CHAPTER 11

OpenVMS Operating Environment

Listing OpenVMS System File Attributes

To list the attributes of a file created on an OpenVMS VAX system, issue this command:

```
DIR/FULL transport-file
```

Typical output is:

```
Directory DISK01:[JOE.XPTTEST]

XPTLIB.DAT;1                  File ID: 31223,952,0)
Size:            7/8          Owner:  [DISK01,JOE]
Created:   30-SEP-1999 16:47:31.34
Revised:   30-SEP-1999 16:47:31.69 (1)
Expires:  <No backup recorded>
Effective: <None specified>
Recording: <None specified>
File organization: Sequential
Shelved state:      Online
File attributes:    Allocation: 8, Extend: 0,
   Global buffer count: 0   Version limit: 2
❶ Record format:Fixed length 80 byte records
❷ Record attributes:  None
RMS attributes:     None
Journaling enabled: None
```

```
File protection:    System:RWED, Owner:RWED,
   Group:RE, World:
Access Cntrl List:  None

Total of 1 file, 7/8 blocks.
$ dir/size xptlib.dat

Directory DISK01:[JOE.XPTTEST]

XPTLIB.DAT;1               7

Total of 1 file, 7 blocks.
```

1 The OpenVMS VAX RECORD FORMAT attribute indicates a fixed record type and an 80–byte record size. These values are required for a successful file transfer across the network. You specify block size by using FTP or the FTP option in the FILENAME statement.

An OpenVMS Alpha system RECORD FORMAT should indicate a record length of 512 bytes.

2 The RECORD ATTRIBUTES field could contain the value NONE.

CAUTION:

If this field contains CARRIAGE RETURN CARRIAGE CONTROL, file corruption results. To prevent corruption before you transfer the transport file, remove this value from the RECORD ATTRIBUTES field. An error message alerts you to this condition after you attempt to transfer the corrupted file. △

Specifying File Attributes for OpenVMS

You may specify transport file attributes by using FTP or FTP access method options in the FILENAME statement, whichever is applicable. For details about syntax for the FILENAME statement, see *SAS Companion for the OpenVMS Operating Environment*. For details about specifying file attributes, see “Using the FILENAME Statement to Specify File Attributes for All Hosts” on page 32.

Determining the SAS Release Used to Create a Member for OpenVMS

Table 11.1 on page 82 identifies the supported file types that are created on the OpenVMS system by member and SAS release:

Table 11.1 OpenVMS Filename Extensions by Member and SAS Release

Member Type	Version 6 Filename Extension	Version 7 Filename Extension
SAS	.SAS	.SAS
PROGRAM (DATA step)	.SASEB$PROGRAM	.sas7bpgm
DATA	.SASEB$DATA	.sas7bdat

Member Type	Version 6 Filename Extension	Version 7 Filename Extension
INDEX	.SASEB$INDEX	.sas7bndx
CATALOG	.SASEB$CATALOG	.sas7bcat
MDDB	.SASEB$MDDB	.sas7bmdb
PROC SQL view	.SASEB$VIEW	.sas7bvew

Furthermore, you can use the CONTENTS procedure to display information about the data, which identifies the member and the engine that was used to create it.

Here is an excerpt of typical PROC CONTENTS output:

```
The SAS System
    The CONTENTS Procedure
Data Set Name: TEST.RECORDS
Member Type:   DATA
Engine:        V8
```

This output reports that the data set TEST.RECORDS is a member of type DATA, and that it was created with the V8 engine.

