Tips and Techniques For Moving Between Operating Environments

J. Meimei Ma, Quintiles, North Carolina, U.S.A.
Sandra Schlotzhauer, Schlotzhauer Consulting, North Carolina, U.S.A.

ABSTRACT

This tutorial introduces basic techniques pertinent to moving data files, catalogs, and/or applications between sites or operating environments. Our focus for files is on situations when you need to move data from one database format to another, not simply access files stored in another data management system using a "viewer" of some sort. To start, we define what we mean by "transfer" and "translate" in this paper. Basic procedures for transferring files or catalogs are discussed and illustrated with simple examples. The importance of documentation and quality control procedures is stressed. Output (listings, tables) or graphic files are not covered.

The primary audience for this presentation includes programmers, data managers, statisticians, and other people responsible for moving data and applications. With a new version of SAS software coming soon, everyone should learn about transfer issues related to upgrading. Only basic knowledge of the SAS system is assumed. No previous experience creating SAS applications is assumed. You should be familiar with the DATA step basics (formats, labels) and data management procedures in Base SAS for manipulating data sets or catalogs. The examples pertain to assorted computer environments. For additional technical details, a list of references is provided as a starting point for learning more.

TERMINOLOGY

The processes we present are special cases of *export* or *import* processes. Generally, exporting means moving entities from your standard computing environment to another one. Importing means bringing in entities to your computer system from a different *host environment*, which encompasses the hardware platform, operating system, and SAS software release. For simplicity, in this paper we restrict the definition of the terms "transfer" and "translate," as well as "transmit."

Transfer

For the following discussion, a *transfer* involves moving SAS entities between host environments. The entity could be a data set, a collection of data sets, a format catalog, or an application catalog.

Transport File Format

The *transport file format* has been around since Version 5 of the SAS System, but was used more often after Release 6.06/6.07 as sharing between computing environments increased in the 1980's. For SAS data sets and catalogs, using transport files simplifies transfers between different operating systems or hardware platforms by providing a work-around for binary incompatibilities that exist because different computers store numbers differently. A few changes have occurred since the first implementation, but the fundamental concept remains that a SAS transport file is a collection of one or more SAS data sets or catalog entries stored in transport format and stacked sequentially into a single file officially called a SAS transport library.

Transmit

Once you have created a transport file it must be physically moved to the new location or computer. If this move is accomplished using an electronic method that does not involve physical media such as a disk, then we would say that you will *transmit* the file.

Translate

Although *translate* can be interpreted in a variety of ways, we discuss only the process required to convert an entire data file—in particular a SAS data set—to or from a file format used by another data system. Using the INPUT statement in a Data Step is the classic method for moving data from one format to another, but we do not address this technique here. We are thinking of translating from commonly found files such as those created by Microsoft Excel, Microsoft Access, Borland dBase, or Oracle software.

Quality Process

You need to have methods for keeping organized when handling data files or the components of an application. A *quality process* is simply a set of predefined activities and operating procedures designed to accomplish a specific programming or data management objective that includes a systematic approach to checking that quality standards are met. Ideally, such a process is described in writing by a concise description or flowchart of all major steps. No matter how simple the transfer or translation process, quality control (QC) checks should be built in.

FILE TRANSFER PROCESSES

File transfers are generally simpler than catalog transfers, which we discuss later. Beginning with Version 5, most data transfers are accomplished by creating SAS transport files as an intermediate step. Increasingly, transport files can be read not only by the SAS system but also other software such as JMP or DBMS/COPY.

The most common situations that require transfers are:

- changing host environments, same location
- sharing work with other locations
- moving between companies or research groups
- upgrading to a later version of SAS software
- sharing with people stuck using an older release

Fischell et. al. noted in several papers (SUGI papers from 1990, 1992, 1993, 1994) that using SAS/CONNECT is an elegant and easy method for transferring files but it is not a realistic option in many situations. Obviously, if SAS/CONNECT is not licensed or the two computers are not on the same network then you cannot use this method. Using a batch process with transport files can be more efficient when you need to transfer a number of files or must transfer the same data sets more than a couple times.

Standard Transfer Process

The standard process for transfers is straightforward. The steps are as follows. See Appendix for flowchart.

