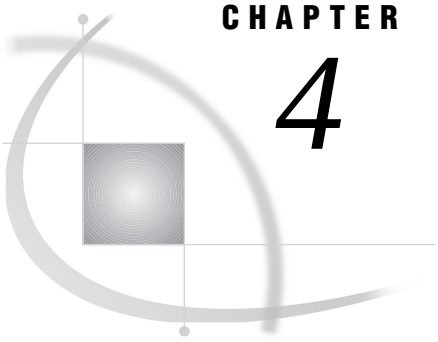

CHAPTER

4

Accessing BMDP, SPSS, and OSIRIS Files

Introduction **63**
Restrictions on the Use of These Engines **63**
Accessing BMDP Files **64**
Assigning a Libref to a BMDP File **64**
Referencing BMDP Files **64**
Examples of Accessing BMDP Files **64**
Accessing SPSS Files **65**
Assigning a Libref to an SPSS File **65**
Referencing SPSS Files **65**
Examples of Accessing SPSS Files **66**
Accessing OSIRIS Files **66**
Assigning a Libref to an OSIRIS File **66**
Referencing OSIRIS Files **67**
Examples of Accessing OSIRIS Files **67**

Introduction

The following read-only engines enable you to access files that were created with other vendors' software as if those files were written by the SAS System:

BMDP	accesses system files that were created with BMDP Statistical Software.
SPSS	accesses SPSS files that were created under Release 9 of SPSS as well as SPSS-X system files and portable files.
OSIRIS	accesses OSIRIS files.

You can use these engines in any SAS applications or procedures that do not require random access. For example, by using one of the engines with PROC CONTENTS and its _ALL_ option, you can determine the contents of an entire SPSS file at once.

Restrictions on the Use of These Engines

Because these are sequential engines, they cannot be used with the POINT= option of the SET statement, nor with the FSBROWSE, FSEDIT, or FSVIEW procedures in SAS/FSP software. However, you can use PROC COPY or a DATA step to copy a BMDP, SPSS, or OSIRIS file to a SAS data set, and then either use POINT= or use SAS/FSP to browse or edit the file.

Accessing BMDP Files

The BMDP engine can read only BMDP "save" files that were created on the same operating environment. For example, the BMDP engine under OS/390 cannot read BMDP files that were created under the OpenVMS operating environment.

Assigning a Libref to a BMDP File

In order to access a BMDP file, you must use the LIBNAME statement or LIBNAME function to assign a libref to the file.

You do not need to use a LIBNAME statement or function before running PROC CONVERT if you are using PROC CONVERT to convert a BMDP file to a SAS data file. (See “CONVERT” on page 235.)

Note that the LIBNAME statement has no options for the BMDP engine.

If you previously used a TSO ALLOC command or a JCL DD statement to assign a DDname to the BMDP file, you can omit the *physical-filename* (a physical file name in the OS/390 operating environment) in the LIBNAME statement or LIBNAME function and use the DDname as the libref. See “Accessing BMDP Files” on page 64.

For information on the LIBNAME statement, see “LIBNAME” on page 313. For information on the LIBNAME function, see “LIBNAME” on page 199.

Referencing BMDP Files

Because there can be multiple “save” files in a single physical BMDP file, you use the value of the BMDP CODE= argument as the name of the SAS data file. For example, if the BMDP "save" file contains CODE=ABC and CODE=DEF, and if the libref is XXX, you reference the files as XXX.ABC and XXX.DEF. All BMDP CONTENT types are treated the same, so even if file DEF has CONTENT=CORR under BMDP, SAS treats it as CONTENT=DATA.

In your SAS program, if you want to access the first BMDP "save" file in the physical file, or if there is only one "save" file, you can refer to the file as _FIRST_. This approach is convenient if you do not know the BMDP CODE= name.

Examples of Accessing BMDP Files

Suppose the physical file MY.BMDP.FILE contains the "save" file ABC. The following statements assign a libref to the data set and then run PROC CONTENTS and PROC PRINT on the BMDP file:

```
libname xxx bmdp 'my.bmdp.file';
proc contents data=xxx.abc;
proc print data=xxx.abc;
run;
```

In the next example, the TSO ALLOC command associates a DDname with the name of the physical file that comprises the BMDP *physical-filename*. The physical file name is omitted in the LIBNAME statement and LIBNAME function, because the libref that is used is the same as the DDname in the TSO statement. The PROC PRINT statement prints the data for the first "save" file in the physical file.

```
tso alloc f(xxx) da('my.bmdp.file') shr reu;
libname xxx bmdp;
proc print data=xxx._first_;
```

```
run;
```

Accessing SPSS Files

The SPSS engine supports native and portable file formats for both SPSS and SPSS-X files. The engine automatically determines which type of SPSS file it is reading and reads the file accordingly.

