



## CHAPTER

## 13

## The CPORT Procedure

Overview	313
Procedure Syntax	314
PROC CPORT Statement	314
EXCLUDE Statement	318
SELECT Statement	319
TRANTAB Statement	320
Concepts	321
Transporting Password-Protected Data Sets	321
Results	322
Data Control Block Characteristics for Mainframe Environments	322
Examples	322
Example 1: Exporting Multiple Catalogs	322
Example 2: Exporting Individual Catalog Entries	323
Example 3: Exporting a Single SAS Data Set	324
Example 4: Applying a Translation Table	325
Example 5: Exporting Entries Based on Modification Date	326

## Overview

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. *Transport files* are sequential files that each contain a SAS data library, a SAS catalog, or a SAS data set in transport format. The transport format that PROC CPORT writes is the same for all environments and for many releases of the SAS System. In PROC CPORT, *export* means to put a SAS data library, a SAS catalog, or a SAS data set into transport format. PROC CPORT exports catalogs and data sets, either singly or as a SAS data library. PROC CIMPORT restores (*imports*) the transport file to its original form as a SAS catalog, SAS data set, or SAS data library.

Only PROC CIMPORT can read the transport files that PROC CPORT creates. For information on the transport files that the transport engine creates, see the section on SAS files in *SAS Language Reference: Concepts*.

PROC CPORT also *converts* SAS files, which means that it changes the format of a SAS file from the format appropriate for one version of the SAS System to the format appropriate for another version. For example, you can use PROC CPORT and PROC CIMPORT to move files from earlier releases of the SAS System to more recent releases. In such cases, PROC CIMPORT automatically converts the contents of the transport file as it imports it.

PROC CPORT produces no output (other than the transport files), but it does write notes to the SAS log.

To export and import files, follow these steps:

- 1 Use PROC CPORT to export the SAS files that you want to transport.
- 2 If you are changing operating environments, move the transport file to the new machine by using either communications software or a magnetic medium.

*Note:* If you use communications software to move the transport file, be sure that it treats the transport file as a *binary* file and that it modifies neither the attributes nor the contents of the file.  $\triangle$

- 3 Use PROC CIMPORT to translate the transport file into the format appropriate for the new operating environment or release.

---

## Procedure Syntax

```
PROC CPORT source-type=libref | <libref.>member-name<option(s)>;
  EXCLUDE SAS file(s) | catalog entry(s)</ MEMTYPE=mttype></
    ENTRYTYPE=entry-type>;
  SELECT SAS file(s) | catalog entry(s) </ MEMTYPE=mttype></
    ENTRYTYPE=entry-type>;
  TRANTAB NAME=translation-table-name
    <option(s)>;
```

---

## PROC CPORT Statement

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

To do this	Use this option
Identify the transport file	
Specify the transport file to write to	FILE=
Direct the output from PROC CPORT to a tape	TAPE
Select files to export	
Export copies of all data sets or catalog entries that have a modification date equal to or later than the date you specify	AFTER=
Exclude specified entry types from the transport file	EET=
Include specified entry types in the transport file	ET=
Specify whether to export all generations of a data set	GENERATION=

To do this	Use this option
Specify that only data sets, only catalogs, or both, be moved when a library is exported	MEMTYPE=
Control the contents of the transport file	
Suppress the conversion of displayed character data to transport format	ASIS
Control the exportation of integrity constraints	CONSTRAINT
Control the exportation of indexes with indexed SAS data sets	INDEX
Suppress the compression of binary zeros and blanks in the transport file	NOCOMPRESS
Write all alphabetic characters to the transport file in uppercase	OUTTYPE= UPCASE
Translate specified characters from one ASCII or EBCDIC value to another	TRANSLATE
Export SAS/AF PROGRAM and SCL entries without edit capability when you import them	NOEDIT
Specify that exported catalog entries contain compiled SCL code, but not the source code	NOSRC
Specify a libref associated with a SAS data library	OUTLIB=

## Required Arguments

### ***source-type=libref* | *<libref>member-name***

identifies the type of file to export and specifies the catalog, SAS data set, or SAS data library to export.