1) Export: create transport file with source documentation
2) QC: test export
3) Move: transmit, or send physical media, with documentation
4) Import: create SAS data set(s) from transport file
5) QC: check new data sets using source documentation

See Examples 1 and 2 for code to export and import files. At a minimum, source documentation includes output from PROC CONTENTS and the transport file name. Send along an appropriate version of all stored formats to make the data more useful. Source documentation can be sent as printed pages and/or as files (CONTENTS output, FORMAT code or FMTLIB output or catalog entries).

Establishing a naming convention for transport files is helpful. In the examples, we use the extension ".XPT" as the last level of the file name regardless of operating system. You may also want to think about guidelines that relate the source data set or data library name to the transport file.

When you transmit a transport file, take care to maintain the proper file attributes. Always choose Binary or Image to transmit all bytes without modification. Generally, the file attributes must be Fixed, 80-byte record length, and BLKSIZE=8000. See Example 1 and the references from 1992 and 1994 for platform specific examples.

When sending physical media, you have a variety of choices. Magnetic media such as floppy disks, tape cartridges, or 9-track tapes are giving way to newer technologies such as high density disk cartridges (such as Zip™ or Jaz®), optical disks, or CDs. To save space, consider using compression software such as PKZIP or WinZip (PC, UNIX). Whatever you choose, be sure that the recipient can handle the media and uncompress the file.

The appropriate level of quality control depends on your situation. For a single transport file containing three data sets, testing the export by importing back to work data sets and running PROC CONTENTS or PROC COMPARE may be worthwhile. This method is less practical for large data sets or transport files containing more than ten data sets. However, any quality process can always start with a check of the number of converted observations noted in the SAS log.

Assuming the transfer is to another location or company, you can enhance your chances for a smooth transfer even further by having direct communication with the receiving programmer before sending anything. Basic questions worth early discussion are:

- What release of SAS software?
- How much documentation?
- What media or transmission method?
- Is a test transfer worthwhile?

In summary, to transfer SAS data set(s) successfully, you need to move a checked transport file and include enough documentation to allow QC after conversion.

Conversion Methods

There are two methods for converting transport files:

- Using the XPORT engine
- Using CPORT/CIMPORT procedures

Using the XPORT engine has two main advantages. The first is that the resulting transport file is more likely to be usable in any host environment or with an older release. The second is that you can select specific data sets from a data library in the COPY procedure. Note that you can use the XPORT engine in a LIBNAME referenced by a DATA step to create or read a transport file directly.

When using PROC CPORT to create a transport file, you must use a compatible version of PROC CIMPORT to convert back. However, an advantage of using CPORT/CIMPORT for data sets is that the resulting files are compressed and can be significantly smaller than what the XPORT engine would produce for the same data set. Only blanks and zeros are compressed. For example, for a small adverse events data set with 29 variables (record length 336) and 144 observations can have the following sizes with Release 6.12 on a PC.

File Type	Size (KB)
Uncompressed, .SD2	61 KB
Compressed, .SD2	25 KB
XPORT engine, transport	52 KB
CPORT, transport	27 KB

File Transfer Cautions

Although the transport file format simplifies transfers between hardware platforms, you must still be on the lookout for potential conversion problems. You should use the default maximum LENGTH=8 when using floating point numbers to avoid loss of precision. By default, both conversion methods increase LENGTHs by 1 byte if a LENGTH is less than 8. You can suppress this feature by specifying EXTENDSN=NO in CIMPORT statement (see Example 2). Another caution is that although changing from ASCII to EDBDIC should be handled automatically, check the transfer carefully.

Examples for Data Sets

Example 1: Using XPORT Engine and COPY

The code in this example produces a transport file on an unlabelled 9-track tape containing two of the ten data sets in the MVS data library. This tape could then be moved to another location that had a VMS machine assuming an appropriate tape drive was available.

```
//XPORT JOB ...
// EXEC SAS
//OUTAPE DD DSN=filedesc.V612.XPT,UNIT=TAPE,
//    LABEL=(1,NL),VOL=SER=vvvvvv,DISP=(OLD,KEEP),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
//SASD DD DSN=projname.filedesc.SASLIB,DISP=SHR
//SYSIN DD *
LIBNAME outape XPORT;
TITLE "SEND datname1, datname2 FOR projname";
PROC COPY IN=SASD OUT=OUTAPE;
        SELECT datname1 datname2;
```

Note that the tape DSN includes a level to document that Release 6.12 was used to create the transport file. You could add PROC CONTENTS for the two selected data sets to this program. Then the SAS log and output would provide basic source documentation.

