



CHAPTER

4

Transferring a Transport File or a CEDA File

<i>File Transfer</i>	31
<i>Transport File Attributes</i>	31
<i>Using the FILENAME Statement or the FTP Utility to Specify File Attributes</i>	32
<i>Using the FILENAME Statement to Specify File Attributes for All Hosts</i>	32
<i>Using FTP to Transfer a File across the Network</i>	33
<i>Transferring a Transport File by Using a Magnetic Medium</i>	33
<i>Example of Using FTP to Transfer Transport Files</i>	34
<i>Using FTP to Transfer a File That Was Created with CEDA Features</i>	35

File Transfer

Transfer is the process of conveying a file between hosts across a network. Various third-party products are available for performing this operation. In this book, we use FTP (File Transfer Protocol) to illustrate the transfer operation.

You perform a transfer operation either by:

pushing a file	from the source host, use the FTP put command to copy a file from the source host to the target host.
pulling a file	from the target host, use the FTP get command to copy a file from the source host to the target host.

Your ability to push a file from the source to the target host may depend on your permission to write to the target host. For complete details, see your network documentation.

Transport File Attributes

File attributes describe the organization and format of the data in the transport file that is transferred to a target host. A transport file must have these attribute values:

Note: A file that is created with CEDA features does not need these attributes.

△

Logical record length (LRECL)	80
Block size (BLKSIZE)	8000 bytes
Record format (RECFM)	Fixed block

Note: In some cases, a Block Size value of less than 8000 may be more efficient for your storage device. The Block Size value must be an exact multiple of the Record Length value. Δ

You must set file attributes to move a transport file from an OS/390 host to any other target host. Although not required, file attributes may be set for all other source hosts. How file attributes are declared depends on the source host at which the transport file is created and on the transfer method used. For details about setting file attributes for these hosts, see the appropriate chapter in Part 5.

Furthermore, through the communications software protocol, you must additionally specify file attributes for files on hosts that require them. For example, if you transfer a transport file from a UNIX host to an OS/390 host, you must specify file attributes through the communications software.

Besides setting file attributes for those hosts that require it, ensure that your communications software does not alter the default file attribute settings for any host.

Alternatively, in order to transfer a transport file from a source host to tape and then from tape to disk at the target host, you use host-specific commands that define the input and output devices for the hosts involved in the transfer. See the appropriate chapter in Part 5 for details about tape commands.

After the transport file is created, it must then be transferred to the target host either across the network or by means of a mountable magnetic medium such as a floppy disk or a tape.

The incorrect setting of file attributes is a major source of transport problems for which error messages are reported.

Using the FILENAME Statement or the FTP Utility to Specify File Attributes

You can specify file attributes through either the FILENAME statement or the FTP utility.

Note: Releases 6.11 through Version 8 support the FILENAME statement with the FTP access method, which specifies file attributes for file transfer. Releases prior to 6.11 do not support the FILENAME statement with the FTP access method. Δ

Using the FILENAME Statement to Specify File Attributes for All Hosts

Here is an example of using the FILENAME statement with the FTP access method to specify file attributes and to transfer a transport file over the network to a target host:

```
filename tranfile ftp 'tport.dat' lrecl=80 blocksize=8000
  recfm=f cd='mydir' host='myhost.mycompany.com'
  user='myuser' pass='mypass'
  rcmd='site umask 022' recfm=s;
```

In this example, the FILENAME statement specifies the fileref TRANFILE, which points to the external file TPORT.DAT for transfer over the network. FTP options specify values for the record attributes: record length, block size, and record format.

Also, FTP options identify the location for the file transfer on the target host and the userid and password that permit access to the target host. Finally, the file mode creation mask on the target host and a binary transfer are specified. For complete details about the FILENAME statement with the FTP access method, see *SAS Language Reference: Dictionary* and the companion that is appropriate to your operating environment. For an example of using the FILENAME statement with the FTP access method, see “OS/390 TSO to UNIX File Transport” on page 146.

Using FTP to Transfer a File across the Network

FTP is a user interface to the File Transfer Protocol. FTP copies files across a network connection between the source host and a target host. FTP runs from the initiating host, which can be either the source host or the target host.

In order to transfer a file to a target host across a network, ensure that a binary (or image) format transfer is specified. This format guarantees a consistent file structure for any operating environment that runs SAS. For example, you must use the FTP BINARY command to declare binary format. For typical FTP command syntax, see “Example of Using FTP to Transfer Transport Files” on page 34.

Transferring a file in ASCII format places extra characters in the transport file on the target host. These characters are typically line feeds, carriage returns, end-of-record markers, and other characters that some operating environments use to define file characteristics.

Target hosts that run SAS expect a transport file to be formatted in a certain structure, without these characters. The introduction of these characters into a file causes corruption, which prevents the file from being successfully restored at the target host. Error messages usually warn of file corruption. For information about file corruption and how to recover from this condition, see Chapter 15, “Preventing and Fixing Problems,” on page 103.

Transferring a Transport File by Using a Magnetic Medium

When transporting a file by means of tape, always use an unlabeled tape. Although using a standard labeled tape is possible, it usually requires extra work to read the file at the target host.

Also, if the transport file exceeds the capacity of one tape, then problems may occur during the restoration process. Rather than using multi-volume tapes, you should divide the original library into two or more libraries and create a separate tape for each one. The original library can be rebuilt at the target host.

