



CHAPTER 19

Procedures

SAS Procedures under Windows 355

CATALOG 355

CIMPORT 356

CONTENTS 357

CONVERT 358

CPORT 361

DATASETS 361

OPTIONS 363

PMENU 364

PRINTTO 365

SORT 367

SAS Procedures under Windows

Base SAS procedures allow you to perform statistical computations, create reports, and manage your data. Most of the base SAS procedures are completely described in *SAS Procedures Guide*. The procedures described here have syntax or behavior that is specific to Windows.

CATALOG

Manages entries in SAS catalogs

Windows specifics: FILE= option in the CONTENTS statement

Syntax

```
PROC CATALOG CATALOG=<libref.>catalog <ENTRYTYPE=etype> <KILL>;  
  CONTENTS <OUT=SAS-data-set> <FILE=fileref>;
```

Note: This is a simplified version of the CATALOG procedure syntax. For the complete syntax and its explanation, see the CATALOG procedure in *SAS Procedures Guide*. Δ

fileref

names a file specification that is specific to the Windows operating environment.

Details

The CATALOG procedure manages entries in SAS catalogs.

The FILE= option in the CONTENTS statement of the CATALOG procedure accepts a file specification that is specific to the Windows operating environment. If an unquoted file specification is given in the FILE= option, but no FILENAME statement, SET system option, or Windows environment variable is used to define the file specification, the file is named *file-specification*.LST and is stored in the working directory. For example, if MYFILE is not a fileref defined by the FILENAME statement, the SET system option, or a Windows environment variable, and you submit the following statements, the file MYFILE.LST, containing the list of contents for SASUSER.PROFILE, is created in your working directory:

```
proc catalog catalog=sasuser.profile;
  contents file=myfile;
run;
```

See Also

- CATALOG procedure in *SAS Procedures Guide*

CIMPORT

Restores a transport file created by the CPORT procedure

Windows specifics: Name and location of transport file

Syntax

PROC CIMPORT *destination=libref* | *<libref.>.member-name <option(s)>*;

destination

specifies the name and location of a file to be transported.

Details

The CIMPORT procedure *imports* a transport file that was created (*exported*) by the CPORT procedure.

Coupled with the CPORT procedure, the CIMPORT procedure enables you to move catalogs and data sets from one operating environment to another.

Note: PROC CIMPORT processes a file generated by PROC CPORT, not a transport file generated by the XPORT engine. △

When you use the CIMPORT procedure under Windows, remember the following:

- The value of the INFILE= option can be a fileref defined in a FILENAME statement, a quoted Windows pathname, or an environment variable.
- If you omit the INFILE= option and have not defined the reserved fileref SASCAT, the SAS System tries to read from a file named SASCAT.DAT in your working

directory. If no file by that name exists, the following error message is issued and the procedure terminates, assuming that C:\SAS has been defined as the working directory:

```
ERROR: Physical file does not exist, C:\SAS\SASCAT.DAT
```

- If you have not transferred the file created by PROC CPORT in binary format, PROC CIMPORT cannot read the file, and you receive the following message:

```
ERROR: Given transport file is bad.
```

See Also

- CIMPORT procedure in *SAS Procedures Guide*
- Procedure: “CPORT” on page 361

CONTENTS

Prints descriptions of the contents of one or more SAS data library files

Windows specifics: Engine/Host Dependent Information output

Syntax

PROC CONTENTS <*option(s)*>;

option(s)

For an explanation of the available options, see the CONTENTS procedure in *SAS Procedures Guide*.

Details

The CONTENTS procedure shows the contents of a SAS data set and prints the directory of the SAS data library.

