CHAPTER 5

Restoring a Transport File at the Target Host

Supported Members

You can restore a transport file that contains either of these SAS member types:

- Data set
- Catalog.

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

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

The method used to create a transport file uniquely defines the transport file format. Therefore, you must use the supported method for reading a transport file at the target host. For example, if PROC CPORT was used at the source host to create the transport file, you must use PROC CIMPORT at the target host to correctly decode the transport file. You cannot, for example, use PROC COPY at the target host to restore a transport file that was created with PROC CPORT.

Templates for Restoring a Transport File

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

- XPORT engine with PROC COPY

□ PROC CIMPORT.

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

❶ **LIBNAME** *libref* XPORT *'transport-file'*;

❷ **LIBNAME** *new-libref 'SAS-data-library'*;

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

❹ **RUN**;

1 Use a LIBNAME statement to define the physical location of the transport file that was transferred to the target host.

Note: The *transport-file* argument that is specified in the LIBNAME statement that includes the XPORT engine is the file from which the transport formatted data is read. If the transport file is not being read from 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. △

2 Use a LIBNAME statement to define a physical location in which to store the file to be restored in the target host's native format.

The base engine is used by default.

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

4 Use the RUN statement to execute the SAS statements.

Here is the syntax template for PROC CIMPORT:

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

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

❸ **PROC CIMPORT** INFILE=*fileref* LIBRARY=*libref* ;

❹ **RUN**;

1 Use a FILENAME statement to define a physical location for the transport file that was transferred to the target host.

Note: The *transport-file* argument that is specified in the FILENAME statement is the file from which the transport formatted data is read. △

2 Use a LIBNAME statement to define the physical location in which to store the file to be imported into the target host native format.

3 Use PROC CIMPORT to read in the transport file and to write out its content in the target host native format.

Note: You can specify either LIBRARY= *libref* or CATALOG=*libref.member*, as appropriate. △

4 Use the RUN statement to execute the SAS statements.

Because LIBNAME statement syntax is host-specific, we use the variables *SAS-data-library* and *transport-file* to represent a disk location. However, to specify the format of a magnetic medium, 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 that you must use to create a transport file, see *SAS Language Reference: Dictionary*. For PROC statements, see *SAS Procedures Guide*.

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.

XPORTIN
is the libref that points to the location of the transport file that is read with the XPORT engine.

IMPORTIN
is the fileref that points to the location of the transport file that is read with PROC CIMPORT.

TARGET
is the libref that points to the location of the file that contains the translated native format.

LIST
is the 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.

Determining the Content of the Transport File

If the person who restores the transport file at the target host is different from the person who creates the transport file at the source host, make sure you obtain information about the transport file in advance of the file restore operation. Here is an example of the type of information that may be useful for restoring the transport file to native format at the target host:

Table 5.1 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	PROC CPORT	TPORT.DAT	TEST.CITY	TEST.FORMATS	REGFMT
SAS Version 8			TEST.CLASS		SALEFMT
					SIZEFMT

You can infer the procedure that was used to create the transport file by using a text editor or by using an operating system read or view command to read the transport file. The XPORT engine and PROC CPORT create transport files whose headers look

different. For details, see “Determining Which Method Was Used to Create the Transport File” on page 20.

Also, you can use these procedures to list the contents of the transport file: PROC CATALOG, PROC CONTENTS, and PROC DATASETS. For details about these procedures, see the *SAS Procedures Guide*.

Restoring a Transport File of Member Type DATA

You can use one of these methods to restore a transport file for member type DATA:

- DATA step
- PROC COPY
- PROC CIMPORT.

Using a Data Step to Restore a Single Data Set from a Transport File

This example uses the DATA step to restore a data set from a transport file.

```
libname xportin xport 'transport-file';
libname target 'SAS-data-library';
data target.grades;
   set xportin.grades;
run;
```

In the preceding example, the libref XPORTIN points to the location of the exported data set that was transferred to the target host. The XPORT engine specifies that the data set is to be read in transport format. The libref TARGET points to a new location where the translated file will be copied. The SET statement reads the data set XPORTIN.GRADES in transport format and translates it and copies it to the location specified in the DATA statement. Because a DATA step with the XPORT engine was used at the source host to create the transport file for a single data set, only a data set can be restored at the target host.

Using PROC COPY to Restore One or More Data Sets from a Transport File

This example uses the COPY procedure to restore one or more data sets from a transport file.

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

In the preceding example, the libref XPORTIN points to the location where the transport file was transferred to the target host. The XPORT engine in this LIBNAME statement specifies that the transport file at this location is to be read in transport format. The libref TARGET points to a new location where the transport file will be copied in native format. The PROC COPY statement copies the selected data set GRADES from the library that is 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 to be copied to type DATA, which excludes catalogs and views.

