

## Export SAS Data to Excel in SAS<sup>®</sup> Drug Development

Chun Feng, Celerion, Lincoln, NE

Xiaopeng Li, Celerion, Lincoln, NE

### ABSTRACT

Exporting SAS data to Excel<sup>®</sup> is a common task for programmers. SAS<sup>®</sup> 9 provides multiple powerful techniques to output SAS data to Excel, such as proc export, libname engine, output delivery system (ODS), dynamic data exchange (DDE), and ExcelXP tagset. This paper discusses capabilities and advantages of each technique in SAS Drug Development (SDD), a relatively new regulatory-compliant SAS system for clinical research. With new interface, Java powered editor, and web-based system, SDD performs differently than PC SAS<sup>1</sup> in some aspects to create an Excel file. New convenient ways are available in SDD, including outputting from data explorer and defining output data table as an Excel file in data steps. This paper will greatly help SDD users to efficiently export SAS data to Excel.

### INTRODUCTION

Exporting a SAS data set into Excel is a common task for SAS programmers. In SAS 9 (versions 9.1 and 9.2), multiple techniques are available, including proc export, libname engine, ODS, DDE, and ExcelXP tagset. Advantages and disadvantages of each technique, as well as reasons for choosing among them while exporting data into Excel using PC SAS 9, have been previously assessed (Miralles, 2011). In this paper, we will discuss the availability of each technique in SDD as well as introduce new and convenient ways of outputting SAS data sets into Excel using SDD.

In recent years, a number of pharmaceutical and biotechnology companies, and clinical research organizations have migrated to SDD for managing, analyzing, and reporting clinical trial data. SDD offers desirable functionality for drug development research, including centralization, security, automated integrity, traceability, transparency, and a graphic user interface design, making the SDD programming environment different from PC SAS 9 for Windows Operating System. As a result, programmers often need to change their programming habits and codes in order to program efficiently in SDD.

While the SDD interface is user-friendly, it does take time for experienced PC SAS users to adjust to the various differences between the platforms as a number of unique features differentiate SDD from PC SAS. First, SDD is a web-based SAS system with global accessibility, residing on a UNIX<sup>®</sup> operating system remotely maintained by SAS Institute. SDD users are provided with a SDD desktop connection which can map a local drive to the remote server. Thus, codes are edited locally, are executed on a remote server over an internet connection, and the results are returned to the user following execution. Second, the code developing area, SAS log, SAS output, work library, input files, output files, and file explorer are integrated into one window known as the process editor. Third, the SDD process editor is a Java application. SDD users install a Java Runtime Environment (JRE) on their local PC in order to develop, execute, and view results in the Java-powered process editor. Importantly, the correct JRE version should be installed to initialize the application. Finally, macro variables are used in place of the path to the location of a SAS library or an input file.

Because of these unique features of SDD, some techniques applied regularly in PC SAS must be altered to be applied in SDD, or are unavailable in SDD. Meanwhile, some new functions become available in SDD that are not found in PC SAS. As will be described below, these new functions are more convenient than some traditional approaches.

### AVAILABILITY AND ADVANTAGES OF EACH TECHNIQUE

The availability, capability, and advantages of proc export, libname engine, ODS, DDE, and ExcelXP tagset are discussed in the sections below. The examples were developed using SDD version 3.4 (SAS 9.1.3).

### PROC EXPORT

Proc Export is a widely used method to output SAS data sets into Excel using PC SAS 9. In the following SAS codes, a SAS data set called "tmpdata" in the WORK library is exported into an Excel file called "tmpdata.xls" in the location "C:\sasfiles".

---

<sup>1</sup> PC SAS refers to PC SAS for Windows Operating System in this article.

```
proc export data=work.tmpdata dbms=excel outfile="C:\sasfiles\tmpdata.xls" replace;
run;
```

Submitting the above codes results in the error message “DBMS type EXCEL not valid for export.”, demonstrating that proc export cannot create an excel file in SDD.

## LIBNAME ENGINE

In PC SAS 9, the libname engine method assigns an Excel workbook as a reference library and an individual Excel sheet will be written to the library. By executing the sample codes below in PC SAS 9, the library reference name of “mylib” is created and a spreadsheet “data1” is created in the workbook “tmpdata.xls” in “C:\sasfiles” directory.

```
libname mylib "C:\sasfiles\tmpdata.xls";
data mylib.data1;
set work.tmpdata;
libname mylib clear;
```

In SDD, the libname engine method does not assign an Excel library or open an Excel workbook for Excel sheets to be written in. The reason is that the codes are executed on the UNIX operating system in SDD; therefore a Microsoft Excel workbook cannot be opened as in Windows operating system. Libname engine method is unavailable in SDD.