While most of the printed output generated by the CONTENTS procedure is the same across all operating environments, the **Engine/Host Dependent Information** output depends on both the operating environment and the engine. Output 19.1 on page 357 shows an example of the **Engine/Host Dependent Information** generated for the V8 engine from the following statements:

```
proc contents data=states;
  title 'Contents of a Version 8 Data Set';
run;
```

Output 19.1 Engine/Host Dependent Information from PROC CONTENTS Using the V8 Engine

```

                                Contents of a Version 8 Data Set                                1

                                The CONTENTS PROCEDURE

Data Set Name: WORK.STATES                Observations:      51
Member Type:  DATA                      Variables:         3
Engine:       V8                         Indexes:           0
Created:      11:21 Tuesday, February 2, 1999  Observation Length: 48
Last Modified: 11:21 Tuesday, February 2, 1999  Deleted Observations: 0
Protection:                                Compressed:       NO
Data Set Type:                               Sorted:           NO
Label:

-----Engine/Host Dependent Information-----

Data Set Page Size:      8192
Number of Data Set Pages:  1
First Data Page:         1
Max Obs per Page:        169
Obs in First Data Page:   51
Number of Data Set Repairs: 0
File Name:                C:\TEMP\SAS Temporary Files\TD257\states.sas7bdat
Release Created:          8.00.00B
Host Created:             WIN_NT

-----Alphabetic List of Variables and Attributes-----

#   Variable   Type   Len   Pos   Format
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
3   COUNT      Num    8     0     COMMA7.
1   STATE      Char   14    8
2   VIDEO      Char   20    22

```

The engine name (V8) is listed in the header information. The **Engine/Host Dependent Information** describes attributes of the data set, such as the data set page size and the maximum number of observations per page. For more information about how to interpret the data set size information, see “Calculating Data Set Size” on page 156.

See Also

- CONTENTS procedure in *SAS Procedures Guide*
- the section on starting with SAS data sets in *SAS Language and Procedures: Usage*

CONVERT

Converts BDMP, OSIRIS and SPSS/PC system files to SAS data sets

Windows specifics: All

Syntax

PROC CONVERT *product-specification* <option(s)>

product-specification

is required and can be one of the following:

BDMP=*fileref* <(CODE=*code*CONTENT=*content-type*)>

converts into a SAS data set the first member a BMDP save file created under DOS. Here is an example:

```
filename save 'c:\myidr\bmdp.dat';
proc convert bmdp=save;
run;
```

If you have more than one save file in the BMDP file referenced by the *fileref* argument, you can use two options in parentheses after *fileref*. The CODE= option lets you specify the code of the save file you want, and the CONTENT= option lets you give the content of the save file. For example, if a file with CODE=JUDGES has a content of DATA, you can use the following statement:

```
filename save 'c:\mydir\bmdpl.dat';
proc convert bmdp=save(code=judges
                        content=data);
run;
```

OSIRIS=*fileref*

specifies a fileref for the OSIRIS file to be converted into a SAS data set. If you use this product specification, you must also use the DICT= option, which specifies the OSIRIS dictionary to use.

SPSS=*fileref*

specifies a fileref for the SPSS/PC file to be converted into a SAS data set. The SPSS/PC file can be in the following formats:

- SPSS/PC format (created under DOS).
- SPSS Portable File Format (from any operating environment). This format is also called the export format.

option-list

DICT=*fileref*

specifies a fileref of the dictionary file for the OSIRIS file. The DICT= option is valid only when used with the OSIRIS product specification.

FIRSTOBS=*n*

gives the number of the observation where the conversion is to begin. This enables you to skip over observations at the beginning of the OSIRIS or SPSS/PC system file.

OBS=*n*

specifies the number of the last observation to convert. This option enables you to exclude observations at the end of the file.

OUT= *SAS-data-set*

names the SAS data set created to hold the converted data. If the OUT= option is omitted, the SAS System still creates a WORK data set and automatically names it DATA*n*, just as if you omitted a data set name in a DATA statement. If it is the first such data set in a job or session, the SAS System names it DATA1, the second is DATA2, and so on. If the OUT= option is omitted or if you do not specify a two-level name (including a libref) in the OUT= option, the converted data set is stored in your WORK data library and by default it is not permanent.

Details

The CONVERT procedure converts a BMDP, OSIRIS or SPSS data file to a SAS data set. It produces one output data set, but no printed output. The new data set contains the same information as the input system file; exceptions are noted in “Output Data Sets” on page 360. The BMDP, OSIRIS and SPSS engines provide more extensive capabilities.

Because the BMDP, OSIRIS and SPSS/PC products are maintained by other companies or organizations, changes may be made that make the system files incompatible with the current version of PROC CONVERT. SAS Institute only upgrades PROC CONVERT to support changes made to these products when a new version of the SAS System is available.