#### *source-type*

identifies the file or files to export as a single catalog, as a single SAS data set, or as the members of a SAS data library. The *source-type* argument can be one of the following:

CATALOG | CAT | C

DATA | DS | D

LIBRARY | LIB | L

#### *libref* | *<libref>member-name*

specifies the specific catalog, SAS data set, or SAS data library to export. If the *source-type* argument is CATALOG or DATA, you can specify both a libref and a member name. If the *libref* is omitted, PROC CPORT uses the default library as the *libref*, which is usually the WORK library. If the *source-type* argument is LIBRARY, specify only a *libref*. If you specify a library, PROC CPORT exports only data sets and catalogs from that library. You cannot export other types of files.

## Options

**AFTER=*date***

exports copies of all data sets or catalog entries that have a modification date later than or equal to the date you specify. The modification date is the most recent date when the contents of the data set or catalog entry changed. Specify date as a SAS date literal or as a numeric SAS date value.

**Tip:** You can find out the modification date of a catalog entry by using the CATALOG procedure.

**Featured in:** Example 5 on page 326.

**ASIS**

suppresses the conversion of displayed character data to transport format.

**Interaction:** The ASIS option invokes the NOCOMPRESS option.

**Interaction:** You cannot use both the ASIS option and the OUTTYPE= options in the same PROC CPORT step.

**CONSTRAINT=YES | NO**

controls the exportation of integrity constraints that have been defined on a data set. When you specify CONSTRAINT=YES, all types of integrity constraints are exported for a library; only general integrity constraints are exported for a single data set. When you specify CONSTRAINT=NO, indexes created without integrity constraints are ported, but neither integrity constraints nor any indexes created with integrity constraints are ported. For more information on integrity constraints, see the section on SAS files in *SAS Language Reference: Concepts*.

**Alias:** CON=

**Default:** YES

**Interaction:** You cannot specify both CONSTRAINT= and INDEX= in the same PROC CPORT step.

**Interaction:** If you specify INDEX=NO, no integrity constraints are exported.

**EET=(*etype(s)*)**

excludes specified entry types from the transport file. If *etype* is a single entry type, then you can omit the parentheses. Separate multiple values with a space.

**Interaction:** You cannot use both the EET= option and the ET= option in the same PROC CPORT step.

**ET=(*etype(s)*)**

includes specified entry types in the transport file. If *etype* is a single entry type, then you can omit the parentheses. Separate multiple values with a space.

**Interaction:** You cannot use both the EET= option and the ET= option in the same PROC CPORT step.

**FILE=*fileref* | '*filename*'**

specifies a previously defined fileref or the filename of the transport file to write to. If you omit the FILE= option, then PROC CPORT writes to the fileref SASCAT, if defined. If the fileref SASCAT is not defined, PROC CPORT writes to SASCAT.DAT in the current directory.

*Note:* The behavior of PROC CPORT when SASCAT is undefined varies from one operating environment to another. For details, see the SAS documentation for your operating environment.  $\triangle$

**Featured in:** All examples.

**GENERATION=YES | NO**

specifies whether to export all generations of a SAS data set. To export only the base generation of a data set, specify GENERATION=NO in the PROC CPORT statement. To export a specific generation number, use the GENNUM= data set option when you

specify a data set in the PROC CPORT statement. For more information on generation data sets, see *SAS Language Reference: Concepts*.

*Note:* PROC CIMPORT imports all generations of a data set that are present in the transport file. It deletes any previous generation set with the same name and replaces it with the imported generation set, even if the number of generations does not match. △

**Alias:** GEN=

**Default:** YES for libraries; NO for single data sets

#### **INDEX=YES | NO**

specifies whether to export indexes with indexed SAS data sets.

**Default:** YES

**Interaction:** You cannot specify both INDEX= and CONSTRAINT= in the same PROC CPORT step.

**Interaction:** If you specify INDEX=NO, no integrity constraints are exported.

#### **MEMTYPE=*mtype***

specifies that only data sets, only catalogs, or both, be moved when a library is exported. Values for *mtype* can be

ALL

both catalogs and data sets

CATALOG | CAT

catalogs

DATA | DS

SAS data sets

**Alias:** MT=

**Default:** ALL

**Featured in:** Example 1 on page 322.

#### **NOCOMPRESS**

suppresses the compression of binary zeros and blanks in the transport file.

