CHAPTER

3

Creating a Transport File at the Source Host

Supported Members

You can create a transport file for these members:

- Data set
- Catalog.

Note: In this document, the term *data set* refers to a SAS file of type DATA. △

Note: Traditional move methods do not support the DATA step view or the SAS/ACCESS view. These types of views must be re-created at the target host. △

Templates for Creating a Transport File

You use either of two traditional methods for creating a transport file:

- XPORT engine with PROC COPY
- PROC CPORT.

Here is the syntax template for the XPORT engine and PROC COPY:

❶ **LIBNAME** *orig-libref* '*SAS-data-library*';

❷ **LIBNAME** *new-libref* XPORT '*transport-file*';

❸ **PROC COPY** IN=*orig-libref* OUT=*new-libref*;

❹ **RUN**;

1 Use a LIBNAME statement to define the physical location of the source file in native format for which you are creating the transport file.

The base engine is assumed, by default.

2 Use a LIBNAME statement to define a physical location to store the transport file that is created.

Note: The *transport-file* argument that is specified in the LIBNAME statement that contains the XPORT engine is the file to which the transport data is written. If the transport file is not being written to the current directory, be sure to specify the full physical name that is recognized by the operating environment. For details, see the appropriate companion for the operating environment. Examples of the LIBNAME statement with the XPORT engine throughout this document assume the current directory. △

3 Use PROC COPY to read the source file in native format and to write out its content in transport format.

4 Use the RUN statement to execute the SAS statements.

Here is the syntax template for PROC CPORT:

❶ **LIBNAME** *libref* '*SAS-data-library*';

❷ **FILENAME** *fileref* '*transport-file*';

❸ **PROC CPORT** DATA=*libref.member* FILE=*fileref*;

❹ **RUN**;

1 Use a LIBNAME statement to define the physical location of the source file in native format for which you are creating the transport file.

The base engine is assumed, by default.

2 Use a FILENAME statement to define a physical location to store the transport file that is created.

Note: The *transport-file* argument that is specified in the FILENAME statement is the file to which the transport data is written. △

Note: For Releases 6.11 through Version 8, the FILENAME statement includes the FTP option that allows you to implicitly transfer the transport file directly to the target host. Using the FILENAME statement with the FTP option makes an explicit invocation of the FTP utility unnecessary. However, the examples in this chapter do not include the FTP option with the FILENAME statement. This chapter assumes an explicit invocation of the FTP utility. For an example of using the FTP option with the FILENAME statement, see "OS/390 TSO to UNIX File Transport" on page 146. △

3 Use PROC CPORT to read the source file from native format and to write its content in transport format.

Note: You can specify member types other than DATA. △

4 Use the RUN statement to execute the SAS statements.

Because LIBNAME statement syntax is host-specific, we use the variable *SAS-data-library* and *transport-file* to represent a disk location. However, to specify a magnetic medium format, see Part 5 for host-specific details. Also, see the appropriate operating environment companion for complete details about the syntax of the LIBNAME statement.

For complete details about the syntax for SAS statements which you must use to create a transport file, see *SAS Language Reference: Dictionary*. For PROC statements, see *SAS Procedures Guide*.

Describing the Content of the Transport File

If the person who creates the transport file at the source host is different from the person who restores the transport file at the target host, make sure that the details of the transport file are communicated in advance of the file transfer. Here is an example of the type of information that may be useful to the person who will restore the transport file at the target host:

Table 3.1 Description of Transport File

Type of Source Host and SAS Release Used	How Transport File Was Created	Transport Filename	Data Sets	Catalogs	Catalog Entries
OS/390 TSO SAS Version 8	PROC CPORT	TPORT.DAT	TEST.CITY TEST.CLASS	TEST.FORMATS	REGFMT SALEFMT SIZEFMT

Naming Conventions Used in the Examples in This Chapter

Consistent naming conventions are used in the examples in this chapter.

WORK
: is the default libref that points to the library that contains the data set GRADES.

XPORTOUT
: is the libref that points to the location of the transport file that is created with the XPORT engine.

CPORTOUT
: is the fileref that points to the location of the transport file that is created with PROC CPORT.

SOURCE
: is the libref that points to the location of the source file to be translated to transport format.

LIST
: is a catalog entry type.

GRADES
: is the name of a data set.

TESTCAT
: is the name of a catalog.

TESTNPGM
: is the name of a catalog entry.

Creating a Simple Data Set

If you choose to experiment, you can create several simple data sets in a library. Here is a sample SAS program that creates a data set named GRADES:

```
data grades;
  input student $ test1 test2 final;
  datalines;
Fred 66 80 70
Wilma 97 91 98
;
proc print data=grades;
run;
```

Here is the output:

```
        The SAS System
Thu                  10:54 Wed, October 20, 1999
OBS   STUDENT  TEST1  TEST2   FINAL
1      Fred     66      80      70
2      Wilma    97      91      98
```

Creating a Transport File for Member Type DATA

You can use any of these methods to create a transport file for member type DATA:

- DATA step
- PROC COPY
- PROC CPORT.

Using the DATA Step to Create a Transport File for One Data Set

This example uses the DATA step to create a transport file for a single data set.

```
libname source 'SAS-data-library';
libname xportout xport 'transport-file';
data xportout.grades;
   set source.grades;
run;
```

In the preceding example, the libref SOURCE points to the original location of the data set that is on the source host. The libref XPORTOUT points to a new location where the transport file will be created. The XPORT engine in this LIBNAME statement specifies that the data set is to be created in transport format. The SET statement reads the data set GRADES and re-creates it in transport format at the location specified in the DATA statement.