Missing Values If a numeric variable in the input data set has either no value or a system missing value, PROC CONVERT assigns it a missing value.

Output Data Sets This section describes the attributes of the output SAS data set for each *product-specification* value.

CAUTION:

Be sure that the translated names are unique. Variable names can sometimes be translated by the SAS System. To ensure the procedure works correctly, be sure your variables are named in such a way that translation results in unique names. Δ

BMDP output Variable names from the BMDP save file are used in the SAS data set, but nontrailing blanks and all special characters are converted to underscores in the SAS variable names. The subscript in BMDP variable names, such as x(1), becomes part of the SAS variable name, with the parentheses omitted: X1. Alphabetic BMDP variables become SAS character variables of corresponding length. Category records from BMDP are not accepted.

OSIRIS Output For single-response variables, the V1-V9999 name becomes the SAS variable name. For multiple-response variables, the suffix R*n* is added to the variable name, where *n* is the response. For example, V25R1 is the first response of the multiple-response variable V25. If the variable after V1000 has 100 or more responses, responses above 99 are eliminated. Numeric variables that OSIRIS stores in character, fixed-point binary, or floating-point binary mode become SAS numeric variables. Alphabetic variables become SAS character variables; any alphabetic variable of length greater than 200 is truncated to 200. The OSIRIS variable description becomes a SAS variable label, and OSIRIS print formats become SAS formats.

SPSS Output SPSS variable names and variable labels become variable names and labels without change. SPSS alphabetic variables become SAS character variables. SPSS blank values are converted to SAS missing values. SPSS print formats become SAS formats, and the SPSS default precision of no decimal places becomes part of the variables' formats. SPSS value labels are not copied. DOCUMENT data are copied so that PROC CONTENTS can display them.

Comparison

The CONVERT procedure is closely related to the BMDP, OSIRIS and SPSS interface library engines. (In fact, the CONVERT procedure uses these engines.) For example, the following two sections of code provide identical results:

```
□ filename myfile 'myspss.dat';
  proc convert spss=myfile out=temp;
    run;
```

```

□ libname myfile spss 'myspss.dat';
  data temp;
    set myfile._first_;
  run;

```

However, the BMDP, OSIRIS and SPSS engines have more extensive capabilities than PROC CONVERT.

See Also

- CONVERT procedure in *SAS Procedures Guide*
- “Reading BMDP, OSIRIS and SPSS Files” on page 99

CPORT

Writes SAS data sets and catalogs into a special format in a transport file

Windows specifics: Name and location of transport file

Syntax

PROC CPORT *source-type=libref* | *<libref.>member-name<option(s)>*;

Note: This is a simplified version of the CPORT procedure syntax. For the complete syntax and its explanation, see the CPORT procedure in *SAS Procedures Guide*. △

libref

specifies the name and location of the file to be transported.

Details

The CPORT procedure writes SAS data sets, SAS catalogs, or SAS data libraries to sequential file formats (transport files). Use PROC CPORT with the CIMPORT procedure to move files from one environment to another.

The value of the FILE= option can be a fileref defined in a FILENAME statement, a quoted Windows pathname, or an environment variable.

If you do not use the FILE= option and have not defined the reserved fileref SASCAT, a file named SASCAT.DAT is created in your working directory.

See Also

- CPORT procedure in *SAS Procedures Guide*
- Procedure: “CIMPORT” on page 356
- “Transferring SAS Files between Operating Systems” on page 102

DATASETS

Lists, copies, renames, and deletes SAS files and manages indexes for and appends SAS data sets in a SAS data library

Windows specifics: Directory information; CONTENTS statement output

Syntax

```
PROC DATASETS <options(s)>;
  CONTENTS<options(s)>;
```

option(s)

This is a simplified version of the DATASETS procedure syntax. For the complete syntax and its explanation, see the DATASETS procedure in *SAS Procedures Guide*.

Details

The DATASETS procedure is a utility procedure that manages your SAS files.

The SAS data library information displayed in the SAS log by the DATASETS procedure depends on the operating environment and the engine. Output 19.2 on page 362 is an example SAS log that shows the information (for the V8 engine) that the DATASETS procedure generates under Windows.