Using a SELECT statement, you specify one or more specific data sets to be copied to the new library. To specify that all data sets in the transport file be copied, omit the SELECT statement from PROC COPY.

Note: You can use the EXCLUDE statement in PROC COPY 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 CIMPORT to Import One or More Data Sets from a Transport File

This example uses the CIMPORT procedure to import multiple data sets from a transport file.

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin library=target memtype=data;
run;
```

In the preceding example, the fileref IMPORTIN points to the location where the transport file was transferred to the target host. The libref TARGET points to a new location where the transport file will be copied. The PROC CIMPORT statement copies as its source the file that is identified in the INFILE= option to the location identified in the LIBRARY= option. The PROC CIMPORT statement implicitly translates the transport file into the target host native format.

Because the LIBRARY= option permits both data sets and catalogs to be copied to the library, you need to specify MEMTYPE=DATA to restrict the operation only to data sets in the library. Omitting the MEMTYPE= option permits both data sets and catalogs, in the file referenced by the fileref IMPORTIN, to be copied to the location referenced by the libref TARGET.

In order to subset the destination member in PROC CIMPORT, use either the SELECT statement, the EXCLUDE statement, or the MEMTYPE= option. Here is an example of subsetting:

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin library=target memtype=data;
  select grades;
run;
```

In the preceding example, the libref TARGET and the MEMTYPE= option points to the new location where the transport file will be copied. The fileref IMPORTIN points to the location where the transport file was transferred to the target host. The PROC CIMPORT statement copies as its source the file that is identified in the INFILE= option to the location identified in the LIBRARY= option. The PROC CIMPORT statement implicitly translates the transport file into the target host native format.

The SELECT statement selects only the data set GRADES for the library TARGET.

Importing a Transport File That Contains Member Type CATALOG

You can use only PROC CIMPORT to import a transport file of member type CATALOG. You must use PROC CIMPORT at the target host when PROC CPORT was used at the source host to create the transport file.

Using Compatible Destination Member Types in PROC CPORT and PROC CIMPORT

Make sure that you use destination member types in PROC CPORT and PROC CIMPORT that are compatible.

For statements used at source host:	At the target host, you are limited to:
CPORT LIBNAME=	CIMPORT LIBNAME= or DATA=
CPORT DATA=	CIMPORT LIBNAME= or DATA=
CPORT CATALOG=	CIMPORT LIBNAME= or CATALOG=

If destination members are incompatible, you receive either an error or warning message. See Chapter 15, “Preventing and Fixing Problems,” on page 103 for recovery actions for common error messages. For complete details about PROC CPORT and PROC CIMPORT syntax, see *SAS Procedures Guide*.

Using PROC CIMPORT to Import Multiple Catalogs from a Transport File

This example uses the CIMPORT procedure to import multiple catalogs from a transport file.

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin library=target memtype=catalog;
run;
```

To import multiple catalogs, specify the LIBRARY= option and MEMTYPE=CATALOG in PROC CIMPORT.

In the preceding example, the fileref IMPORTIN points to the location where the transport file was transferred to the target host. The libref TARGET points to a new location where the transport file will be copied. The PROC CIMPORT statement copies as its source the file that is identified in the INFILE= option to the location identified in the LIBRARY= option. Because the destination is a library, only the libref is specified. The MEMTYPE= option restricts the import to catalogs. PROC CIMPORT implicitly translates the transport file into the target host native format.

Using PROC CIMPORT to Import an Entire Catalog from a Transport File

This example uses the CIMPORT procedure to import an entire catalog from a transport file.

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin catalog=target.testcat;
run;
```

To import a single catalog, specify the CATALOG= option in PROC CIMPORT.

Using PROC CIMPORT to Import a Specific Catalog Entry Type from a Transport File

This example uses the CIMPORT procedure to import a specific catalog entry type from a transport file.

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin catalog=target.testcat et=list;
run;
```

To import a single entry type in a catalog, specify the ET= option in PROC CIMPORT. Also, the CATALOG= option in PROC CIMPORT must be specified.

Using PROC CIMPORT to Import Selected Catalog Entries from a Transport File

This example uses the CIMPORT procedure to import selected catalog entries from a transport file.

```
filename importin 'transport-file';
libname target 'SAS-data-library';
proc cimport infile=importin catalog=target.testcat;
 select testnpgm.list one.scl;
run;
```

The SELECT option specifies the catalog entries that you want by name. In this example, SELECT TESTNPGM.LIST ONE.SCL explicitly names the selected catalog entries. Also, the CATALOG= option in PROC CIMPORT must be specified.

As an alternative, you can use the EXCLUDE statement in PROC CIMPORT to omit explicitly catalog entries that you do not 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