Example 2: Using PROC CIMPORT

Suppose you are working under a Microsoft Windows OS and have saved a transport file created using the CPORT procedure in the directory MYPROJ/DATALIB with the file name CT100.XPT. Assuming you are using the same version of SAS software as the sender, then the code to convert back to SAS data sets stored in an empty directory MYPROJ/SASLIB is shown below.

```
FILENAME xptfile "c:\myproj\datalib\ct100.xpt";
LIBNAME newdir "c:\myproj\saslib";

TITLE "Convert From CT100.XPT";
PROC CIMPORT FILE=xptfile LIBRARY=newdir;
RUN;
PROC CONTENTS DATA=newdir._all_;
```

Example 3: CPORT Backwards Compatibility

Release 6.11 brought changes to PROC CPORT. The procedure was changed so that you could not transport data sets that were read-protected. To transfer a protected data set, you must first remove the password. Remember to document the fact that a password existed if the protection should be restored after the transfer. Another addition was the V608 option to provide the first step for compatibility with older releases still in common use.

In Release 6.12, changes were made in transport file format for data created by PROC CPORT. If you need to create a transport file compatible with Release 6.08, 6.09, 6.10, or 6.11 you can use the V608 option or the SORTINFO option. The main improvement for data transport files is that they retain information about the sort status of the tramsferred data sets. The sort information includes the names of the sort variables, an indicator of whether PROC SORT or the SORTEDBY= option was used, an indicator of whether the NODUPKEY or NODUPREC options of PROC SORT were used, the character set used, and the collating sequence used in the sort. Assuming that appropriate LIBNAME and FILENAME are defined, the syntax is

```
PROC CPORT LIBRARY=olddir FILE=xptfile  V608;
```
or
```
PROC CPORT LIBRARY=olddir FILE=xptfile
            SORTINFO=NO;
```

```
1)  XPORT engine:      HEADER RECORD*******LIBRARY HEADER RECORD!!!!!!!000000000000000
2)  PROC CPORT:        **COMPRESSED** **COMPRESSED** **COMPRESSED** **COMPRESSED**
```

Figure 1. Header Line in Transport Files

Example 4: XPORT or CPORT?

What can you do when you receive a transport file without documentation about what type it is? This program will let you know whether to use the XPORT engine or CIMPORT to convert the file. Depending on your host environment, you may also look for the pertinent header information using an editor that displays ASCII text for variable length records.

```
FILENAME tranfile 'file-specification';
DATA _NULL_;
   INFILE tranfile OBS=5;
   INPUT theline $ASCII80.;
   PUT theline;
RUN:
```

Run the program and compare the output with the two possibilities show in Figure 1: Header #1 appears when the XPORT engine is used; Header #2 appears when PROC CPORT is used to create a transport file (compressed by default). If you try to use CIMPORT on an XPORT file, you may get the unhelpful error message "CIMPORT is not able to read Version 5 transport data sets."

CATALOG TRANSFER PROCESSES

Transferring catalogs is often needed when changing versions or releases of SAS software, or to move an application from one operating system to another. We consider format catalogs and application catalogs (such as from SAS/AF) separately because the likely problems are differenet. The overall quality process is essentially the same as for files:

1) Export: create transport file with source doc.
2) QC: test export
3) Move: transmit, or send physical media, with doc.
4) Import: create SAS catalog from transport file
5) QC: check new catalog using source documentation

Format Catalog Transfers

When transferring SAS format catalogs, you only need to worry about formats that you have defined using the FORMAT procedure. The SAS software standard formats, such as $QUOTEw., DATE9., DATEAMPMw.,

COMMAw.d, PERCENTw., NUMXw.d, Zw.d, DOLLARw.d, and SSNw. are automatically available on every operating system.

Format catalogs can be transferred using two approaches. The simplest approach is to use PROC CPORT and CIMPORT to transfer the catalog. A variation on this approach is to use PROC CATALOG first to create a second catalog that contains only a few formats to be transferred as shown in Example 6. Another approach is to use the CNTLOUT and CNTLIN options of the FORMAT procedure (see Example 8).