## OUTPUT DELIVERY SYSTEM (ODS)

There are two approaches to create an Excel file using ODS, “ODS CSV” and “ODS HTML”. “ODS CSV” outputs a “Comma-Separated Value File” (CSV) which can be easily read in Excel. Compared to “.xls” files, “.csv” files without formats have smaller file sizes and are more easily read by other software. “ODS HTML” outputs an Excel file while defining HTML body as “.xls” format. The following two sets of PC SAS codes create a CSV file “tmpdata.csv” and an Excel file “tmpdata.xls” separately in the location “C:\sasfiles”.

```
ods csv file="C:\sasfiles\tmpdata.csv";
proc print data=work.tmpdata;
run;
ods csv close;
run;

ods html body="C:\sasfiles\tmpdata.xls";
proc print data=work.tmpdata;
run;
ods html close;
run;
```

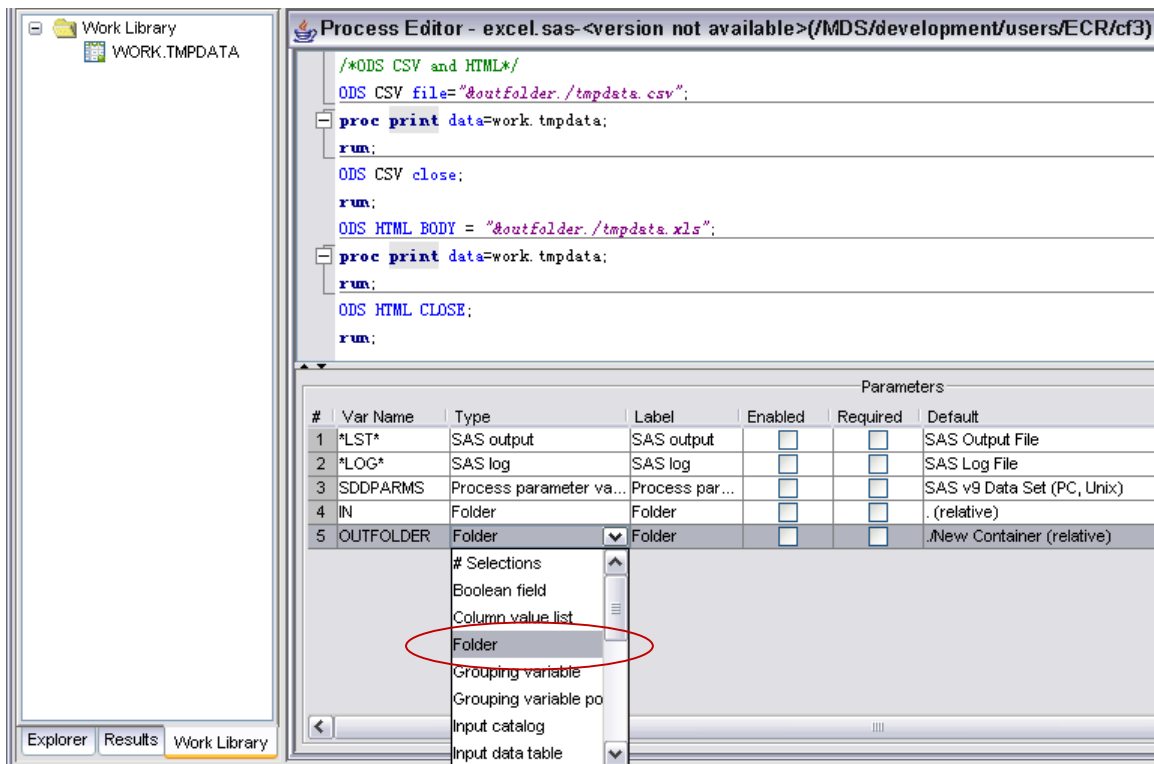
ODS CSV and ODS HTML are available in SDD and capable of generating “.csv” and “.xls” files. However, since SDD is on a remote server and a local disk “C:\” does not exist, an appropriate output location other than “C:\sasfiles” need to be identified before executing the codes. To identify an output location, the following steps should be followed. First, as shown in the following modified codes, the macro variable named “&outfolder.” represents the path to the output location. Thus, the traditional PC SAS code “C:\sasfiles\tmpdata.xls” needs to be replaced by “&outfolder./tmpdata.xls”.

