



From *Exchanging Data Between SAS® and Microsoft Excel*. Full book available for purchase [here](#).

# Contents

<b>Preface .....</b>	<b>xi</b>
<b>About This Book .....</b>	<b>xiii</b>
<b>About The Author .....</b>	<b>xvii</b>
<b>Chapter 1: Easy Data Movement between SAS and Microsoft Excel .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Examination of Excel Files .....	2
1.2.1 Purpose .....	2
1.2.2 Excel Data Types .....	2
1.2.3 General Excel Workbook Limitations .....	2
1.2.4 Excel Workbook Formatting Groups .....	3
1.2.5 Excel Data Ranges .....	4
1.3 Examples of Copy-and-Paste Techniques .....	4
1.3.1 Highlight, “Cut” or “Copy,” and Then “Paste” .....	5
1.3.2 Convert Text Data to Excel Column Data Fields .....	5
1.3.3 Copy Data to the SAS Enhanced Editor Window for Use in a SAS Program .....	7
1.3.4 Save Multiple Lines of Text in a Single Excel Cell .....	8
1.3.5 Converting Excel Tables to Text .....	9
1.4 Accessing Excel Data from the SAS Explorer Window and Toolbar .....	9
1.4.1 SAS Explorer Window and Toolbar Processing Method Descriptions .....	10
1.4.2 Picking the Export Wizard from the SAS Explorer Window “Export” Menu .....	11
1.4.3 Using the “Copy Contents to Clipboard” Option of the SAS Explorer Window .....	11
1.4.4 Selecting the “Save as Html” Option of the SAS Explorer Window .....	12
1.4.5 Using the “View in Excel” Option to Copy Data to Excel Files via HTML .....	13
1.4.6 SAS Toolbar File Option, the Gateway to the SAS Export / Import Wizards .....	15
1.4.7 Choosing the “Export Data” SAS Toolbar (Export Wizard) File Option .....	16
1.4.8 Electing the “Import Data” SAS Toolbar (Import Wizard) File Option .....	16
1.4.9 Using the Export / Import Wizards in a 32/64-Bit Mixed Environment .....	17
1.5 Chapter Summary .....	18
<b>Chapter 2: Use PROC EXPORT to Write SAS Data to External Files and Excel Workbooks .....</b>	<b>19</b>
2.1 Introduction .....	19
2.2 Purpose .....	20
2.3 Syntax of the SAS EXPORT Procedure .....	20
2.4 Data Access Methods for Excel Files Supported by PROC EXPORT .....	21
2.5 Overview of the Examples .....	22

2.6 List of Examples.....	23
Example 2.1 SAS Code to Export Data to an Excel 4 or Excel 5 Format File.....	23
Example 2.2 PROC EXPORT Using the DBMS=DLM Option .....	24
Example 2.3 PROC EXPORT Using the DBMS=EXCEL Option.....	25
Example 2.4 PROC EXPORT Using the DBMS=EXCELCS Option.....	27
Example 2.5 SAS Code to Export Data to an Excel File with No Column Headers .....	28
Example 2.6 SAS Code to Export Data to a Network Windows Computer .....	28
2.7 Conclusion.....	29
<b>Chapter 3: Use PROC IMPORT to Read External Data Files and Excel Workbooks into SAS .....</b>	<b>31</b>
3.1 Introduction .....	31
3.2 Purpose .....	32
3.3 Syntax of the SAS IMPORT Procedure .....	32
3.4 Data Access Methods for Excel Files Supported by PROC IMPORT .....	33
3.5 Overview of the Examples.....	34
3.6 List of Examples.....	34
Example 3.1 PROC IMPORT Using the DBMS=EXCEL4 or EXCEL5 Option .....	35
Example 3.2 PROC IMPORT Using the DBMS=DLM Option.....	35
Example 3.3 PROC IMPORT Using the DBMS=EXCEL Option.....	37
Example 3.4 PROC IMPORT Using the DBMS=EXCELCS Option.....	40
Example 3.5 PROC IMPORT Using the DBMS=XLS or XLSX to Select Columns .....	42
Example 3.6 PROC IMPORT Using the DBMS=XLS or XLSX to Select Rows.....	43
Example 3.7 PROC IMPORT Using the DBMS=XLS or XLSX to Select Excel Ranges.....	44
3.7 Conclusion.....	45
<b>Chapter 4: Using the SAS LIBNAME to Process Excel Files .....</b>	<b>47</b>
4.1 Introduction .....	47
4.2 Purpose .....	48
4.3 Excel-Specific Features of the SAS LIBNAME Statement .....	48
4.4 Syntax of the SAS LIBNAME Statement .....	49
4.5 LIBNAME Statement ENGINE CONNECTION OPTION Descriptions .....	50
4.5.1 HEADER Option to Read Variable Names .....	50
4.5.2 MIXED Option to Select Data Types.....	50
4.5.3 PATH Option to Define Physical File Locations .....	51
4.5.4 VERSION Option to Identify Excel File Version .....	52
4.5.5 PROMPT Option to Interactively Assign a Libref .....	52
4.5.6 Other Common SAS PC File LIBNAME Options.....	53
4.6 Excel-Specific Dataset Options.....	53
4.7 UNIX, LINUX, and 64-Bit Windows Connection Options .....	54
4.8 Overview of the Examples.....	55
4.8 List of Examples .....	55
4.9 Examples.....	56
Example 4.1 Using the Engine Connection HEADER Option .....	56
Example 4.2 Using the Engine Connection MIXED Option .....	57
Example 4.3 Using the Engine Connection PATH Option .....	58