**Alias:** NOCOMP

**Default:** By default, PROC CPORT compresses binary zeros and blanks to conserve space.

**Interaction:** The ASIS and OUTTYPE= options invoke the NOCOMPRESS option.

*Note:* Compression of the transport file does not alter the flag in each catalog and data set that indicates whether the original file was compressed. △

#### **NOEDIT**

exports SAS/AF PROGRAM and SCL entries without edit capability when you import them.

The NOEDIT option produces the same results as when you create a new catalog to contain SCL code by using the MERGE statement with the NOEDIT option in the BUILD procedure of SAS/AF software.

*Note:* The NOEDIT option affects only SAS/AF PROGRAM and SCL entries. It does not affect FSEDIT SCREEN or FSVIEW FORMULA entries. △

**Alias:** NEDIT

**NOSRC**

specifies that exported catalog entries contain compiled SCL code but not the source code.

The NOSRC option produces the same results as when you create a new catalog to contain SCL code by using the MERGE statement with the NOSOURCE option in the BUILD procedure of SAS/AF software.

**Alias:** NSRC

**OUTLIB=libref**

specifies a libref associated with a SAS data library. If you specify the OUTLIB= option, PROC CIMPORT is invoked automatically to re-create the input data library, data set, or catalog in the specified library.

**Alias:** OUT=

**OUTTYPE=UPCASE**

writes all displayed characters to the transport file and to the OUTLIB= file in uppercase.

**Interaction:** The OUTTYPE= option invokes the NOCOMPRESS option.

**TAPE**

directs the output from PROC CPORT to a tape.

**Default:** The output from PROC CPORT is sent to disk.

**TRANSLATE=(*translation-list*)**

translates specified characters from one ASCII or EBCDIC value to another. Each element of *translation-list* has the form

*ASCII-value-1* TO *ASCII-value-2*

*EBCDIC-value-1* TO *EBCDIC-value-2*

You can use hexadecimal or decimal representation for ASCII values. If you use the hexadecimal representation, values must begin with a digit and end with an x. Use a leading zero if the hexadecimal value begins with an alphabetic character.

For example, to translate all left brackets to left braces, specify the TRANSLATE= option as follows (for ASCII characters):

```
translate=(5bx to 7bx)
```

The following example translates all left brackets to left braces and all right brackets to right braces:

```
translate=(5bx to 7bx 5dx to 7dx)
```

---

## EXCLUDE Statement

**Excludes specified files or entries from the transport file.**

**Tip:** There is no limit to the number of EXCLUDE statements you can use in one invocation of PROC CPORT.

**Interaction:** You can use either EXCLUDE statements or SELECT statements in a PROC CPORT step, but not both.

---

```
EXCLUDE SAS file(s) | catalog entry(s) </ MEMTYPE=mtype></  
ENTRYTYPE=entry-type>;
```

## Required Arguments

### SAS file(s) | catalog entry(s)

specifies either the name(s) of one or more SAS files or the names of one or more catalog entries to be excluded from the transport file. Specify SAS filenames when you export a SAS data library; specify catalog entry names when you export an individual SAS catalog. Separate multiple filenames or entry names with a space. You can use shortcuts to list many like-named files in the EXCLUDE statement. For more information, see “Shortcuts for Specifying Lists of Variable Names” on page 58.

## Options

### ENTRYTYPE=*entry-type*

specifies a single entry type for the catalog entries listed in the EXCLUDE statement. See *SAS Language Reference: Concepts* for a complete list of catalog entry types.

**Restriction:** ENTRYTYPE= is valid only when you export an individual SAS catalog.

**Alias:** ETYPE=, ET=

### MEMTYPE=*mtype*

specifies a single member type for the SAS file(s) listed in the EXCLUDE statement. Valid values are CATALOG or CAT, DATA, or ALL. If you do not specify the MEMTYPE= option in the EXCLUDE statement, then processing is restricted to those member types specified in the MEMTYPE= option in the PROC CPORT statement.