The modified codes for “ODS CSV” and “ODS HTML” are as follows:

```
ods csv file="&outfolder./tmpdata.csv";
proc print data=work.tmpdata;
run;
ods csv close;
run;

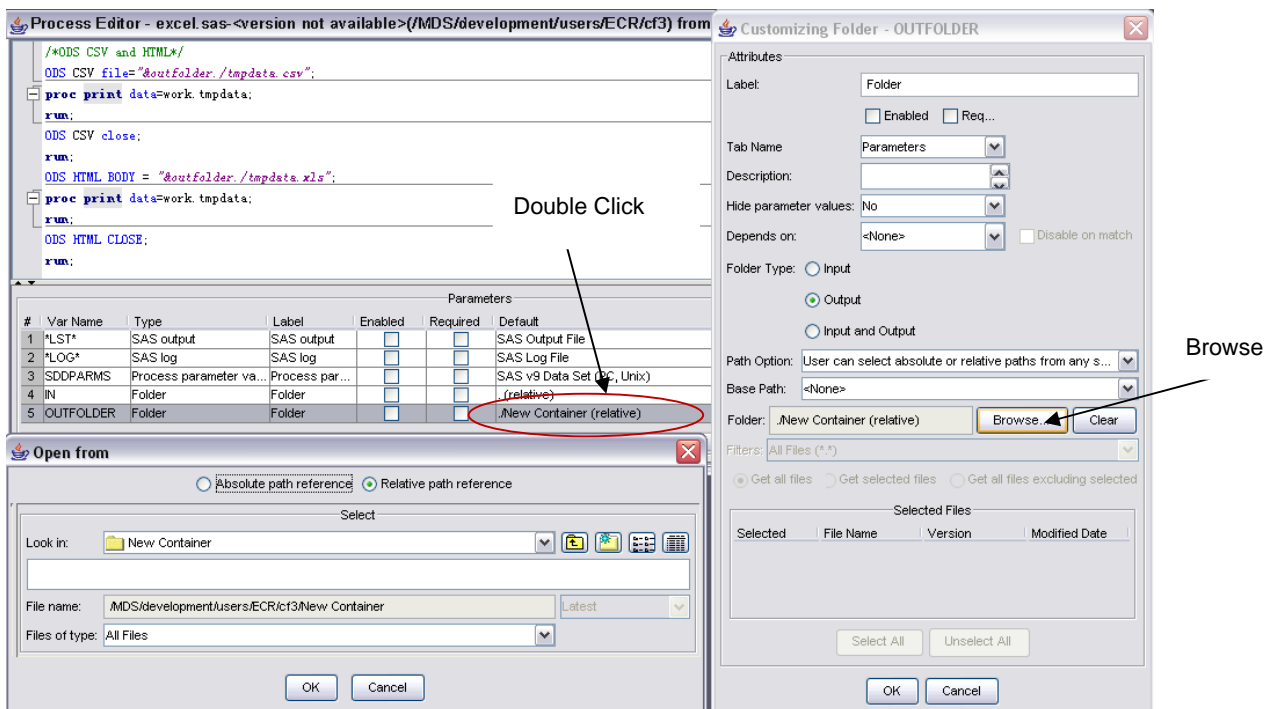
ods html body="&outfolder./tmpdata.xls";
proc print data=work.tmpdata;
run;
ods html close;
run;
```

Upon saving the new code, a new parameter “outfolder” is created within the parameter window (Display 1, parameter window line 4). Next, the macro variable “Type” is defined as a folder.



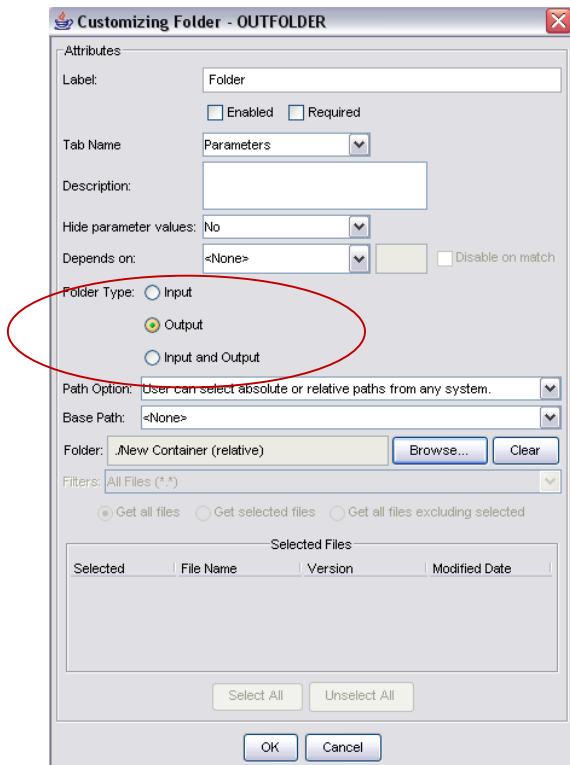
**Display 1. Select Type=Folder for “Outfolder” Macro Variable in the Parameter Window**

To customize the folder, double click the “Default” cell for the parameter “outfolder”. A customizing folder window appears by this action (Display 2). Within the “Customizing Folder” window, the user selects the output location by clicking the browse button and then selecting the appropriate folder in the “Open From” window that appears.