Example 4.4 Using the Engine Connection VERSION Option .....	58
Example 4.5 Using Named Literals with the LIBNAME Statement.....	59
Example 4.6 Using PROC CONTENTS to Examine an Excel Workbook.....	60
Example 4.7 Using Dataset Options to Process Date and Time Values .....	62
Example 4.8 Using Dataset Options to Process Variable Type Conversions .....	63
Example 4.9 Processing on 64-Bit Operating Systems .....	64
4.10 Conclusion.....	65
<b>Chapter 5: SAS Enterprise Guide Methods and Examples .....</b>	<b>67</b>
5.1 Introduction .....	67
5.2 Purpose .....	68
5.3 Typical Methods to Access Excel from SAS Enterprise Guide .....	68
5.4 Overview of the Examples.....	68
5.5 List of Examples.....	68
5.6 Examples.....	69
Example 5.1 Using the Export Method with Enterprise Guide.....	69
Example 5.2 Using the “Send To” Method.....	71
Example 5.3 Using the “Send To” Method to Output a Graph or Report .....	71
Example 5.4 Using the “Export” Method to Output a Graph or Report .....	75
Example 5.5 Using “Open” or “Import” Toolbar Options to Read Excel Workbooks .....	77
Example 5.6 Using the “Import Data” Toolbar Option to Read a Range of Cells .....	80
5.7 Conclusion.....	84
<b>Chapter 6: Using JMP to Share Data with Excel.....</b>	<b>85</b>
6.1 Introduction .....	85
6.2 Purpose .....	85
6.3 Methods of Sharing Data between JMP and Excel .....	86
6.4 List of Examples.....	87
6.5 Examples.....	87
Example 6.1 Within Excel, Set the JMP Preferences for Loading Excel Data .....	87
Example 6.2 Reading Data from Excel to JMP .....	88
Example 6.3 Writing Data from JMP to Excel.....	89
6.6 Conclusion.....	90
<b>Chapter 7: SAS Add-In for Microsoft Office (Excel) .....</b>	<b>91</b>
7.1 Introduction .....	91
7.2 Purpose .....	91
7.3 Methods of Sharing Data Using SAS Add-In for Microsoft Office .....	92
7.4 List of Examples.....	94
7.5 Examples.....	94
Example 7.1 Open a SAS Dataset Using SAS Add-In for Microsoft Office.....	94
Example 7.2 Open a SAS Report Dataset (*.srx) Using SAS Add-In for Microsoft Office.....	99
7.6 Conclusion.....	105