You can also specify the MEMTYPE= option, enclosed in parenthesis, immediately after the name of a file. In parenthesis, MEMTYPE= identifies the type of the file name that just precedes it. When you use this form of the option, it overrides the MEMTYPE= option that follows the slash in the EXCLUDE statement, but it must match the MEMTYPE= option in the PROC CPORT statement:

**Restriction:** MEMTYPE= is valid only when you export a SAS data library.

**Restriction:** If you specify a member type for MEMTYPE= in the PROC CPORT statement, it must agree with the member type that you specify for MEMTYPE= in the EXCLUDE statement.

**Alias:** MTYPE=, MT=

**Default:** If you do not specify MEMTYPE= in the PROC CPORT statement or in the EXCLUDE statement, the default is MEMTYPE=ALL.

---

## SELECT Statement

**Includes specified files or entries in the transport file.**

**Tip:** There is no limit to the number of SELECT statements you can use in one invocation of PROC CPORT.

**Interaction:** You can use either EXCLUDE statements or SELECT statements in a PROC CPORT step, but not both.

**Featured in:** Example 2 on page 323

---

```
SELECT SAS file(s) | catalog entry(s) </ MEMTYPE=mtype> </
ENTRYTYPE=entry-type> ;
```

## Required Arguments

### SAS *file(s)* | **catalog entry(s)**

specifies either the name(s) of one or more SAS files or the names of one or more catalog entries to be included in the transport file. Specify SAS filenames when you export a SAS data library; specify catalog entry names when you export an individual SAS catalog. Separate multiple filenames or entry names with a space. You can use shortcuts to list many like-named files in the SELECT statement. For more information, see “Shortcuts for Specifying Lists of Variable Names” on page 58.

## Options

### ENTRYTYPE=*entry-type*

specifies a single entry type for the catalog entries listed in the SELECT statement. See *SAS Language Reference: Concepts* for a complete list of catalog entry types.

**Restriction:** ENTRYTYPE= is valid only when you export an individual SAS catalog.

**Alias:** ETYPE=, ET=

### MEMTYPE=*mtype*

specifies a single member type for the SAS file(s) listed in the SELECT statement. Valid values are CATALOG or CAT, DATA, or ALL. If you do not specify the MEMTYPE= option in the SELECT statement, then processing is restricted to those member types specified in the MEMTYPE= option in the PROC CPORT statement.

You can also specify the MEMTYPE= option, enclosed in parenthesis, immediately after the name of a member. In parenthesis, MEMTYPE= identifies the type of the member name that just precedes it. When you use this form of the option, it overrides the MEMTYPE= option that follows the slash in the SELECT statement, but it must match the MEMTYPE= option in the PROC CPORT statement.

**Restriction:** MEMTYPE= is valid only when you export a SAS data library.

**Restriction:** If you specify a member type for MEMTYPE= in the PROC CPORT statement, it must agree with the member type that you specify for MEMTYPE= in the SELECT statement.

**Alias:** MTYPE=, MT=

**Default:** If you do not specify MEMTYPE= in the PROC CPORT statement or in the SELECT statement, the default is MEMTYPE=ALL.

---

## TRANTAB Statement

**Specifies translation tables for characters in catalog entries you export.**

**Tip:** You can specify only one table for each TRANTAB statement, but there is no limit to the number of TRANTAB statements you can use in one invocation of PROC CPORT.

**Featured in:** Example 4 on page 325.

**See also:** Chapter 40, “The TRANTAB Procedure,” on page 1291

---



**TRANTAB** NAME=*translation-table-name*  
           <option(s)>;

## Required Arguments

### **NAME=***translation-table-name*

specifies the name of the translation table to apply to the character data in the SAS file you export. The *translation-table-name* is the name of a catalog entry in either the SASUSER.PROFILE catalog or the SASHELP.HOST catalog. PROC CPORT prints an error message in the SAS log if it cannot find the translation table.

*Note:* The translation takes place before PROC CPORT writes to the transport file. △

## Options

### **OPT=**

specifies how to apply the translation table. Use one of the following values for the OPT= option:

#### **DISP**

applies the translation table to all the DISPLAY window text.

#### **SRC**

applies the translation table to all the SCL text.

#### **(DISP SRC)**

applies the translation table to all the DISPLAY window text and SOURCE window text.