Using PROC COPY to Create a Transport File for One or More Data Sets

This example uses the COPY procedure to create a transport file for multiple data sets.

```
libname source 'SAS-data-library';
libname xportout xport 'transport-file';
```

```
proc copy in=source out=xportout memtype=data;
run;
```

In the preceding example, the libref SOURCE points to the original location of the library that is on the source host. The libref XPORTOUT points to a new location to which the transport file will be copied. The XPORT engine in this LIBNAME statement specifies that the library is to be created in transport format. The PROC COPY statement copies all data sets in the library that are identified in the IN= option to the new library that is identified in the OUT= option. The MEMTYPE=DATA option limits the files that are copied to type DATA, which excludes catalogs and views.

CAUTION:

Do not omit the MEMTYPE=DATA option. Otherwise, SAS attempts to copy the entire contents of the library (including catalogs and views) to the transport file. The XPORT engine does not support the CATALOG or the VIEW member type. Error and warning messages are written to the SAS log. △

This example uses PROC COPY to create a transport file for one data set:

```
libname source 'SAS-data-library';
libname xportout xport 'transport-file';
proc copy in=source out=xportout memtype=data;
   select grades;
run;
```

In the preceding example, the libref SOURCE points to the original location of the data set that is on the source host. The libref XPORTOUT points to a new location where the transport file will be copied. The XPORT engine in this LIBNAME statement specifies that the data set is to be created in transport format. The PROC COPY statement copies all data sets that are identified in the IN= option to the new library that is identified in the OUT= option. The MEMTYPE=DATA option limits the files that are copied to type DATA, which excludes catalogs and views. The SELECT statement specifies that only the data set GRADES be copied to the new library. You could specify more than one data set here. The omission of the SELECT statement would imply that all data sets be copied to the transport file.

Note: You can use the EXCLUDE statement to omit explicitly the data sets that you do not want rather than the SELECT statement to specify the data sets that you want. △

Using PROC CPORT to Create a Transport File for One or More Data Sets

This example uses the CPORT procedure to create a transport file for one data set.

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport data=source.grades file=cportout;
run;
```

In the preceding example, the libref SOURCE points to the original location of the data set that is on the source host. The fileref CPORTOUT points to a new location where the transport file will be created. The PROC CPORT statement copies as its source the file identified in the DATA= option to the new transport file identified in the FILE= option. The DATA= option specifies only one data set to be transported.

Specifying the LIBRARY= option instead of the DATA= option to PROC CPORT would expand the operation to include the entire contents of the library, which may contain multiple catalogs and data sets.

Here is an example of PROC CPORT that specifies all data sets in the library:

```
proc cport library=source file=cportout memtype=data;
```

Creating a Transport File for Member Type CATALOG

PROC CPORT is the only traditional method that you can use to create a transport file for member type CATALOG.

Using PROC CPORT to Create a Transport File for Multiple Catalogs

This example uses the CPORT procedure to create a transport file for multiple catalogs in a library.

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport library=source file=cportout memtype=catalog;
run;
```

In the preceding example, SOURCE points to the library that contains the catalogs that are on the source host. The fileref CPORTOUT points to a new location where the transport file will be created. The PROC CPORT statement copies from the specified library all members of the types that are identified in the MEMTYPE= option to the new transport file that is identified in the FILE= option.

You can use the EXCLUDE statement in PROC CPORT to omit explicitly the catalog entries that you do not want, or use the SELECT statement in PROC CPORT to specify the catalog entries that you want.

Using PROC CPORT to Create a Transport File for an Entire Catalog

This example uses the CPORT procedure to create a transport file for an entire catalog.

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport catalog=source.testcat file=cportout;
run;
```

In the preceding example, the libref SOURCE points to the original location of the catalog that is on the source host. Fileref CPORTOUT points to a new location where the transport file will be created. The PROC CPORT statement copies as its source the file that is identified in the CATALOG= option to the new transport file identified in the FILE= option. SOURCE specifies the libref and TESTCAT specifies the catalog name. The omission of the SELECT or EXCLUDE statements in PROC CPORT implies that the entire catalog be copied.

Using PROC CPORT to Create a Transport File for a Specific Catalog Entry Type

This example uses the CPORT procedure to create a transport file for a specific catalog entry type:

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
```

```
proc cport catalog=source.testcat file=cportout et=list;
run;
```

In the preceding example, the libref SOURCE points to the original location of the catalog that is on the source host. The fileref CPORTOUT points to a new location where the transport file will be created. The PROC CPORT statement copies as its source the file that is identified in the CATALOG= option to the new transport file identified in the FILE= option. The ET= option in PROC CPORT specifies that all catalog entries of type LIST be written to the new library. Alternatively, you can use the EET= option to exclude an entire entry type.

Using PROC CPORT to Create a Transport File for One or More Catalog Entries

This example uses the CPORT procedure to create a transport file for one or more catalog entries:

```
libname source 'SAS-data-library';
filename cportout 'transport-file';
proc cport catalog=source.mycat file=cportout;
  select testnpgm.list;
run;
```

In the preceding example, the libref SOURCE points to the original location of the catalog that is on the source host. The fileref CPORTOUT points to a new location where the transport file will be created. The PROC CPORT statement copies as its source the file that is identified in the CATALOG= option to the new transport file identified in the FILE= option.

In this example, SELECT TESTNPGM.LIST explicitly names a single catalog entry. However, you can specify one or more catalog entries by name.

You can use the EXCLUDE statement in PROC CPORT to omit explicitly the catalog entries that you do not want, or use the SELECT statement in PROC CPORT to specify catalog entries that you want.

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.