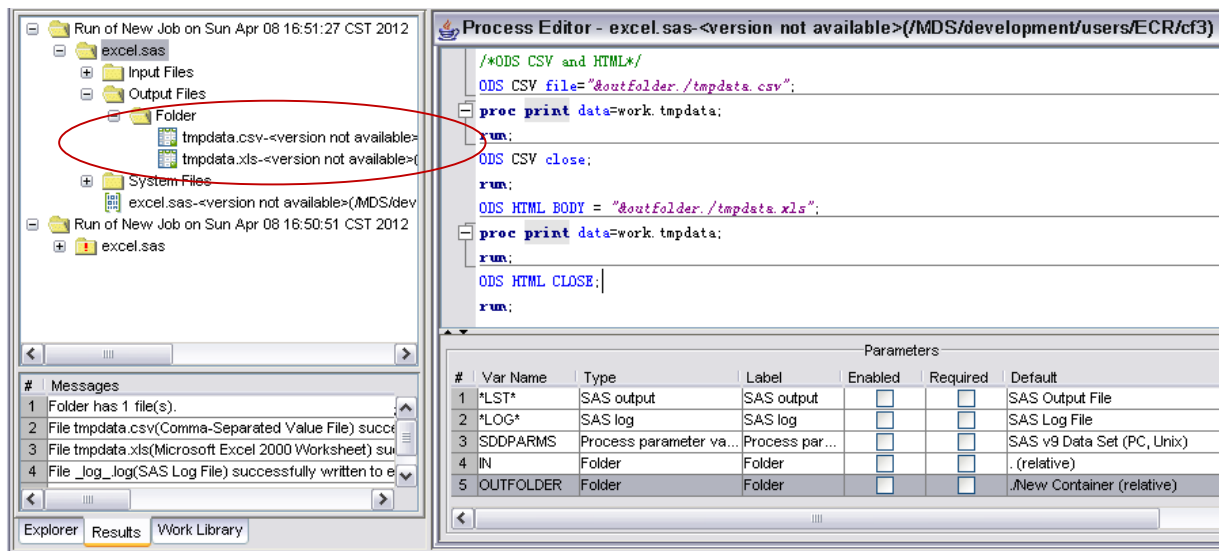
**Display 2. Double Click “Default” Cell and Use “Browse...” to Customize the Path**

Display 3 shows how to select the folder type. Within the “Customizing Folder” window, the user selects the “Output” folder type to define the output location for the Excel file that will be created.



**Display 3. Select Folder Type as Output**

With simple code modifications, the processes successfully execute in SDD, creating tmpdata.csv and tmpdata.xls files (Display 4).



**Display 4. Tmpdata.csv and Tmpdata.xls are Created by ODS CSV and ODS HTML**

### DYNAMIC DATA EXCHANGE (DDE)

In a standard DATA step, DDE allows SAS data content to be placed precisely into Excel. The following sample codes intend to create an Excel sheet named “myexcel” using PC SAS. Three variables, clientid, ds\_stdat, and

ds\_ans, originally in the SAS dataset “tmpdata” in the WORK library are written to the location row 2 column 1 to row 7 column 3 (R2C1:R7C3) in the Excel spreadsheet and the FILENAME statement defines the output location within the Excel file. DDE in the FILENAME statement specifies that DDE is used to exchange data.

```
filename tmpdata dde 'EXCEL|myexcel!R2C1:R7C3';
data _null_;
set work.tmpdata;
file tmpdata;
put clientid ds_stdat ds_ans;
```

However, after executing the codes in SDD, an error message in SAS log indicates that DDE is an invalid device type. Therefore, the DDE procedure is unavailable in SDD.