Output 19.2 SAS Data Library Information from PROC DATASETS

```
4  proc datasets library=work;
      -----Directory-----
      Libref:          WORK
      Engine:          V8
      Physical Name:   C:\TEMP\SAS Temporary Files\_TD2$7
      File Name:      C:\TEMP\SAS Temporary Files\_TD2$7

      #  Name      Memtype      File
      size  Last modified
      $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      1  CLASS    DATA          16384  02FEB1999:08:38:37
      2  HAT      DATA          311296 02FEB1999:08:58:20
      3  ODSOUT   ITEMSTOR       61440  02FEB1999:10:17:22
      4  STATES   DATA          16384  02FEB1999:11:22:56
5  run;
```

The output shows you the libref, engine, and physical name associated with the library, as well as the names and other properties of the SAS files contained in the library.

The CONTENTS statement of the DATASETS procedure generates the same **Engine/Host Dependent Information** as the CONTENTS procedure.

See Also

- DATASETS procedure in *SAS Procedures Guide*
- Procedure: “CONTENTS” on page 357
- the section on modifying data set names and attributes in *SAS Language and Procedures: Usage*

OPTIONS

Lists the current values of all SAS system options

Windows specifics: Host options

Syntax

PROC OPTIONS <*options(s)*>;

option(s)

This is a simplified version of the OPTIONS procedure syntax. For the complete syntax and its explanation, see the OPTIONS procedure in *SAS Procedures Guide*.

Details

The OPTIONS procedure lists the current settings of the SAS system options.

The options displayed by the OPTIONS procedure that are not host specific (session and configuration) are the same for every operating environment, although the default values may differ slightly. However, the host (environment-specific) options displayed by this procedure are different for each operating environment. Output 19.3 on page 363 shows some sample host options displayed for the Windows environment, as generated by this code:

```
proc options host;
run;
```

Output 19.3 Host Options Displayed by PROC OPTIONS

```

HOST OPTIONS:
  ALTLOG=           Specifies the destination for a copy of the SAS log.
  ALTPRINT=        Specifies the destination for a copy of the SAS procedure output file.
  AUTHSERVER=      Specify the authentication server or domain.
  AUTOEXEC=        Specifies the autoexec file to be used.
  NOAUTOMATION     Do not enable SAS for OLE automation server support
  AWSCONTROL=(SYSTEMMENU MINMAX TITLE)
                  Used to customize the appearance for the SAS AWS. Valid parameters are:
                  TITLE/NOTITLE SYSTEMMENU/NOYSTEMMENU MINMAX/NOMINMAX.
  AWSDEF=( 0 0 91 100)
                  Specify the initial size and position of the SAS AWS. This should be
                  specified as follows: 0 0 100 100
  AWSMENU          Specify whether the main window's (AWS) menu should be shown or not.
  NOAWSMENUMERGE  Host specific items should not be added to the main window's (AWS) menu.

...
NOTE: PROCEDURE OPTIONS used:
      real time           0.35 seconds
      user cpu time       0.12 seconds
      system cpu time     0.23 second
      Memory              2k

```

The option values listed in Output 19.3 on page 363 are examples only. The output of PROC OPTIONS depends on many things. Some option values depend on what method you use to run the SAS System. For example, the default line size under the SAS windowing environment is 75 lines on a VGA display, while it is 132 lines in batch mode. Also, the way you have set up your process affects the default values of system options. For example, the default value of the SASAUTOS= option depends on where you store your autocall macros.

Using PROC OPTIONS, you can check the values of all system options. If you want more information about a particular host option, refer to “SAS System Options under Windows” on page 395 or the Host Information in SAS online Help.

See Also

- OPTIONS procedure in *SAS Procedures Guide*
- “SAS System Options under Windows” on page 395

PMENU

Defines pull-down menu facilities for windows created with SAS software

Windows specifics: ACCELERATE= option accepted for four key combinations

Syntax

```

PROC PMENU <CATALOG=<libref.>catalog> <DESC 'entry-description'>;
  ITEM command <option(s)>;
  ITEM 'menu-item' <option(s)>;
  ACCELERATE=name-of-key;

```

This is a simplified version of the PMENU procedure syntax. For the complete syntax and its explanation, see the PMENU procedure in *SAS Procedures Guide*.

ACCELERATE=*name-of-key*

defines a key sequence that can be used instead of selecting an item. When you press the key sequence, it has the same effect as selecting the item from the menu bar or pull-down menu.