Mounting a Tape Device on OpenVMS

In order to move a transport file from disk to tape at the source system and to move a transport file from tape to disk at the target system, issue the following DCL commands to assign the tape device before starting a SAS session:

Note: Use the INITIALIZE command *only* if you have a new tape. The INITIALIZE command destroys any files that already may be on the tape. △

```
$ DEFINE TRANFILE tape-name
$ ALLOCATE TRANFILE
$ INITIALIZE TRANFILE DUMMY
$ MOUNT/FOREIGN/BLOCKSIZE=8000 TRANFILE
```

Note: TRANFILE in the DCL commands is identical to the libref that points to the location of the transport file. △

Designating a Host File Format for the Accessing Host

Although the OpenVMS operating system transparently runs on two different host architectures, Alpha and VAX, if you create a SAS file on one host architecture that you intend for access on the other host architecture, then you must create the file in the file format of the accessing host. For example, at an OpenVMS Alpha host, you can create a SAS data set in OpenVMS VAX format. This feature is useful when migrating the OpenVMS operating environment between the two hardware architectures.

You use the HOSTFMT= option in the LIBNAME statement to designate the format of all files to be created in the specified library. Here is an example of designating OpenVMS Alpha format from an OpenVMS VAX host:

```
libname tranfile '[grades]' HOSTFMT=alpha;
```

By default, if you omit the HOSTFMT= option in the LIBNAME statement, you create SAS files in the format of the host at which you are creating the SAS files.

For details about the syntax for the LIBNAME statement and the HOSTFMT= option, see *SAS Companion for the OpenVMS Operating Environment*.

After you use the LIBNAME statement to designate the format of files to be created in that library, you must make the library available for use by using either of these methods:

- □ Use NFS (Network File System) third-party communications software to mount the file system for network access.
- □ Use NFS (Network File Services) to move the files between hosts.

Migrating SAS Files from an OpenVMS VAX System to an OpenVMS Alpha System

In order to migrate from an OpenVMS VAX environment to an OpenVMS Alpha environment, because their internal file representations are different, you must transport all data sets and catalogs.

Because the smallest numeric variable that can be represented on the OpenVMS systems are different from each other (VAX, 2 bytes; Alpha, 3 bytes) numeric variables in OpenVMS VAX SAS files must be increased to 3 bytes before the files can be used by the Alpha systems.

Use PROC CONTENTS or PROC DATASETS with the CONTENTS statement to verify the size of a data file's smallest numeric variables.

After you have determined that the VAX environment contains numeric variables that are 2 bytes long, you can use either of two SAS procedures to increase the OpenVMS VAX SAS variables to 3 bytes. One procedure is for data sets and catalogs; the other is for data sets only.

Using PROC CPORT for Data Sets and Catalogs

For both data sets and catalogs, at the source system, use PROC CPORT to create transport files.

At the target system, use PROC CIMPORT with the option and value EXTENDSN=YES. Here is an example.

```
filename tranfile 'SAS-data-library';
libname grades 'SAS-data-library';
proc cimport library=grades file=tranfile EXTENDSN=YES;
```

At the Alpha system, a numeric variable whose length is less than 8 bytes is automatically increased by 1 byte. Not only is a 2-byte variable increased to 3 bytes but so is a 5-byte variable increased to 6 bytes.

The advantage of this method is that PROC CPORT and PROC CIMPORT can be used to transport both data sets and catalogs. The disadvantages are that two systems are needed to perform the transfer and extra memory is used for storing 1 additional byte for each variable whose length is between 2 and 8, inclusive.

Using PROC VAXTOAXP for Data Sets Only

For data sets only, at the OpenVMS Alpha target system, use the VAXTOAXP procedure to convert 2-byte variables to 3-byte variables.

CAUTION:

Perform this procedure at the OpenVMS Alpha system after all files have been migrated from the source OpenVMS VAX system. If you perform this procedure on the OpenVMS VAX system, you will receive an error message. △

Here is an example of using PROC VAXTOAXP:

```
libname vlib v6 'user$disk:[dir]';
libname alib v8 '[nwdir]';

proc vaxtoaxp data=vlib.grades out=alib.grades;
run;
```

The Version 6 and Version8 behaviors of this procedure are different. In Version 6, this procedure is restricted to 2-byte variables only. For example, it does not increase a 5-byte variable to a 6-byte variable. However, in Version 8, this procedure does support a range of 2- to 7-byte variables. For example, the procedure increases a 7-byte variable to 8 bytes.