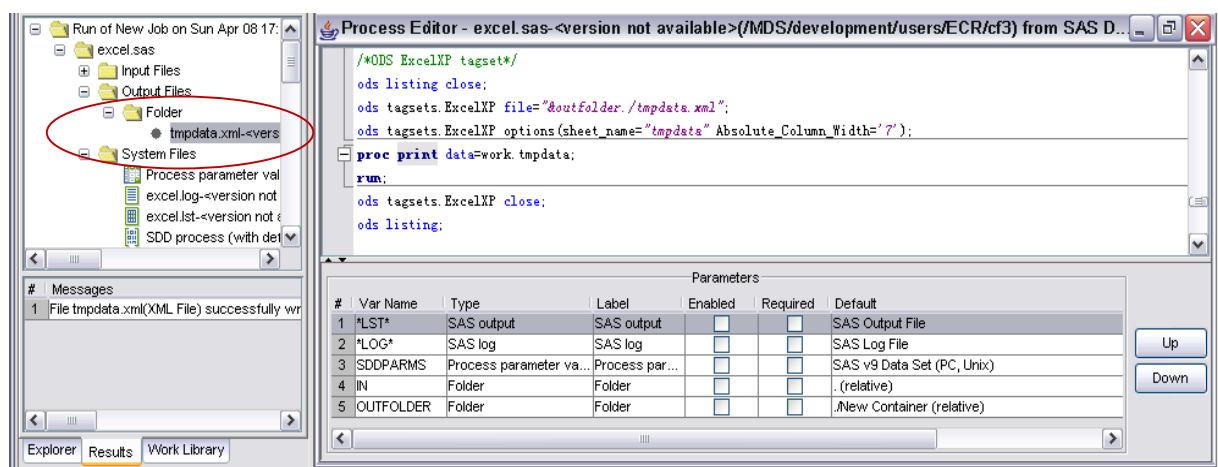
## EXCELXP TAGSET

ExcelXP Tagset is an advanced way of creating an Excel file. ExcelXP Tagset not only controls the output file location, but also controls the appearance of the Excel file, with the user specifying formats such as font size, font color, background color, bold, and italics in the options. In addition, ExcelXP tagset is also preferred to create multiple sheets in a single workbook. For more information on these two advantages, see Mahipal Vanam, Kiran Karidi and Sridhar Dodlapati (2010). In PC SAS, the following codes will create a “tmpdata.xml” in “C:\sasfiles”. The first excel sheet name is “tmpdata” and the column width of each column is 7.

```
ods listing close;
ods tagsets.excelxp file="C:\sasfiles\tmpdata.xml";
ods tagsets.excelxp options(sheet_name="tmpdata" Absolute_Column_Width='7');
proc print data=work.tmpdata;
run;
ods tagsets.excelxp close;
ods listing;
```

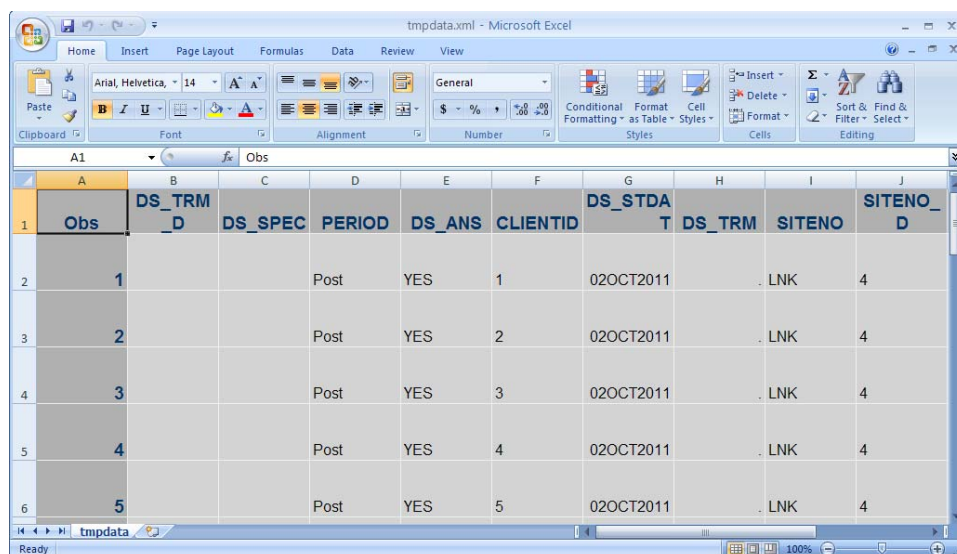
ODS Tagsets. ExcelXP can be used similarly in either PC SAS or SDD. Replacing “C:\sasfiles” by “&outfolder.” and following the steps in above “ODS” section to identify an output location, the following sample codes in SDD create an XML file “tmpdata.xml” that can be opened in Excel 2003 or 2007. Display 5 shows the outcome of the SDD codes.

```
ods listing close;
ods tagsets.excelxp file="&outfolder./tmpdata.xml";
ods tagsets.excelxp options(sheet_name="tmpdata" Absolute_Column_Width='7');
proc print data=work.tmpdata;
run;
ods tagsets.excelxp close;
ods listing;
```



Display 5. Tmpdata.xml is generated by ODS Tagsets.ExcelXP in SDD

Display 6 shows the Excel sheet “tmpdata” inside the Excel workbook “tmpdata.xml” in Excel 2007. The absolute column width of each column is 7.



**Display 6. Tmpdata.xml in Excel 2007**

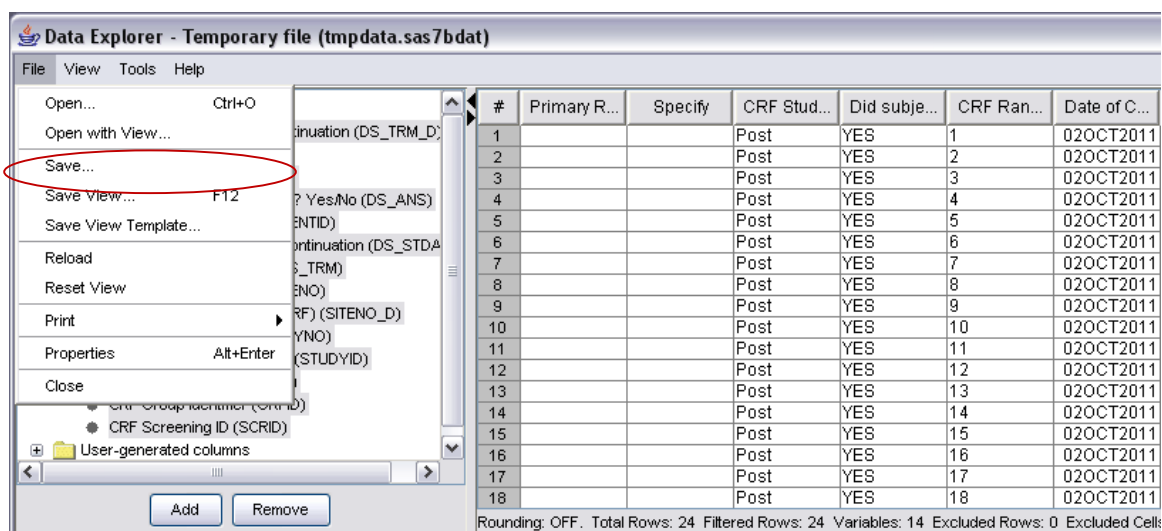
In summary, among often used methods of transforming SAS datasets to Excel files in PC SAS 9, ODS CSV, ODS HTML, and ODS tagsets. ExcelXP are equally easily applied in PC SAS® and SDD. However, proc export, libname engine, and DDE cannot be used in SDD even with modified code.