At the source host, use the LIBNAME statement to assign the transport file to a magnetic medium. Here are two examples:

```
UNIX                libname tranfile xport '/dev/tape';
Windows            libname tran xport 'a:\test';
```

Specification of the file path varies by host.

The method used to move the transport file to a physical tape varies by host. Here is a UNIX example:

```
dd if=tranfile of=/dev/tape1 bs=8000;
```

The UNIX `dd` command copies the specified input file to the specified output device. Block size is 8000.

At the target host, you must copy the transport file from tape to disk. Here is a UNIX example:

```
dd if=/dev/tape1 of=tranfile bs=8000;
```

At the target host, you use the LIBNAME statement to translate the transport file to native format, assigning the resulting translated file to a specific file location. Here is a UNIX example:

```
libname tranfile xport '/dev/tape1';
```

For host-dependent details about using tape devices, see the appropriate chapter in Part 5 and the companion documentation that is appropriate to your operating environment.

Example of Using FTP to Transfer Transport Files

In the following examples, TRANFILE specifies the name of the transport file that is transferred across the network. TARGET specifies the destination for the transport file on the target host.

Example Code 4.1 on page 34 shows the typical FTP commands used at the source host to put a transport file on the target host:

Example Code 4.1 Typical FTP PUT Commands

```
/* putting transport file on the target host    */
> open target-host
> binary
> put tranfile target-host-filename
> close
> quit
```

Example Code 4.2 on page 34 shows the typical FTP commands used at the target host to get a transport file from the source host:

Example Code 4.2 Typical FTP GET Commands

```
/* At the source host, getting transport file from */
/* the target host                                */
> open target-host
> binary
> get tranfile source-host-filename
> close
> quit
```

If you have access to a UNIX system, see the **ftp(1)** manual page for more details.

Note: In order to copy a file with the FTP **put** command to a remote host location, you must have write permission to the target location on the remote host. Because a local user's permission to put a file at a remote host location is uncertain, it is recommended that the remote user use the FTP **get** command to obtain the file from the local host instead. The local user must grant read and write permission to the file that the remote user accesses. \triangle

The following code shows an example of user JOE at the target host getting two transport files from an OpenVMS Alpha source host:

Example Code 4.3 Typical FTP Session

```
❶ hp> ftp myhost.mycompany.com
Connected to myhost.mycompany.com
```

```

220 myhost.mycompany.com MultiNet FTP Server Process V4.0(15)
    at Mon 28-Sep-98 12:59PM-EDT
Name (myhost.mycompany.com:): joe
331 User name (joe) ok. Password, please.
Password:
230 User JOE logged into DISK01:[JOE] at Mon 28-Sep-98
    12:59PM-EDT, job 27a34cef.
Remote system type is VMS.
❷ ftp> cd [.xpttest]
250 Connected to DISK01:[JOE.XPTTEST].
❸ ftp> binary 80
200 Type I ok.
❹ ftp> get xptds.dat xptds.dat
200 Port 14.83 at Host 10.26.2.45 accepted.
150 IMAGE retrieve of DISK01:[JOE.XPTTEST]XPTDS.DAT;1 started.
❺ 226 Transfer completed. 1360 (8) bytes transferred.
1360 bytes received in 0.02 seconds (87.59 Kbytes/s)
❻ ftp> get xptlib.dat xptlib.dat
200 Port 14.84 at Host 10.26.2.45 accepted.
150 IMAGE retrieve of DISK01:[JOE.XPTTEST]XPTLIB.DAT;1 started.
❼ 226 Transfer completed. 3120 (8) bytes transferred.
3120 bytes received in 0.04 seconds (85.81 Kbytes/s)
❽ ftp> quit

```

- 1 From an HP-UX host, the user invokes FTP to connect to the OpenVMS Alpha host MYHOST.MYCOMPANY.COM.
- 2 After a connection is established between the HP-UX source host and the OpenVMS Alpha target host, at the FTP prompt, the user JOE changes to the directory on the target host that contains transport file XPTTEST.
- 3 Transport file attributes BINARY 80 indicate that the OpenVMS transport file be transferred to the source host in BINARY format in 80-byte records.
- 4 The FTP command gets the transport file named XPTDS.DAT from the target host and copies it to a new file that has the same name, XPTDS.DAT, in the current directory.
- 5 Messages indicate that the transfer was successful and that the length of the transport file was 1360 bytes.
- 6 The FTP command gets another transport file named XPTLIB.DAT from the target host and copies it to a new file that has the same name, XPTLIB.DAT, in the current directory.
- 7 Messages indicate that the transfer was successful and that the length of the transport file was 3120 bytes.
- 8 The user quits the FTP session.

Using FTP to Transfer a File That Was Created with CEDA Features

You transfer a file that was created with CEDA features in the same way that you transfer a transport file. The only difference between the two is the filename. SAS appends a transport filename with an appropriate member type extension, such as .DAT for a data set. A file that was created with CEDA features is appended with an appropriate Version 8 filename extension; for example, .SAS7BDAT for a data set.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Moving and Accessing SAS Files across Operating Environments, Version 8*, Cary, NC: SAS Institute Inc., 1999. 186 pages.

Moving and Accessing SAS Files across Operating Environments, Version 8

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

ISBN 1-58025-480-2

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, by 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 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, 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[®], AIX[®], DB2[®], OS/2[®], OS/390[®], and System/390[®] are registered trademarks or trademarks of International Business Machines Corporation. ORACLE[®] is a registered trademark or 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.