The limitation of this method is that PROC VAXTOAXP operates on only one data set at a time. For details about PROC VAXTOAXP, see *SAS Companion for the OpenVMS Operating Environment.*

However, as an alternative, you can use the SAS macro language to write instructions to process automatically an entire library of data sets. For details, see *SAS Macro Language: Reference.*

OpenVMS Error Messages

Given transport file is bad

For general recovery actions for this error message, see "Given transport file is bad" on page 110.

The transport file is suspected to be corrupt. When checking file attributes, the output confirms that the transport file contains a corrupting character:

```
$DIR/FULL transport-file
```

Output includes:

```
Record attributes: Carriage return Carriage control
```

If your system has the NFTCOPY (Network File Transfer Copy) command and you are moving the transport file to a DOS target system, remove the carriage return (CC) attribute from the transport file and move the transport file again to the target system:

```
NFTCOPY/IMAGE/FIXED/CC=NONE NODE"userid password"
  ::disk:[dir] tranfile target
```

Here is an example:

```
NFTCOPY/IMAGE/FIXED/CC=NONE CHEX "brown bird":
 dua0[brown]tranfile c:\blue\target
```

If your source system is running SAS Release 6.08 at maintenance level TS405 or later, set the NONE value to the CC= option in the LIBNAME or FILENAME statement, whichever is appropriate.

Note: See the top of the SAS log for the SAS release and maintenance level. △

Here is an example.

```
libname grades 'file-path';
libname tranfile xport 'file-path' cc=none;
proc copy in=grades out=tranfile;
run;
```

If you are running a SAS release that precedes Release 6.08 at maintenance level TS405, you must post-process the transport file to remove the carriage returns.

Create a new file named REMCC.FDL to contain these entries, including CARRIAGE_CONTROL to NONE.

```
RECORD
BLOCK_SPAN YES
CARRIAGE_CONTROL NONE
FORMAT FIXED
SIZE 80
```

Issue this DCL command to create a new file named NEWTRAN.SEQ:

```
$ CONVERT/FDL=REMCC.FDL TRAN.SEQ NEWTRAN.SEQ
$ DELETE TRAN.SEQ
```

Verify that the file attributes of the new transport file do not include carriage returns:

```
$ DIR/FULL NEWTRAN.SEQ
```

At the source system, transfer the transport file to the target system again.

If you are still unable to import a transport file that has the correct attributes, you may try using the re-blocking program in "Reblocking a Transport File" on page 114.

Member or library unavailable for use in file *file*

The transport file is suspected to be corrupt. See "Given transport file is bad" on page 85 for recovery actions.

Truncated record

For general recovery actions for this error message, see "Truncated record" on page 112.

This message is typically displayed when the transport file is moved to a virtual disk or a shared disk with other operating systems such as DOS, Macintosh, or UNIX. Virtual disk or shared disk directories often have a record format of STREAM instead of FIXED.

Verify the transport file attributes by using the DIR/FULL command.

To set record attributes correctly, create a new file named FIXREC.FDL file to contain these entries.

```
RECORD
BLOCK_SPAN YES
CARRIAGE_CONTROL NONE
FORMAT FIXED
SIZE 80
```

Issue this DCL command to create a new file named NEWTRAN.FDL:

```
$ EXCHANGE/NETWORK/TRANSFER_MOD=BLOCK/FDL=TRAN.FDL
 TRAN.SEQ NEWTRAN.SEQ
```

Verify that the new transport file attributes do not include carriage returns:

```
$ DIR/FULL NEWTRAN.SEQ
```

At the source system, transfer the transport file to the target system again.

Internal error from getting data

The transport file is suspected to be corrupt. See “Truncated record” on page 86 for recovery actions.

Procedure VAXTOAXP is not supported on VAX

You get this message when migrating SAS files from an OpenVMS VAX system to an OpenVMS Alpha system. Only the Alpha system supports this procedure. The VAX system *does not* support this procedure.

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

ISBN 1-58025-480-2

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

1st printing, October 1999

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