**Default:** PROC CPORT applies all options to the specified translation table.

### **TYPE=***(target-list)*

applies the translation table only to the specified targets. If the *target-list* is a single target, then you can omit the parentheses. The *target-list* can be one of the following types:

#### *etype-list*

applies the translation table only to the entries with the catalog entry type you specify.

#### **CATDESC**

applies the translation table to the description of each exported catalog entry.

#### **DATASET**

applies the translation table to the observations, the data set label, and the variable labels in each exported data set.

**Default:** PROC CPORT applies the translation table to all entries and data sets in the specified catalog.

**Featured in:** Example 4 on page 325.

---

## Concepts

---

### Transporting Password-Protected Data Sets

For password-protected data sets, the password(s) are applied to the destination data set when it is imported. If the data set is transported as part of a library, it is not

necessary to supply the password. If the data set is transported singly, you must supply the read password. If you omit the password in the PROC CPORT step, SAS prompts you for the password. If the target SAS engine does not support passwords, then the import will fail. For example, the following SAS code transports a password-protected data set called WORK.ONE:

```
proc cport data=one(read=hithere) file='bin';
```

---

## Results

---

### Data Control Block Characteristics for Mainframe Environments

A common problem when you create or import a transport file under the OS/390 or CMS environments is a failure to specify the correct Data Control Block (DCB) characteristics. When you reference a transport file, you must specify the following DCB characteristics:

LRECL	80
BLKSIZE	8000
RECFM	FB

*Note:* A BLKSIZE value of less than 8000 may be more efficient for your storage device in some cases. The BLKSIZE value should be an exact multiple of the LRECL value. △

Another common problem can occur if you use communications software to move files from another environment to OS/390 or CMS. In some cases, the transport file does not have the proper DCB characteristics when it arrives on OS/390 or CMS. If the communications software does not allow you to specify file characteristics, try the following approach for OS/390:

- 1 Create a file under OS/390 with the correct DCB characteristics and initialize the file.
- 2 Move the transport file from the other environment to the newly created file under OS/390 using binary transfer.

Try this approach for CMS:

- 1 Log onto CMS.
- 2 Connect to the other operating environment.
- 3 Pull the file from the other operating environment.

---

## Examples

---

### Example 1: Exporting Multiple Catalogs

Procedure features:

PROC CPORT statement options:

FILE=  
MEMTYPE=

---

This example shows how to use PROC CPORT to export entries from all of the SAS catalogs in the SAS data library you specify.

## Program

The LIBNAME statement assigns a libref for the SAS data library that contains the source files to be exported. The FILENAME statement assigns a fileref and any operating environment options for file characteristics for the transport file that PROC CPORT creates.

```
libname source 'SAS-data-library';
filename tranfile 'transport-file'
               host-option(s)-for-file-characteristics;
```

The PROC CPORT step executes on the operating environment where the source library is located. MEMTYPE=CATALOG writes all SAS catalogs in the source library to the transport file.

```
proc cport library=source file=tranfile memtype=catalog;
run;
```

## SAS Log

```
NOTE: Proc CPORT begins to transport catalog SOURCE.FINANCE
NOTE: The catalog has 5 entries and its maximum logical record length is 866.
NOTE: Entry LOAN.FRAME has been transported.
NOTE: Entry LOAN.HELP has been transported.
NOTE: Entry LOAN.KEYS has been transported.
NOTE: Entry LOAN.PMENU has been transported.
NOTE: Entry LOAN.SCL has been transported.

NOTE: Proc CPORT begins to transport catalog SOURCE.FORMATS
NOTE: The catalog has 2 entries and its maximum logical record length is 104.
NOTE: Entry REVENUE.FORMAT has been transported.
NOTE: Entry DEPT.FORMATC has been transported.
```

---

## Example 2: Exporting Individual Catalog Entries

Procedure features:

PROC CPORT statement options:

FILE=  
SELECT statement

---

This example shows how to use PROC CPORT to export individual catalog entries, rather than all of the entries in a catalog.