Example 5: CPORT/CIMPORT for Formats

The code example below creates a transport file for the FMTS format catalog. The transport file is placed on a diskette for the move. The special LIBNAME of LIBRARY is used for the format catalog, which is the reference used automatically by SAS software.

```
LIBNAME library 'c:\sasdata';
FILENAME fmtrans 'a:\fmtrans.xpt';

PROC CPORT CATALOG=library.fmts
     FILE=fmtrans;
RUN;
```

To import the file, simply reverse the transfer process as shown in the sample code below. In this example, the file is transferred to the D drive of another PC.

```
LIBNAME library 'd:\sasdata';
FILENAME fmtrans 'a:\fmtrans.xpt';

PROC CIMPORT CATALOG=library.fmts
     FILE=fmtrans;
RUN;
```

Always read the SAS log carefully when transferring any type of file. For format catalogs, the log shows messages for each format processed. Reviewing and saving the log is a useful QC check on your work. Confirm that the format catalog has the same number of formats on each side of the transfer process. You can include the log as part of the source documentation sent with the transport file. The PRINTTO procedure or DM commands can save the log to a file automatically.

4

Example 6: Moving A Few Formats

Suppose you want to transfer only a few formats from a large format catalog stored on another PC. And, you are running short of space on your hard drive so you do not want to transfer the huge catalog and then delete formats. A solution is to use PROC CATALOG to copy the needed formats to a second catalog. Then, use PROC CPORT and CIMPORT as in Example 5 to transfer the smaller catalog to your PC.

```
PROC CATALOG C=fmts.formats;
   COPYOUT=fmts.small;
       SELECT agegrp bmiclas
       / ET=FORMAT;
RUN;
QUIT;

PROC CATALOG C=fmts.formats;
   COPYOUT=fmts.small;
       SELECT gender sevcat
       / ET=FORMATC;
RUN;
QUIT;
```

The code for this example uses two copy steps, one for numeric and one for character formats.

Example 7: Upgrading From Release 6.08

There are situations where format catalogs from a prior version need to be used with the current version of SAS software. The following example shows how to use the C16PORT procedure on a PC to convert a Release 6.08 format catalog to a Release 6.11 catalog. Remember that catalogs from Release 6.03/6.04 had the extension "SCT" as opposed to "SC2." PROC C16PORT will not work on those catalogs. In this example, the catalog is not transferred across operating systems.

```
/*-- Libnames. Use V608 engine for formats cat --*/
LIBNAME newstuf 'c:\sasdata';
LIBNAME flopy V608 'a:';

/*-- Convert catalog to transport file       --*/
PROC C16PORT C=flopy.formats
               FILE='c:\temp\fmts.xpt';
RUN;

/*-- Import file to catalog              --*/
PROC CIMPORT
       INFILE='c:\temp\fmts.xpt'
       CAT=newstuf.formats;
RUN;
```

The older version of the catalog, named FORMATS.SC2, is stored on a floppy diskette and is from Release 6.08. The C16PORT procedure converts the catalog to a transport file that can be read by newer releases. The CIMPORT procedure then imports the transport file to a catalog that is usable with the current SAS software release.

Example 8: Using CNTLOUT and CNTLIN

When changing versions of SAS software, not just releases, you may want to use the CNTLOUT/CNTLIN options of PROC FORMAT instead of transferring the catalog directly. The process is an extension of the file transfer process.

1) Create the CNTLOUT data set and source doc.
2) Export: create transport file of CNTLOUT data set
3) QC: test export
4) Move: transmit, or send physical media, with doc.
5) Import: create CNTLIN data set from transport file
6) Create format catalog from CNTLIN data set
7) QC: check new formats using source documentation

Application Catalog Transfers

Transferring application catalogs is handled using the CPORT and CIMPORT procedures, as was discussed earlier with format catalogs. The code steps are the same as in the format case.

The larger issues with application catalogs have to do not with the transfer process itself but with the contents of the application catalogs and with transfers between releases of SAS software. Designing a quality transfer process for an application is much more complicated than for data files.

First, consider the issues in transferring across operating systems. Do you know if the application was designed to be able to run in the target host environment? As an example, if the application creates 8-character-named SAS data sets that are also indexed, the application will not run properly under CMS. Does the application take advantage of host-specific features for screen displays, use host-specific code for libnames and filenames, or take advantage of SAS/AF or SCL features only available for a given release of the SAS system? The source of answers to these questions is the development team for the application, but finding them may not be easy. Programmers transfer to other projects or other companies, so the program documentation may be your only source of information. The only truly safe way to assure a simple transfer from one operating system to any other operating system is to use a "lowest common denominator"

approach, which means building an application that uses only features common to all operating systems. Since this approach often limits the functionality of an application on the "most commonly used" host environment, many applications do rely on host-specific or release-specific features.

