



CHAPTER

11

The COPY Procedure

Overview	269
Procedure Syntax	269
Concepts	270
<i>Transporting SAS Data Sets between Hosts</i>	270
Example	270
<i>Example 1: Copying SAS Data Sets between Hosts</i>	270

Overview

The COPY procedure copies one or more SAS files from a SAS data library. Generally, the COPY procedure functions the same as the COPY statement in the DATASETS procedure. The two differences are as follows:

- The IN= argument is required with PROC COPY. In the COPY statement, IN= is optional. If omitted, the default value is the libref of the procedure input library.
- PROC DATASETS cannot work with libraries that allow only sequential data access.

The complete documentation for the COPY statement and the COPY procedure is “COPY Statement” on page 349 in Chapter 14, “The DATASETS Procedure,” on page 329

Procedure Syntax

Reminder: You can use global statements with this procedure. See Chapter 2, “Fundamental Concepts for Using Base SAS Procedures,” for a list.

```

PROC COPY OUT=libref-1 IN=libref-2
  <CLONE|NOCLONE>
  <MEMTYPE=(mtype(s))>
  <MOVE <ALTER=alter-password>>
  <INDEX=YES|NO>
  <CONSTRAINT=YES|NO>
  <FORCE>;
EXCLUDE SAS-file(s) </ MEMTYPE=mtype>;
SELECT SAS-file(s) </ <MEMTYPE=mtype>
  <ALTER=alter-password>>;

```

Concepts

Transporting SAS Data Sets between Hosts

The COPY procedure, along with the XPORT engine, can create and read transport files that can be moved from one host to another. PROC COPY can create transport files only with SAS data sets, not with catalogs or other types of SAS files.

Transporting is a three-step process:

- 1 Use PROC COPY to copy one or more SAS data sets to a file that is created with the transport (XPORT) engine. This file is referred to as a *transport file* and is always a sequential file.
- 2 After the file is created, you can move it to another operating environment via communications software, such as FTP, or tape. If you use communications software, be sure to move the file in binary format to avoid any type of conversion. If you are moving the file to a mainframe, the file must have certain attributes. Consult the SAS documentation for your operating environment and the SAS Technical Support Web page for more information.
- 3 After you have successfully moved the file to the receiving host, use PROC COPY to copy the data sets from the transport file to a SAS data library.

For an example, see Example 1 on page 270.

For details, see the SAS Technical Support Web page. In addition, see "An Overview of Transporting SAS between Hosts" in the SAS technical journal, *Observations* (3rd Qtr 1992), for a comprehensive overview of the transporting process and some problem-solving tips.

The CPORT and CIMPORT procedures also provide a way to transport SAS files. For information, see Chapter 8, "The CIMPORT Procedure," on page 211 and Chapter 13, "The CPORT Procedure," on page 313.

Example

Example 1: Copying SAS Data Sets between Hosts

Features:

PROC COPY statement options:

IN=
MEMTYPE=
OUT=

Other features: XPORT engine

This example illustrates how to create a transport file on a host and read it on another host.

In order for this example to work correctly, the transport file must have certain characteristics, as described in the SAS documentation for your operating environment. In addition, the transport file must be moved to the receiving operating system in binary format.

Program

Assign a libref, such as SOURCE, to the SAS data library that contains the SAS data set you want to transport. Also, assign a libref to the transport file. Use the XPORT keyword to specify that the library XPTOUT is a transport file.

```
libname source 'SAS-data-library-on-sending-host';
libname xptout xport 'filename-on-sending-host';
```

Use PROC COPY to copy the SAS data sets from the IN= library to the transport file. MEMTYPE=DATA specifies that only SAS data sets are copied. SELECT selects the data sets you want to copy.

```
proc copy in=source out=xptout memtype=data;
  select bonus budget salary;
run;
```

SAS Log

SAS Log on Sending Host

```
1  libname source 'SAS-data-library-on-sending-host ';
NOTE: Libref SOURCE was successfully assigned as follows:
      Engine:          V7
      Physical Name:  SAS-data-library-on-sending-host
2  libname xptout xport 'filename-on-sending-host';
NOTE: Libref XPTOUT was successfully assigned as follows:
      Engine:          XPORT
      Physical Name:  filename-on-sending-host
3  proc copy in=source out=xptout memtype=data;
4  select bonus budget salary;
5  run;

NOTE: Copying SOURCE.BONUS to XPTOUT.BONUS (memtype=DATA).
NOTE: The data set XPTOUT.BONUS has 1 observations and 3 variables.
NOTE: Copying SOURCE.BUDGET to XPTOUT.BUDGET (memtype=DATA).
NOTE: The data set XPTOUT.BUDGET has 1 observations and 3 variables.
NOTE: Copying SOURCE.SALARY to XPTOUT.SALARY (memtype=DATA).
NOTE: The data set XPTOUT.SALARY has 1 observations and 3 variables.
```

The XPORT engine in the LIBNAME statement enables the procedure to read the data from the transport file.

```
libname insource xport 'filename-on-receiving-host';
```

After you copy the files (for example, using FTP in binary mode to the Windows NT host), use PROC COPY to copy the SAS data sets to the WORK data library on the receiving host.

```
proc copy in=insource out=work;
run;
```

SAS Log on Receiving Host

```
1  libname insource xport 'filename-on-receiving-host';
NOTE: Libref INSOURCE was successfully assigned as follows:
      Engine:          XPORT
      Physical Name:  filename-on-receiving-host
2  proc copy in=insource out=work;
3  run;
NOTE: Input library INSOURCE is sequential.
NOTE: Copying INSOURCE.BUDGET to WORK.BUDGET (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines.
      System Option for BUFSIZE was used.
NOTE: The data set WORK.BUDGET has 1 observations and 3 variables.
NOTE: Copying INSOURCE.BONUS to WORK.BONUS (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines.
      System Option for BUFSIZE was used.
NOTE: The data set WORK.BONUS has 1 observations and 3 variables.
NOTE: Copying INSOURCE.SALARY to WORK.SALARY (memtype=DATA).
NOTE: BUFSIZE is not cloned when copying across different engines.
      System Option for BUFSIZE was used.
NOTE: The data set WORK.SALARY has 1 observations and 3 variables.
```

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.