## NEW APPROACHES OF OUTPUTTING TO EXCEL IN SDD

SDD provides new, convenient methods to export SAS data sets into Excel. The following section introduces two new approaches of outputting to Excel using SDD.

### APPROACH 1: SAVE AN EXCEL FILE FROM DATA EXPLORER

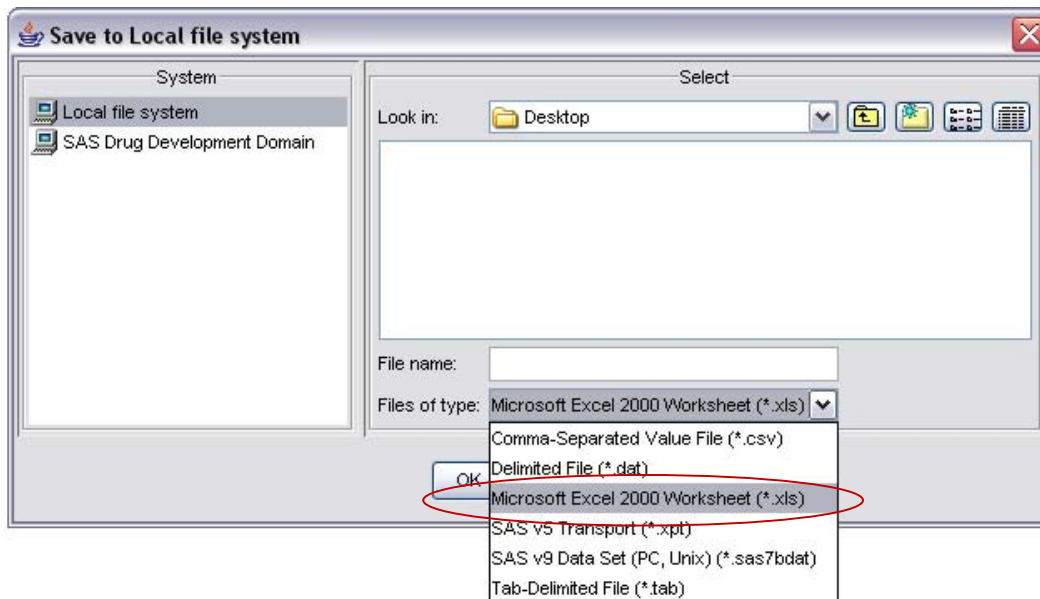
Data sets can be reviewed in data explorer in SDD. In the data explorer, there is a “Save...” option under the “File” dropdown menu (Display 7).



**Display 7. File->Save in Data Explore**

Following “File->Save...”, the user can select the file type as “Microsoft Excel Workbook 2000” in the dropdown menu of the “Save to Local File System” window (Display 8), with the file location and file name also specified in the

window. A local drive or a server location can be used to save the file. Display 8 shows the file being saved on the local Desktop.



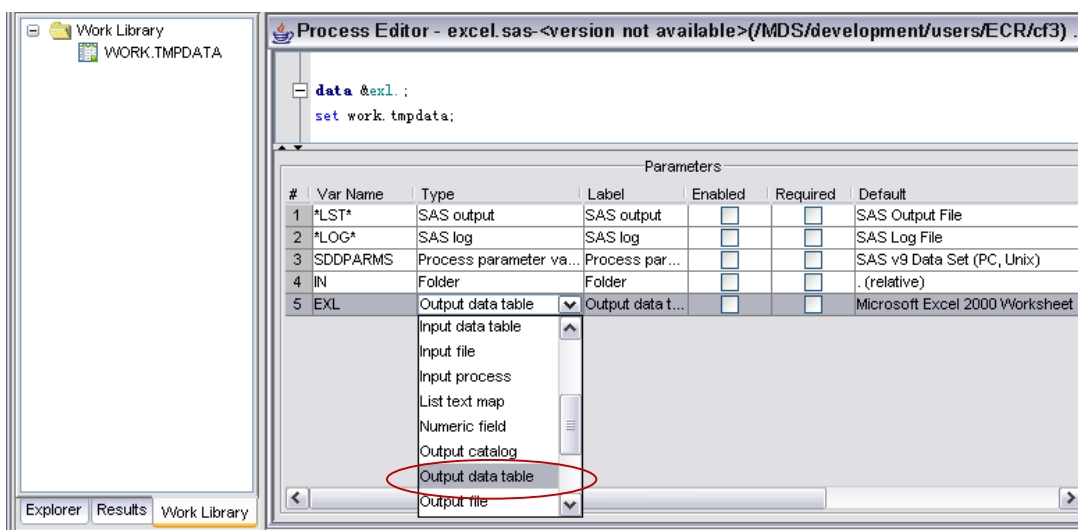
**Display 8. Microsoft Excel 2000 Worksheet (\*.xls) is Selected as File Type**