Under Windows, the ACCELERATE= option in the ITEM statement is accepted only for the following key combinations:

- Ctrl+C (Copy)
- Ctrl+V (Paste)
- Ctrl+Z (Undo)
- Ctrl+X (Cut).

Details

The PMENU procedure defines pull-down menus that can be used in DATA step windows, macro windows, SAS/AF and SAS/FSP windows, or in any SAS application that enables you to specify customized menus.

If you want your program to be portable between Windows and OS/2, you can use the following alternate key combinations:

- Ctrl+Insert
- Shift+Insert
- Alt+Backspace
- Shift+Delete.

If you use these alternate key combinations in your SAS program, the **Edit** pull-down menu shows the standard key combination; however, you can use either the standard or alternate key combination to activate the menu item.

See Also

- PMENU procedure in *SAS Procedures Guide*

PRINTTO

Defines destinations for SAS procedure output and the SAS log

Windows specifics: Valid values for *file-specification*; UNIT= option

Syntax

PROC PRINTTO <*option(s)*>

Note: This is a simplified version of the PRINTTO procedure syntax. For the complete syntax and its explanation, see the PRINTTO procedure in *SAS Procedures Guide*. △

option(s)

LOG=*file-specification*

PRINT=*file-specification*

can be a fileref defined in a FILENAME statement or function, a quoted Windows pathname, or an environment variable. If the file specification you use is not a

fileref defined by the FILENAME statement function, the SET system option, or a Windows SET command, the destination is a file named *file-specification.LST* or *file-specification.LOG* stored in your working directory. To send your SAS output or log directly to the printer, use a FILENAME statement or function with the PRINTER device-type keyword. (See the first example in the following Examples section.)

UNIT=*nn*

sends your SAS procedure output to the file FT*nn*F001.LST, where *nn* represents the UNIT= value, which can range from 1 to 99. The file is located in the SAS working directory.

Details

The PRINTTO procedure defines destinations for SAS procedure output and for the SAS log.

Examples

Example 1: Redirecting SAS Log Output The following statements redirect any SAS log entries that are generated after the RUN statement to an output file with a fileref of TEST, which is associated with the LPT1: device:

```
filename test printer 'lpt1:';
proc printto log=test;
run;
```

When these statements are issued, a dialog box is opened that informs you PROC PRINTTO is running. All SAS log entries are redirected to the TEST output file as specified; however, they are not printed on the LPT1: device until the output file is closed, either by redirecting the SAS log entries back to the default destination or to another file.

The following statements send any SAS log entries that are generated after the RUN statement to the external file associated with the fileref MYFILE:

```
filename myfile 'c:\mydir\mylog.log';
proc printto log=myfile;
run;
```

Example 2: Redirecting SAS Procedure Output The following statements send any SAS procedure output to a file named MYPRINT.LST in your working directory (assuming MYPRINT is not a previously defined fileref or environment variable):

```
proc printto print=myprint;
run;
```

The following statements send any SAS procedure output to the printer port, which is usually defined by the system as LPT1:

```
proc printto print='lpt1:';
run;
```

Example 3: Restoring the Output Destinations to the Default The following statements (including a PROC PRINTTO statement with no options) redirect the SAS log and procedure output to the original default destinations:

```
proc printto;
run;
```

See Also

- PRINTTO procedure in *SAS Procedures Guide*

SORT

Sorts observations in a SAS data set by one or more variables, then stores the resulting sorted observations in a new SAS data set or replaces the original data set

Windows specifics: Sort utilities available; SORTSIZE= and TAGSORT statement options

Syntax

PROC SORT <option(s)> <collating-sequence-option>;

Note: This is a simplified version of the SORT procedure syntax. For the complete syntax and its explanation, see the SORT procedure in *SAS Procedures Guide* Δ

SORTSIZE=memory-specification

specifies the maximum amount of memory available to the SORT procedure. For further explanation of the SORTSIZE= option, see the following Details section.

TAGSORT

stores only the BY variables and the observation number in temporary files. For further explanation of the TAGSORT option, see the following Details section.

Details

The SORT procedure sorts observations in a SAS data set by one or more character or numeric variables, either replacing the original data set or creating a new, sorted data set. By default under Windows, the SORT procedure uses the ASCII collating sequence.