<b>Chapter 8: Creating Output Files with ODS for Use by Excel .....</b>	<b>107</b>
8.1 Introduction .....	108
8.2 Purpose .....	108
8.3 An Introduction to SAS Tagset Templates That Create Files for Excel .....	109
8.3.1 How to Locate a Tagset Template.....	110
8.4 Difference Between an ODS Tagset and an ODS Destination .....	111
8.5 Syntax of the ODS CSV and CSVALL Output Processes .....	111
8.6 CSV and CSVALL Tagset Options .....	111
8.7 Overview of CSV and CSVALL Examples .....	113
8.8 CSV and CSVALL Examples to Write *.csv Files.....	113
Example 8.8.1 Simple CSV and CSVALL File Default Output Differences .....	113
Example 8.8.2 CSV and CSVALL Title and Footnote Output Differences .....	115
Example 8.8.3 Write Currency Values as Unformatted Numbers .....	118
Example 8.8.4 Change Delimiters When Outputting Data with CSV Tagset .....	120
Example 8.8.5 Save Leading Zeroes in Character Fields Sent to Excel .....	123
8.9 Syntax of ODS MSOFFICE2K Output Processes to Write HTML Files .....	124
8.10 MSOFFICE2K Tagset Template Options .....	125
8.11 Overview of MSOFFICE2K Examples.....	126
8.12 MSOFFICE2K Examples to Write HTML Files .....	126
Example 8.12.1 Generating an HTML Output File with No Options .....	126
Example 8.12.2 Generating an HTML File Using the Summary_Vars Option .....	127
8.13 Syntax of the ODS EXCELXP Tagset Template Output Processes .....	128
8.14 ODS EXCELXP Tagset Options .....	130
8.15 Overview of EXCELXP Examples .....	132
8.16 EXCELXP Examples to Write XML Files .....	133
Example 8.16.1 Generating an XML Output File with No Options .....	133
Example 8.16.2 Adjusting Column Width Using Tagset Template Options.....	134
Example 8.16.3 Tagset Option to Hide Columns While Writing the File .....	135
Example 8.16.4 Apply an Excel “AUTOFILTER” to Selected Output Columns.....	136
Example 8.16.5 Using Multiple Options to Produce a “Ready-to-Print” Spreadsheet.....	137
Example 8.16.6 Creating a Table of Contents in an Excel Workbook.....	138
Example 8.16.7 Methods of Naming Excel Worksheets .....	140
Example 8.16.8 Splitting One Report onto Multiple Excel Worksheets .....	141
Example 8.16.9 Methods of Placing Labels in Excel Worksheet Names .....	142
Example 8.16.10 Use SHEET_INTERVAL= BYGROUP to Create Worksheets .....	143
Example 8.16.11 Use SHEET_INTERVAL= PROC to Create Worksheets .....	144
Example 8.16.12 Build Separate Worksheets with Titles on Each Sheet .....	146
8.17 The New ODS Destination EXCEL for Writing Workbooks.....	147
8.18 Conclusion.....	148
<b>Chapter 9: Accessing Excel with OLE DB or ODBC Application Program Interfaces (API Methods) .....</b>	<b>149</b>
9.1 Introduction .....	149
9.2 Purpose .....	149
9.3 Concept of the OLE DB or ODBC API Processes.....	149

9.4 Guidelines for Setting Up OLE DB or ODBC Connections .....	150
9.5 List of Examples.....	150
9.6 Examples.....	151
Example 9.1 Assign a Libref to an Excel Worksheet with the OLE-DB Dialog Box.....	151
Example 9.2 Using LIBNAME Prompt Mode to Build an OLE-DB Connection .....	152
Example 9.3 Using an OLE-DB init_string to Open an Excel Workbook .....	154
Example 9.4 Using PROC CONTENTS to Verify Excel to OLE DB Connection.....	154
9.7 Conclusion.....	156
<b>Chapter 10: Using PROC SQL to Access Excel Files.....</b>	<b>157</b>
10.1 Introduction .....	157
10.2 Purpose.....	158
10.3 Basic Syntax of the SQL Procedure.....	158
10.4 A Simple Explanation of SQL “PASS-THROUGH” Processing.....	160
10.5 Overview of the Examples.....	160
10.5.1 List of Examples .....	160
10.6 Examples.....	160
Example 10.1 LIBNAME Assignments to Access Excel Using PROC SQL.....	160
Example 10.2 Create an Excel File, Read It with SQL, and Then Compare the Files .....	161
Example 10.3 Use PROC SQL to Read a Subset of Records from an Excel Workbook .....	162
Example 10.4 Use PROC SQL Pass-Through Facilities to Process an Excel File .....	162
Example 10.5 Read a Pre-defined Range of Cells from an Excel Workbook.....	163
Example 10.6 Calculate a New Variable within the SQL Code and Sort the Output .....	165
Example 10.7 Examine the Contents and Structure of an Excel Workbook with a “PCFILES:” Special Query .....	165
10.7 Conclusion.....	166
<b>Chapter 11: Using DDE to Read and Write to Excel Workbooks.....</b>	<b>167</b>
11.1 Introduction .....	167
11.2 Purpose.....	167
11.3 Basic Concept of the DDE Client-Server Environment.....	168
11.3.1 How the DDE Client-Server Relationship Works .....	168
11.3.2 General DDE Syntax and Options .....	168
11.4 List of User-Written SAS Macros That Can Enhance DDE Processing .....	171
11.4.1 SAS Macro to Start Excel .....	171
11.4.2 SAS Macro to SAS to Issue Commands to Excel .....	172
11.4.3 SAS Macro to Define a Range of Excel Cells for Processing .....	172
11.4.4 SAS Macro to Save the Contents of an Excel Workbook .....	174
11.4.5 SAS Macro to Close Excel Workbook.....	174
11.4.6 SAS Macro to Write All or Selected Variables to an Excel Output Workbook .....	175
11.5 List of Examples.....	177
11.6 Examples.....	177
Example 11.6.1 The Hello World Project .....	177
Example 11.6.2 The Hello World Project When the Excel Workbook Is Closed .....	179
Example 11.6.3 The Hello World Project Using NOTAB and LRECL= Options .....	180