This approach provides a fast, simple way to save a data set as an Excel file in SDD; however, it is not without limitations. While any permanent or temporary data set can be saved as an Excel file from data explorer, Excel files generated by this approach cannot be updated automatically if the source SAS data sets are updated. Therefore, only by repeating the process manually will the Excel file be updated. In addition, creation of a multi-sheet Excel file is not possible.

## APPROACH 2: OUTPUT DATA TABLE AS EXCEL IN DATA

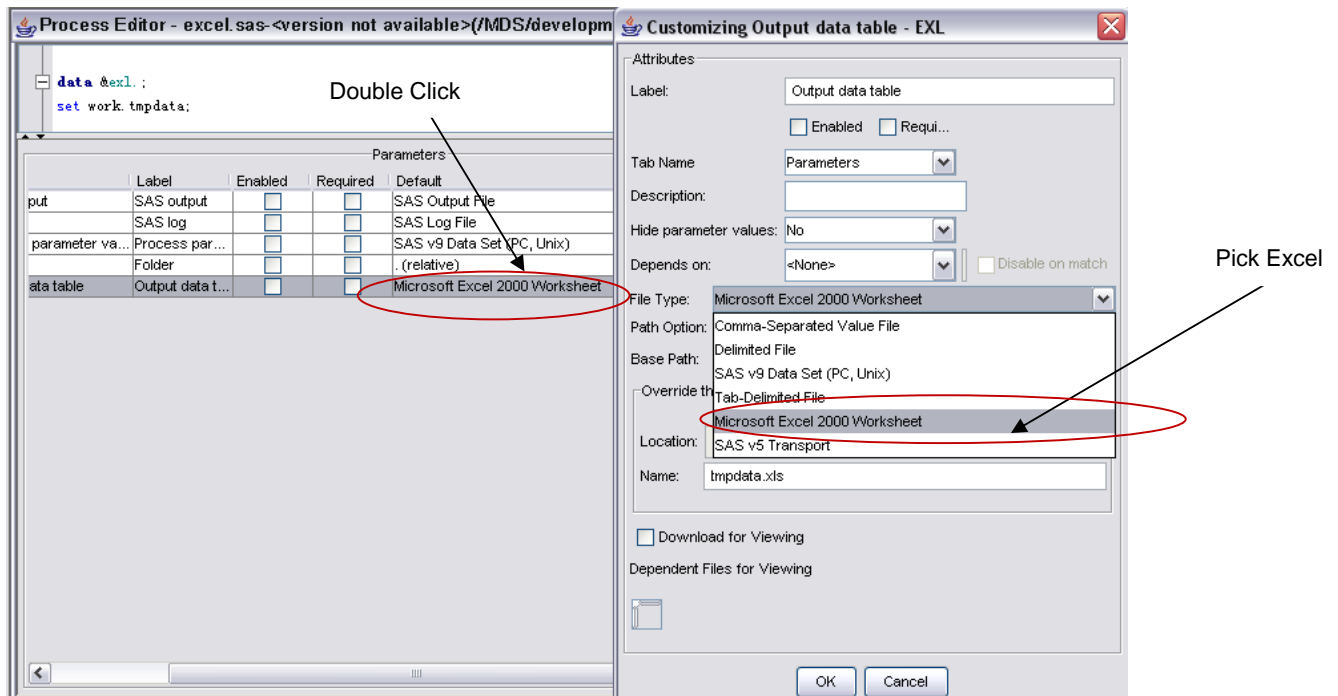
A DATA step is another convenient way of creating an Excel file in the process editor. To use this method, the macro variable “&exl.” is generated in the process editor and the parameter is defined as an “output data table” in the parameter window (Display 9).

```
data &exl;
set work.tmpdata;
```