Second, consider the issues in transporting applications across releases of SAS software. The potential problems may also apply when transferring across operating systems, since different operating systems often also use different releases. As an example, the V608 option of PROC CPORT allows you to transfer a Release 6.11 application to Release 6.08, 6.09, or 6.10. Although this option removes unsupported features, such as widgets made from classes available starting with Release 6.11, it is unlikely to automatically solve all issues with "back-porting" an application to a previous release.

Application Transfer Checklist

There is not a single approach to solving issues raised when transporting applications. The checklist below is a starting point. The SAS manuals and technical papers listed at the end of the paper contain additional information.

- What release of SAS was the application built for?
- Does the application use features new to that release?
- Is it known if the SCL code contains host-specific aspects (libnames, filenames, and so on)?
- Are you attempting to port to a prior release?
- Once ported, does the SAS log indicate problems with CPORT or CIMPORT?
- Does the SAS log indicate that the SCL code needs to be recompiled? Have you recompiled?

TRANSLATION PROCESSES

The quality process for translating a complete data file is similar to what you do for a transfer. You can use facilities built into the SAS System or other software such as JMP to translate to or from SAS data sets without writing code to read separate fields.

Suppose you want to use data stored in an Excel spreadsheet for an analysis. The process would be:

1) Setup: create source documentation
2) Move: transmit, or send, spreadsheet with doc.
3) QC: confirm or test move, backup original file
4) Import: create SAS data set from spreadsheet
5) QC: check data set using source documentation

Source documentation should include descriptive statistics whenever possible. Beyond knowing the number of records to expect, knowing frequencies of primary categorical variables or the mean for a few variables is useful for verifying that the transfer was successful.

The variety of techniques available for translating data files is beyond the scope of this paper. The simplest method for one PC file is to use the Import/Export Wizard in Release 6.12. You can check out the facility by looking under the File menu. Note that you cannot use this wizard to create or convert transport files. If you must translate many files or repeat the process then you should learn about access and view descriptors created by PROC ACCESS. For file formats not covered by SAS Institute products, consider DBMS/COPY (designed by Conceptual Software) for file translation of spreadsheet or database files. In the PC environment, making use of ODBC, OLE, or DDE is another way to go for applications that involve other file types.

JMP TRANSFER ISSUES

For those not familiar with JMP, it is another SAS Institute product that focuses on interactive statistical and graphical exploration of data. JMP is available for Macintosh and PCs; for this paper, we focus on PC use with Version 3.2.1 of JMP. JMP has the unusual feature that it can read transport files and SAS data sets directly (as well as other types of files not discussed here). Each approach has benefits and potential issues as summarized in Table 1.

When you are thinking about which option will be better for your situation, keep the following in mind.

- For JMP to read a SAS transport file, the file must be created using the XPORT engine.
- For JMP to read a SAS data set, the data set must be from a PC version of SAS software. If you start on another hardware platform, you need to create a transport file, transfer to the PC, import the transport file on the PC, and save the data set on the PC.
- There is a known problem reading compressed SAS data sets that have missing values for variables. The problem is that other values are substituted for missing values and this substitution does not occur in a predictable way. The solution is to be sure the SAS data sets were not created with the COMPRESS option.
- For situations where you want to transfer from JMP to the SAS System, you can create a transport file in JMP and then import it using SAS software.

Feature	SAS Data Sets	SAS Transport Files
Read multiple data sets at once	No	Yes
Use SAS labels as column headings (31 character maximum)	Yes	No
Keep SAS variable names	No	Yes
Use SAS labels in Column Info notes (all 40 characters from SAS labels)	Yes	Yes
Read from non-PC operating systems	No	Yes
Save JMP data table as …	No	Yes
Require formatting of dates in SAS data sets	Yes	Yes

Table 1. Moving From The SAS System to JMP 3.2.1 for Microsoft Windows

Regardless of the approach, there are two issues to consider on the SAS System side to facilitate translation in JMP: labels and dates.