## Program

The LIBNAME and FILENAME statements assign a libref for the source library and the fileref for the transport file, respectively.

```
libname source 'SAS-data-library';
filename tranfile 'transport-file'
               host-option(s)-for-file-characteristics;
```

SELECT writes only the LOAN.SCL entry to the transport file for export.

```
proc cport catalog=source.finance file=tranfile;
select loan.scl;
run;
```

## SAS Log

```
NOTE: Proc CPORT begins to transport catalog SOURCE.FINANCE
NOTE: The catalog has 5 entries and its maximum logical record length is 866.
NOTE: Entry LOAN.SCL has been transported.
```

---

## Example 3: Exporting a Single SAS Data Set

### Procedure features:

PROC CPORT statement option:  
FILE=

---

This example shows how to use PROC CPORT to export a single SAS data set.

## Program

The LIBNAME and FILENAME statements assign a libref for the source library and the fileref for the transport file, respectively.

```
libname source 'SAS-data-library';
filename tranfile 'transport-file'
               host-option(s)-for-file-characteristics;
```

The DATA= specification in the PROC CPORT statement tells the procedure that you are exporting a SAS data set, rather than a library or a catalog.

```
proc cport data=source.times file=tranfile;
run;
```

## SAS Log

```
NOTE: Proc CPORT begins to transport data set SOURCE.TIMES
NOTE: The data set contains 2 variables and 2 observations.
      Logical record length is 16.
NOTE: Transporting data set index information.
```

---

## Example 4: Applying a Translation Table

### Procedure features:

PROC CPORT statement option:

FILE=

TRANTAB statement option:

TYPE=

---

This example shows how to apply a customized translation table to the transport file before PROC CPORT exports it. For this example, assume that you have already created a customized translation table called TTABLE1.

## Program

The LIBNAME and FILENAME statements assign the libname for the source library and the filename for the transport file, respectively.

```
libname source 'SAS-data-library';
filename tranfile 'transport-file'
               host-option(s)-for-file-characteristics;
```

The TRANTAB statement applies the translation that you specify with the customized translation table TTABLE1. TYPE= limits the translation to FORMAT entries.

```
proc cport catalog=source.formats file=tranfile;
      trantab name=ttable1 type=(format);
run;
```

## SAS Log

```
NOTE: Proc CPORT begins to transport catalog SOURCE.FORMATS
NOTE: The catalog has 2 entries and its maximum logical record length is 104.
NOTE: Entry REVENUE.FORMAT has been transported.
NOTE: Entry DEPT.FORMATC has been transported.
```

---

## Example 5: Exporting Entries Based on Modification Date

### Procedure features:

PROC CPORT statement options:

AFTER=  
FILE=

---

This example shows how to use PROC CPORT to transport only the catalog entries with modification dates equal to or later than the date you specify in the AFTER= option.

## Program

The LIBNAME and FILENAME statements assign the libref for the source library and the fileref for the transport file, respectively.

```
libname source 'SAS-data-library';
filename tranfile 'transport-file'
                host-option(s)-for-file-characteristics;
```

AFTER= specifies that only catalog entries with modification dates on or after September 9, 1996, should be written to the transport file.

```
proc cport catalog=source.finance file=tranfile
          after='09sep1996'd;
run;
```

## SAS Log

PROC CPORT writes messages to the SAS log to inform you that it began the export process for all of the entries in the specified catalog. However, PROC CPORT wrote only the entries LOAN.FRAME and LOAN.HELP in the FINANCE catalog to the transport file because only those two entries had a modification date equal to or later than September 9, 1996. That is, of all the entries in the specified catalog, only two met the requirement of the AFTER= option.

```
NOTE: Proc CPORT begins to transport catalog SOURCE.FINANCE
NOTE: The catalog has 5 entries and its maximum logical record length is 866.
NOTE: Entry LOAN.FRAME has been transported.
NOTE: Entry LOAN.HELP has been transported.
```





The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® *Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1729 pp.

**SAS® Procedures Guide, Version 8**

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

ISBN 1-58025-482-9

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**U.S. Government Restricted Rights Notice.** Use, duplication, or disclosure of the software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions 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, October 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.

IBM® and DB2® are registered trademarks or trademarks of International Business Machines Corporation. ORACLE® is a registered trademark of Oracle Corporation. ® 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.