic checks and macro variables. The following are examples of some of these safeguards.

1. Logic checks: Logic checks are the first guard against incorrect transfer. They are established at the beginning of the macro to verify the existence of the Excel file, the correct specifications of required parameters and the starting and ending rows and columns of the imported data. The data process does not start if the Excel file does

not exist, the required parameters are not specified or the orders are not correct for the starting and ending rows and/or columns.

2. Data importing order: During the data transfer process, the Excel file is read into SAS from top to bottom in a pre-defined order, that is, from the starting row and column to the ending row and column. If the starting and/or ending column numbers are specified before the starting and/or ending row numbers, for example,

```
Filename one DDE "Excel|U:\Excel Data Issue\[test.xls]Sheet1!C1R1:C10R10";
```

the transfer process is stopped with the following message in the log file/window:

```
ERROR: DDE session not ready.
FATAL: Unrecoverable I/O error detected in the execution of the data step
program.
Aborted during the EXECUTION phase.
```

The message is very difficult to interpret without the understanding of the required data reading order, resulting in frustration in the data handling.

By defining parameters &StartR, &StartC, &EndR and &EndC, it is possible to set up the sequence for reading the file so that the data are always processed in the appropriate order.

3. Horizontally stretched data: It is a common practice within the pharmaceutical industry for an Excel file to contain a large amount of information on one sheet with multiple columns. When importing such horizontally stretched files, the default record-length (Lrecl) of 256 may not provide enough buffering space, thus leading to incorrect output SAS data without any warning or error messages provided in the log file/window. In this case, the inconsistencies between the input Excel file and the output SAS data, which may vary from one variable for one record to multiple variables for multiple records, sometimes, can hardly be identified without a thorough data validation. This issue can be solved by increasing the Lrecl. The value of Lrecl can range from 1 to 1,048,576 (1 megabyte). According to the experience of SAS experts, an Lrecl of 8192 covers 99% of the cases that most SAS programmers encounter. However, the larger the Lrecl, the slower the data is read because of the efforts involved in allocating a big buffer. Therefore, macro %EXDDE sets up the optional parameter &Lrecl and gives the user the choice to specify the Lrecl when it is necessary to make sure that all the fields can be handled correctly. If not specified, the default value is then used.

EFFICIENCY

Efficiency was a primary concern in the design of macro %EXDDE since one of its purposes was to automate the data transfer process. The following examples characterize how the macro efficiently deals with errors and multiple sheets, as well as other information concerning data transfer.

1. Logic checks: Instead of running the whole program, the logic checks make it possible to stop the program before the data transfer begins when a logic error exists. Error or warning messages can be viewed in the SAS LOG file/Window. Therefore, a user can locate the error early and make a quick fix. For example, for the macro call below importing data from a larger number to a smaller number of columns,

```
%EXDDE(Rawdir=&rawdir,
      File=devtest,
      Sheet=Sheet1,
      StartR=2,
      StartC=4,
      EndR=5,
      EndC=1,
      Outdata=test);
```

the following error message will be generated at the end of the SAS LOG Window/File:

```
@@@@@@@ MESSAGE FROM MACRO EXDDE @@@@@@@@@@
@ LOGIC_ERR = 3. @
@ PARAMETERS ENDR/ENDC MUST BE LARGER THAN @
@ OR EQUAL TO STARTR/STARTC. @
@ PLEASE TRY AGAIN. @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

The message clearly indicates the point to debug the program.

2. Multiple sheets: Macro %EXDDE also includes a small utility macro which parses a list of tokens for the parameter &Sheet. This macro allows for the reading of multiple sheets nested in the same Excel file with a single macro call by specifying the parameter &Sheet with multiple sheet names. In the case of transferring data from multiple sheets of a file, this feature is highly efficient not only because of the reduced number of macro calls, but also because of the reduction in cpu time. The cpu time decreases significantly when two sheets are transferred with one macro call, instead of two.

3. Extraneous Information:

Some information in the Excel file, such as certain columns (variables) or missing records, may not necessarily be included in the output SAS data set. In addition, it is sometimes important to identify the data origin by sheet since each sheet may represent a specific investigator site or protocol. To make the data transfer an efficient process, macro %EXDDE has the option of removing missing records and/or extra columns, and also of flagging the records by sheet before creating the output SAS data set. The optional parameter &Dum is designed to drop columns in the middle of the file since all the columns between the first and last ones have to be imported. The parameter &Flag is used for indicating the sheet origin of each record.

Finally, the generated SAS data set, imported from all the specified sheets of the Excel file by one macro call, only contains the necessary fields with non-missing records and the data origin identifier.

CONVENIENCE

Besides being powerful, the macro is also designed to be user-friendly. For example, the macro integrates functions for opening the input Excel file either automatically by the program, or manually by the user. In addition, with the minimum requirement of defining only three required parameters for data transfer, the macro provides the flexibility of specifying the variable name and format in a one-to-one correspondence. Basically, the macro provides the user with some control during the data transfer process, such as keeping or deleting all the intermediate SAS data sets, and leaving the Excel file open or closed. The list below offers some details about these useful features.

1. The control of opening the Excel file: The disadvantage of the DDE approach is that the Excel file must be open during the data process. However, macro %EXDDE integrates the function of opening the Excel file with an X command. Three optional parameters are responsible for this feature: &Excel, &Sleep and &Debug. They provide the options of opening the Excel file by the program or by the user, establishing the time in seconds that the SAS system should wait for execution if opened by the program, and leaving the Excel file open for validation purposes.

2. Minimum requirements: This feature allows the user to have a quick overview of the imported SAS data set before spending more effort in defining all the parameters. The following are the only parameters that must be specified in order to run the macro:

- (1) input Excel file directory (Rawdir)
- (2) input Excel file name (File)
- (3) output SAS data set name (Outdata)

When these three parameters are specified, the macro automatically imports the first 101 rows (records), and 30 columns (variables) on the Excel file into SAS, with the default variable names col1-col30 and format \$30. for all the variables.

3. Variable name and format: There is flexibility in specifying variable names and formats in a one-to-one correspondence which makes it convenient to define the format for each variable. Moreover, the format can be character, numeric, or decimal.

THE MACRO CALL SYNTAX AND PARAMETERS

The macro call syntax is:

```
%EXDDE(Excel =,
        Xlsexex =,
        Rawdir =,
        File =,
        Sheet =,
        Lrecl =,
        StartR =,
        StartC =,
        EndR =,
        EndC =,
        Var =,
        Fmt =,
        Dum =,
        Flag =,
        Outdata=,
        Sleep =,
        Debug =);
```

Macro %EXDDE contains seventeen parameters, three of them are required, and the rest are optional and provide further control for the input Excel file and output SAS data sets. The parameters can be categorized into three groups:

1. Excel file related parameters

Excel	When = ON, the Excel file is already open; otherwise, the program will open the Excel file.
Xlsexex	Location of Microsoft Excel on the C: drive
Rawdir	Directory of the Excel file (Required parameter)
File	Excel file name (Required parameter)
Sheet	Sub-sheet name to be read
Lrecl	Length of the Excel file, default is 256
StartR	Row number of the 1st cell to read
StartC	Column number of the 1st cell to read
EndR	Row number of the last cell to read
EndC	Column number of the last cell to read

2. Parameters related to the output SAS data set

Var	Variable list
Fmt	Format of the variables
Dum	Dummy variables needed to be dropped
Flag	Flag for the sub-sheet origin of the data
Outdata	Output SAS data set name (Required parameter)

3. Process-related parameters

Sleep	Number of seconds that a SAS data step is suspended from execution while opening up the Excel file
Debug	Whether to keep intermediate data sets, default=NO

ILLUSTRATION AND DISCUSSION OF A SAMPLE CALL

This section demonstrates how macro %EXDDE is used in an example of a data transfer. It demonstrates the macro's features and offers some suggestions for proper usage.

1. Input Excel Data

There are two sheets in one Excel file (devtest.xls) to be imported into SAS: Sheet1 and Sheet2 (Figures 1 and 2). At least the first four columns of each sheet have format inconsistencies – combined character, numeric and/or date formats. There are also missing records on both sheets – Row 7 and Columns C, E and F are not needed in the output SAS data set.

	A	B	C	D	E	F	G	H	I	J
1	ID	Site	ddd	Date	dd2	dd3	comments	date	num	num2
2	a	001	999	abc	111	111	this is a test d.	2-Jan-03	11	11.2
3	b	001	999	09-Jun-202	11	111		3-Feb-03	12	22.34
4	1	1234	999	9-Jun-02	11-Jan-00	20-Apr-00		4-Mar-03	13	33.2
5	c	2222	999	12	111			5-Apr-03	14	34.5
6	002	002	99	9-Jun-02	11-Jan-00				15	23.1
7										
8	3	2	99	111	111	20		8-Jun-03	55	21.2

Figure 1. Input Excel Data – devtest.xls (Sheet1)

	A	B	C	D	E	F	G	H	I	J
1	ID	Site	ddd	Date	dd2	dd3	comments	date	num	num2
2	g	001	999	abc	111	111	Sheet2 test data	2-Jan-03	11	11.2
3	f	001	999	09-Jun-202	11	111		3-Feb-03	12	22.34
4	2	1234	999	9-Jun-02	11-Jan-00	20-Apr-00		4-Mar-03	13	33.2
5	m	2222	999	12	111			5-Apr-03	14	34.5
6	005	002	99	9-Jun-02	11-Jan-00				15	23.1
7										
8	8	2	99	111	111	20		8-Jun-03	55	21.2

Figure 2. Input Excel Data – devtest.xls (Sheet2)

	id	site	date	comments	dd	num	num2	rflag
1	a	001	abc	this is a test data	15707	11	11.2	001
2	b	001	09-Jun-202		15739	12	22.34	001
3	1	1234	9-Jun-02		15768	13	33.2	001
4	c	2222	12		15800	14	34.5	001
5	002	002	9-Jun-02		.	15	23.1	001
6	3	2	111		15864	55	21.2	001
7	g	001	abc	Sheet2 test data	15707	11	11.2	002
8	f	001	09-Jun-202		15739	12	22.34	002
9	2	1234	9-Jun-02		15768	13	33.2	002
10	m	2222	12		15800	14	34.5	002
11	005	002	9-Jun-02		.	15	23.1	002
12	8	2	111		15864	55	21.2	002

Figure 3. Output SAS Data – t2.sas7bdat

2. Macro Call Syntax:

```
%EXDDE(Rawdir=U:\Excel Data Issue,
        File=devtest,
        Sheet=Sheet1|Sheet2,
        StartR=2,
        StartC=1,
        EndR=8,
        EndC=10,
        Var=id site dum1 date dum2 dum3 comments dd num num2,
        Fmt=$4.$4. $1. $15. $1. $1. $80. date9. 3. 5.2,
        Dum=dum1-dum3,
        Flag=001|002,
        Outdata=t2);
```

3. Output SAS data set:

The macro call generates a SAS data set as presented in Figure 3.

Some features of such a data set are:

- i) No format inconsistency issue: All the data values are imported as being populated on the Excel file along with the format specified by the macro, regardless of the format inconsistencies in the Excel data.
- ii) Combined SAS data set: Two sheets of the Excel file are combined into one SAS data set with the variable FLAG identifying the sheet origin of each record.
- iii) No dummy variables: Only variables necessary are included.
- iv) No blank records: Records with missing values for all variables are removed from the data set.

Macro %EXDDE is no doubt a powerful tool for data transfer from Excel to SAS. However, it is still necessary to take certain cautions for correct usage of this macro. The following is a list of some helpful suggestions:

1. It is advisable to use character format for columns (variables) that may contain inconsistent formats so that the data value can be imported as is from the Excel file.
2. Hidden columns before the ending column in the Excel file should be considered in the transfer.
3. A comma, or space before or after sheet name is not permitted.

CONCLUSION

Macro %EXDDE offers a practical approach to transferring data from Excel to SAS. In addition to assuring the accuracy and efficiency of the data transfer, it paves the way for formal review and analyses of experimental data with SAS. As a powerful tool, it significantly streamlines the data transfer process. Most importantly, it incorporates the basic principles for designing a dynamic macro – accuracy, efficiency and convenience.

REFERENCES

1. Mumma, M. T. (1999) "The Redmond To Cary Express – A Comparison of Methods to Automate Data Transfer Between SAS® and Microsoft Excel®" in *Proceedings of the 12th Annual NorthEast SAS® Users Group Conference in 1999*. P.654-62.
2. Smith, R., Carpenter, A. (1999) "The Use of External Software to Import Data into the SAS® System" in *Proceedings of the 24th Annual SAS® Users Group International in 1999*. P.222-25.
3. Sun, E. T. (2001) "When PROC ACCESS Says NO to Excel 97/2000, DDE Says YES Still" in *Proceedings of the 26th Annual SAS® Users Group International in 2001*. P.119-26.
4. Vyverman, K. (2003) "Using Dynamic Data Exchange to Export Your SAS® Data to MS Excel – Against All ODS, Part I -" in *Proceedings of the 28th Annual SAS® Users Group International in 2003*. P.11-26.

ACKNOWLEDGMENT

The author thanks Allan Glaser for his continuous support and valuable comments, Haoli Chai for his inspired work, Peter Ruzsa for his help with some technical issues, Frederic Coppin and Stat programmers in Merck Belgium Biostatistics group for utilizing the macro in their projects and their active feedback, Michael Senderak for his time and efforts in assessing the macro, and Donna Usavage for reviewing the paper and her helpful comments.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author by mail or e-mail at the following address:

Hong Qi
Merck & Co. Inc.
785 Jolly Road, UNA-102
Blue Bell, PA 19422

Work Phone: 484-344-7643
Fax: 484-344-7105
Email: hong.qi@merck.com

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.