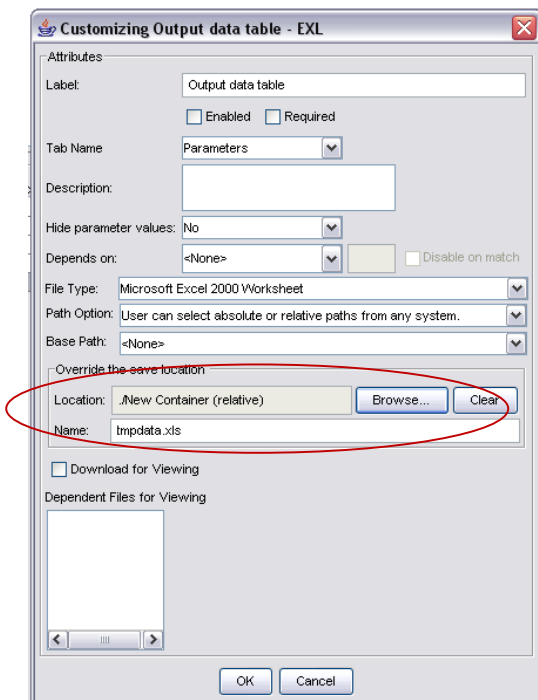
**Display 9. Pick Type=Output Data Table for Macro Viable “EXL” in Parameter Window**

As shown in Display 10, by double-clicking the “Default” cell, a “Customizing Output data table – EXL” window appears. In the window, the user should select “Microsoft Excel 2000 Worksheet” for the file type.



**Display 10. File Type = Microsoft Excel 2000 Worksheet in Customizing Output Data Table Window**

By browsing, intended output location on the server or on a local drive can be pointed (Display 11). In this example, a folder “New Container” at the same location of the program on the server is selected as an output folder. File name “tmpdata.xls” is typed as the Excel file’s name.



**Display 11. Browse to File Location and Name the File**



One advantage of this approach is that the Excel file will be updated automatically when the program is executed. However, the approach is relatively time-consuming compared to previous method as it takes more time to define the file type, name, and location of the file. Besides, the approach is also unable to generate multi-sheet Excel files.

## CONCLUSION

Creating an Excel file from a SAS data set is a common task for programmers. Excel /CSV formats offer a convenient method to exchange data with the advantages of being portable, easy to review, and can be easily imported into other analytical software by internal and external clients. This paper shows SAS users who are new to the SDD programming environment methods to efficiently export SAS datasets to Excel, as well as identifies barriers that might be encountered by programmers attempting to use traditional PC SAS codes within SDD.

While SDD offers organizations and programmers a number of advantages, programmers need a thorough understanding of SDD's functionalities, operating system, devices, stability, and compatibility to effectively and efficiently perform SAS programming tasks accurately. As demonstrated, some traditional PC SAS codes can simply be altered in SDD in order to perform certain functions offering programmers some familiarity in the codes and procedures used, whereas other techniques are not possible within SDD. However, SDD does allow the use of new and convenient methods to create Excel files from SAS data sets.

While other convenient ways for exporting to Excel in SDD may be explored in the future, this paper serves to initiate the discussion of programming environment alterations from PC SAS to SDD. Knowing the differences between two platforms of SAS enhances accuracy and efficiency while applying a familiar procedure in one system to the other.

## REFERENCES

SAS 9.1.3 Online Documentation

SAS Drug Development 3.4 User's Guides

Miralles, R., "Creating an Excel report: A comparison of the different techniques." *Proceedings of the SAS Global 2011 Forum*. Available at: <http://support.sas.com/resources/papers/proceedings11/074-2011.pdf>

Vanam, M., Karidi K., and Dodlapati, S., "SAS to Excel with ExcelXP Tagset." *Proceedings of the PharmaSUG 2010*. Available at: <http://www.lexjansen.com/pharmasug/2010/cc/cc22.pdf>

DelGobbo, V., "Creating Multi-Sheet Excel Workbooks they Easy Way with SAS". *Proceedings of the SAS Global 2007 Forum*. Available at: <http://www2.sas.com/proceedings/forum2007/120-2007.pdf>

Winters, R., "Excellent Ways of Exporting SAS Data to Excel". *Proceedings of the NESUG2004*. Available at: <http://www.nesug.org/proceedings/nesug04/io/io09.pdf>

Shu, A., "The SAS Programming Experience in the SAS Drug Development (SAS DD) Environment – Comparing with the Regular SAS Programming Environment". *Proceedings of the NESUG2010*. Available at: <http://www.nesug.org/Proceedings/nesug10/hl/hl05.pdf>

Seah, A., "Achieving Efficiencies using SAS® Drug Development". *Proceedings of the SAS Global 2011 Forum*. Available at: <http://support.sas.com/resources/papers/proceedings11/203-2011.pdf>

## ACKNOWLEDGMENTS

The authors wish to thank colleagues Nancy Wang, Donald Graff, and Shawna Fox who gave their time to review and provide valuable feedback.

## CONTACT INFORMATION

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

Name: Chun Feng  
Enterprise: Celerion  
Address: 621 Rose Street Lincoln, NE 68502  
Work Phone: (402)437-4799  
E-mail: [chun.feng@celerion.com](mailto:chun.feng@celerion.com), [fengchun200283@hotmail.com](mailto:fengchun200283@hotmail.com)

Name: Xiaopeng Li  
Enterprise: Celerion  
Address: 621 Rose Street Lincoln, NE 68502  
Work Phone: (402)437-6260  
E-mail: [xiaopeng.li@celerion.com](mailto:xiaopeng.li@celerion.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.