The SORT procedure uses the sort utility specified by the SORTPGM system option. Under Windows, although all three SORTPGM keywords (HOST, BEST, and SAS) are accepted for compatibility, the SAS sort is always used. You can use all the options available to the SAS sort utility, such as the SORTSEQ and NODUPKEY options. For a complete list of all options available, see the SORT procedure in *SAS Procedures Guide*.

SORTSIZE= Option Under Windows, you can use the SORTSIZE= option in the PROC SORT statement to limit the amount of memory available to the SORT procedure. This option may reduce the amount of swapping the SAS System must do to sort the data set. If PROC SORT needs more memory than you specify, it creates a temporary utility file in your SASWORK directory to store the data in. The SORT procedure's algorithm can swap data more efficiently than Windows can.

The syntax of the SORTSIZE= option is as follows:

SORTSIZE=memory-specification

where *memory-specification* can be one of the following:

- | | |
|-----------|---|
| <i>n</i> | specifies the amount of memory in bytes. |
| <i>nK</i> | specifies the amount of memory in 1-kilobyte multiples. |
| <i>nM</i> | specifies the amount of memory in 1-megabyte multiples. |

The default SAS configuration file sets this option to 2MB using the SORTSIZE= system option. A value of 2MB is optimal for all memory configurations. If your machine has more than 12 MB of physical memory and you are sorting large data sets, setting the SORTSIZE= option to a value greater than 2M might improve performance.

You can override the default value of the SORTSIZE= system option by specifying a different SORTSIZE= value in the PROC SORT statement, or by submitting an OPTIONS statement that sets the SORTSIZE= system option to a new value.

The SORTSIZE= option is also discussed in “Improving Performance of the SORT Procedure” on page 155.

TAGSORT Option The TAGSORT option in the PROC SORT statement is useful when there may not be enough disk space to sort a large SAS data set. When you specify the TAGSORT option, only sort keys (that is, the variables specified in the BY statement) and the observation number for each observation are stored in the temporary files. The sort keys, together with the observation number, are referred to as *tags*. At the completion of the sorting process, the tags are used to retrieve the records from the input data set in sorted order. Thus, in cases where the total number of bytes of the sort keys is small compared with the length of the record, temporary disk use is reduced considerably. You should have enough disk space to hold another copy of the data (the output data set) or two copies of the tags, whichever is greater. Note that while using the TAGSORT option may reduce temporary disk use, the processing time may be much higher. However, on PCs with limited available disk space, the TAGSORT option may allow sorts to be performed in situations where they would otherwise not be possible.

Creating Your Own Collating Sequences If you want to provide your own collating sequences or change a collating sequence that has been provided for you, use the TRANTAB procedure to create or modify translate tables. For complete details on the TRANTAB procedure, see *SAS Procedures Guide*. When you create your own translate tables, they are stored in your PROFILE catalog and they override any translate tables by the same name that are stored in the HOST catalog.

Note: System managers can modify the HOST catalog by copying newly created tables from the SASUSER.PROFILE catalog to the HOST catalog. Then all users can access the new or modified translate table. Δ

If you want to see the names of the collating sequences stored in the HOST catalog (using the SAS Explorer), submit the following statement:

```
dm 'catalog sashelp.host' catalog;
```

Alternatively, you can select the **View** pull-down menu, then select the **Libraries** item, then double-click on the SASHELP library, and then double-click on the HOST catalog. In batch mode, you can use the following statements to generate a list of the contents of the HOST catalog:

```
proc catalog catalog=sashelp.host;
  contents;
run;
```

Entries of type TRANTAB are the collating sequences.

If you want to see the contents of a particular translate table, use the following statements:

```
proc trantab table=table-name;
  list;
run;
```

The contents of the collating sequence are displayed in the SAS log.

See Also

- SORT procedure in *SAS Procedures Guide*
- TRANTAB procedure in *SAS Procedures Guide*
- “Improving Performance of the SORT Procedure” on page 155

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Companion for the Microsoft Windows Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp.555.

SAS Companion for the Microsoft Windows Environment, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-524-8

All rights reserved. Printed in the United States of America.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

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.

The Institute is a private company devoted to the support and further development of its software and related services.