Because JMP automatically makes use of labels during a translation, you should have labels in the source SAS data set. An idiosyncrasy is that SAS software allows 40 characters for variable labels and JMP allows 31 characters for column headings. When reading SAS data sets directly, only the first 31 characters appear in the column heading. The remaining characters can be read from the Column Info notes for the column.

The other important issue involves dates. Because the dates in SAS data sets and JMP tables are from different worlds, it is very important to format the dates in the source data sets. SAS software stores dates internally as the number of days since January 1, 1960. JMP stores dates internally as the number of seconds since January 1, 1904. If the date variables have date formats permanently associated (as seen in CONTENTS), then JMP reads and formats date columns correctly. Otherwise, JMP reads the dates into simple numeric columns.

In many situations, working with SAS transport files provides the approach with the fewest transitions. This is especially true in most corporate situations where SAS software is used in a non-PC environment. Regardless of which approach you use, be sure to perform QC checks on the transfer using JMP summaries and graphs. This approach is a more meaningful method to verify the transfer result because JMP does not have a programming language or log that you can use to check aspects of reading or importing the data.

Finally, JMP provides the Table Info feature where you can enter notes on the data table. Use this area to identify

where the SAS data set originated and basic data documentation. Output from PROC CONTENTS is a good place to start. For instance, answer the following questions.

- What is the datetime stamp of the original file?
- Is the data set sorted? How?
- Where is the SAS data set or transport file?
- Is the data likely to be updated? How often?

By providing this type of detail in the Table Info notes, you can avoid potential miscommunications with users reviewing the data with JMP.

CONCLUSION

As sharing files and applications becomes commonplace, we hope this tutorial gives you a taste of how to use a quality process to successfully move between computing environments. Consult the references when you need more platform-specific examples or ideas of how to make transfers easier. Another good source of information is the on-line information accessed through the SAS Institute's Web site at www.sas.com.

Keys to a Quality Transfer Process

- Good source documentation
- Planned QC checks
- Current technical information

USEFUL SAS PROCEDURES

The following Base SAS procedures are useful for transferring files or catalogs, or creating useful source documentation.

CATALOG
COMPARE
CONTENTS
COPY
CPORT/CIMPORT
DATASETS
FORMAT
FREQ
MEANS
PRINT
PRINTTO
UNIVARIATE

RECOMMENDED READING

The manuals listed here were written and published by SAS Institute, Inc., Cary, NC, USA. The conference proceedings were also published by SAS Institute.

Books By Users

Calvert, William and Ma, J. Meimei, *Concepts and Case Studies in Data Management,* Cary, NC: SAS Institute Inc., 1996. 150 pages. Introduction to QC concepts and research data management processes, with suggestions about detecting data corruption during transfers.

Delwiche, Lora D. and Slaughter, Susan J., *The Little SAS Book: A Primer,* Cary, NC: SAS Institute Inc., 1995. 228 pages. Examples of transfers in Chapter 3.

SAS Institute Technical Sources

In addition to these resources, we strongly recommend that you consult the "companion" manuals applicable to your hardware platforms because they contain technical details related to the transport engine, transport file attributes, and translation options.

SAS Technical Report P-195: Transporting SAS Files between Host Systems, 1989. Covers Version 5 to Version 6 (6.03, 6.06/6.07) for all operating systems.

SAS Companion for the Microsoft Windows Environment, Version 6, First Edition, 1993. Written for Release 6.08 under Microsoft Windows 3.1, but still a relevantresource.

Getting Started with SAS/ACCESS Software, Version 6, First Edition, 1994.

SAS/ACCESS Software for PC File Formats: Reference, Version 6, First Edition, 1995.

Microsoft Windows Environment: Changes and Enhancements, Release 6.10, 1995. Documentation of PROC C16PORT (not in Release 6.12 on-line help).

SAS Software: Changes and Enhancements, Release 6.11, 1995. Changes in CIMPORT, CPORT including V608 option and other information about potential problems transporting applications to previous releases, pp. 35-38 (text available as on-line help in Release 6.12).

Release 6.12 on-line documentation for PROC CPORT can be found using Find on "transports" to locate the entry for "CPORT: Index." (Does not include ENTENDSN= option.)

The complete record layout of a transport file is at http://www.sas.com/techsup/download/technote/ts140.text

Papers Related to Data Transfers

Barrios, A. (1997), "SAS Changes and Enhancements for Version 6.11 and Version 6.12," *Proceedings of the Fourth Annual SouthEast SAS Users Group Conference*, 220-223. Examples of FTP option in the filename statement to read and create files on remote hosts. Also has MVS code example.