Example 11.6.4 Writing “Hello World” to an Excel File Using DDE Macros .....	182
Example 11.6.5 Writing a SAS Dataset to an Excel File Using the SAS_2_EXCEL DDE Macro .....	184
11.7 Conclusion .....	187
<b>Chapter 12: Building a System of Excel Macros Executable by SAS .....</b>	<b>189</b>
12.1 Introduction .....	189
12.2 Purpose .....	190
12.3 General Design of a Tool to Control Excel Macros from SAS.....	190
12.3.1 Prepare a SAS File and Execute Excel to Process the Output.....	191
12.3.2 Prepare Excel to Open the File Output by SAS .....	192
12.3.3 Prepare Excel Macros to Reformat the Excel Workbooks .....	194
12.4 Automate the Tool So That SAS Creates a Formatted Excel Output Workbook .....	197
12.4.1 Eliminate the Manual Steps from the Processing.....	197
12.4.2 Create a SAS Output File with More Data and Control Information .....	202
12.4.3 Create an Excel Macro to Process the Output SAS File .....	203
12.4.4 Build an Excel Graph Using an Excel Macro .....	207
12.5 Conclusion .....	209
<b>Chapter 13: Building a System of Microsoft Windows Scripts to Control Excel Macros .....</b>	<b>211</b>
13.1 Introduction .....	211
13.2 Purpose .....	212
13.3 Guidelines for Building and Using a VBS/VBA Macro Library.....	214
13.3.1 Create Naming Conventions for Storing and Executing VBS/VBA Macros .....	214
13.3.2 Set Up Workstation Options.....	215
13.3.3 Where to Store VBS/VBA Scripts and Macros .....	217
13.3.4 SAS Code to Execute a Visual Basic Script.....	219
13.3.5 Build a Parameter-Driven VBS Script to Control the Execution of Excel.....	220
13.3.6 Build a Control Macro for Each Excel Report .....	223
13.4 Conclusion .....	229
<b>Chapter 14: Create an Excel Workbook That Runs SAS Programs .....</b>	<b>231</b>
14.1 Introduction .....	231
14.2 Purpose .....	232
14.3 Guidelines for Building an Excel User Form Interface .....	233
14.3.1 Common Excel and Excel User Form Terms.....	233
14.3.2 Introduction to the Integrated Development Environment (IDE).....	235
14.3.3 Using the Integrated Development Environment (IDE) Toolbox Menu.....	236
14.3.4 Building a Sample Integrated Development Environment (IDE) Menu .....	237
14.3.5 Linking the Integrated Development Environment (IDE) Menu and the Data .....	239
14.3.6 Storing Control Information in the Excel Workbook Worksheets .....	240
14.3.7 Set Up Control Variables to Access Data Stored in the Workbook .....	241
14.3.8 Learn How to Make the Excel UserForm Execute .....	245
14.4 Excel VBA Routines to Make the Workbook UserForm Active .....	248
14.4.1 Initialize the User Form .....	248
14.4.2 Write the User Parameters to a File in a Working Directory.....	253
14.4.3 Copy Source Program from a Production Directory to the Working Directory.....	253

14.4.4 Verify the Output Batch File Points to the Correct SAS Run Time Module .....	254
14.4.5 A Routine to Save the Changes and Exit the Program.....	255
14.4.6 Directory Structure Associated with the Processing .....	255
14.4.7 Common Issues That Might Occur.....	257
14.4.8 Prepare a VBA Macro to Process Your Output Report .....	258
14.5 Conclusion .....	259
<b>Index .....</b>	<b>261</b>

From *Exchanging Data Between SAS® and Microsoft Excel: Tips and Techniques to Transfer and Manage Data More Efficiently*, by William E. Benjamin, Jr. Copyright © 2015, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.