Coen, M.M. and Fiola, M.J. (1995), "Experiences Using SAS Transport Libraries," *Proceedings of the Twentieth Annual SAS Users Group Intl. Conference*, 20, 537-541. Using FTP to move between Microsoft Windows 3.1, OpenVMS, VM/CMS under Release 6.08.

Fischell, T. and Menache, M.G. (1992), "Moving SAS Transport Files Across Different Hardware Platforms: An Advanced Tutorial," *Proceedings of the Seventeenth Annual SAS Users Group Intl. Conference*, 17, 208-215. Examples for a number of host environments.

Hardy, K., Barrios, A., and Muller, S. (1996), "You Want Me to Move How Many Thousand Files from MVS to UNIX?", *Proceedings of the Twenty-first Annual SAS Users Group Intl. Conference*, 21, 1611-1621. Case study in academic setting.

Michel, D. (1997), "Portable Data Processing," *Proceedings of the Tenth Annual NorthEast SAS Users Group Conference*, 410-416. Procedures for moving clinical trials data with appropriate documentation.

Selby, Y. (1992), "Transporting Files Between Host Operating Systems," *Proceedings of the Seventeenth Annual SAS Users Group Intl. Conference*, 17, 630-634. Transporting catalogs from Release 5.18 and 6.03/6.04 to later Version 6 releases.

Shinn, D. and Driggs, D., "An Overview of Transporting SAS Files between Hosts," *Observations,* Volume 1, Number 4, 33-40. Cary, NC: SAS Institute Inc., 1992.

Upson, R.H., Barnes G.M., and Fischell, T. (1994), "From the Ridiculous to the Sublime: Getting Files from There to Here," *Proceedings of the Nineteenth Annual SAS Users Group Intl. Conference*, 19, 419-427. Overview of a number of methods for electronic transmission such as FTP.

Wright, S. (1997), "Converting an Old MVS Mainframe Project to Run on Open-VMS: A Guidebook to Success," *Proceedings of the Twenty-Second Annual SAS Users Group Intl. Conference*, 22, 1442-1445. Case study with practicaladvice.

Other Papers of Interest

Franklin, D. (1997), "You Want This New Application to Run on Our VAX, PC and SUN Computers???!!! (Or Cross-Platform Application Development Using SAS Software," *Proceedings of the Twenty-Second Annual SAS Users Group Intl. Conference*, 22, 53-57.

Kolb, D. (1995), "Integrating the SAS System for Personal Computers into Your Enterprise," *Proceedings of the Twentieth Annual SAS Users Group Intl. Conference*, 20, 1306-1327. Comprehensive overview of all SAS System methods for accessing different data types or remote files.

LeBouton, K.J (1997), "The Realities of Downsizing: Moving a SAS Application from MVS to UNIX," *Proceedings of the Twenty-Second Annual SAS Users Group Intl. Conference*, 22, 658-663. Case study overview with many references and sample program code.

Mehler, G. (1997), "Integrating Windows Clients and the SAS System into the Enterprise," *Proceedings of the Twenty-Second Annual SAS Users Group Intl. Conference*, 22, 1396-1404. SAS Institute overview of a variety of data sharing options and products.

Wilson, S.A. (1996), "Using SAS Views for Data Source Visualization," *Proceedings of the Twenty-first Annual SAS Users Group Intl. Conference*, 21, 484-493. Solid introduction to all types of views.

CONTACT INFORMATION

J. Meimei Ma
Quintiles
P. O. Box 13979
Research Triangle Park, NC 27709
    Voice: 919-941-7136
    Fax: 919-941-0972
    Internet: mma@quintiles.com

SandraSchlotzhauer
Schlotzhauer Consulting
400 Gibbon Drive
Chapel Hill, NC 27516
    Voice: 919-9336341
    Fax: 919-933-5569
    Internet:sdschlotz@earthlink.net

APPENDIX: Transfer Process Flowchart

Source SAS data set

Export: create transport file and documentation

Transport okay? — NO

YES

Source transport file

Source doc.

Move to new location or computer

New transport file

Source doc.

If import not okay then repeat steps as needed:

1) Import
or
2) Move, Import
or
3) Export, Move, Import

Import: create SAS data set

Import okay?

NO

YES

New SAS data set