This engine can read only SPSS data files that were created under the same operating environment. For example, the SPSS engine under OS/390 cannot read SPSS files that were created under the OpenVMS operating environment. The only exception is an SPSS portable file, which can originate from any operating environment.

Assigning a Libref to an SPSS File

In order to access an SPSS file, you must use the LIBNAME statement or LIBNAME function to assign a libref to the file. Specify the SPSS engine in the LIBNAME statement as follows:

LIBNAME *libref* SPSS '*physical-filename* ';

libref
is a SAS libref.

SPSS
is the SPSS engine.

physical-filename
is the physical file name of the SPSS file.

The syntax of the LIBNAME function for SPSS is as follows:

LIBNAME(*libref*, '*physical-filename*', 'SPSS');

You do not need to use a LIBNAME statement or function before running PROC CONVERT if you are using PROC CONVERT to convert an SPSS file to a SAS data file. (See "CONVERT" on page 235.)

Note that the LIBNAME statement and function have no options for the SPSS engine.

If you previously used a TSO ALLOC command or a JCL DD statement to assign a DDname to the SPSS file, you can omit the *physical-filename* in the LIBNAME statement or function and use the DDname as the libref. (See the second example in "Examples of Accessing SPSS Files" on page 66.)

Referencing SPSS Files

SPSS data files do not have names. For these files, use a member name of your choice in SAS programs.

SPSS data files have only one logical member per file. Therefore, you can use _FIRST_ in your SAS programs to refer to the first data file.

Examples of Accessing SPSS Files

Suppose you want to read the physical file MY.SPSSX.FILE. The following statements assign a libref to the data set and then run PROC CONTENTS and PROC PRINT on the SPSS file:

```
libname xxx spss 'my.spssx.file';
proc contents data=xxx._first_;
proc print data=xxx._first_;
run;
```

In the next example, the TSO ALLOC command associates a DDname with the name of the physical file that comprises the SPSS *physical-filename*. The physical file name is omitted in the LIBNAME statement, because the libref that is used is the same as the DDname in the TSO command. The PROC PRINT statement prints the data in the first member of the SPSS data file.

```
tso alloc f(xxx) da('my.spssx.file') shr reu;
libname xxx spss;
proc print data=xxx._first_;
run;
```

Accessing OSIRIS Files

Although OSIRIS runs only under OS/390 and CMS, the SAS OSIRIS engine accepts an OS/390 data dictionary from any other operating environment that is running the SAS System. The layout of an OSIRIS data dictionary is the same on all operating environments. The data dictionary and data files should not be converted between EBCDIC and ASCII, however, because the OSIRIS engine expects EBCDIC data.

Assigning a Libref to an OSIRIS File

In order to access an OSIRIS file, you must use the LIBNAME statement or LIBNAME function to assign a libref to the file. Specify the OSIRIS engine in the LIBNAME statement as follows:

LIBNAME *libref* OSIRIS '*physical-filename* ' DICT='*dictionary-file-name*';

libref
is a SAS libref.

OSIRIS
is the OSIRIS engine.

physical-filename
is the physical file name of the data file.

dictionary-file-name
is the physical file name of the dictionary file. The *dictionary-file-name* can also be a DDname. However, if you use a DDname for the *dictionary-file-name*, do not use quotes.

Specify the OSIRIS engine in the LIBNAME function as follows:

LIBNAME(*libref*, '*physical-filename* ', 'OSIRIS', "DICT='*dictionary-file-name*'");

You do not need to use a LIBNAME statement or function before running PROC CONVERT if you are using PROC CONVERT to convert an OSIRIS file to a SAS data file. (See "CONVERT" on page 235.)

If you previously used a TSO ALLOC command or a JCL DD statement to assign a DDname to the OSIRIS file, you can omit the *physical-filename* in the LIBNAME statement or function. However, you must still use the DICT= option, because the engine requires both files. (See "Accessing OSIRIS Files" on page 66.)

Referencing OSIRIS Files

OSIRIS data files do not have individual names. Therefore, for these files you can use a member name of your choice in SAS programs. You can also use the member name _FIRST_ for an OSIRIS file.

Under OSIRIS, the contents of the dictionary file determine the file layout of the data file. A data file has no other specific layout.

You can use a dictionary file with an OSIRIS data file only if the data file conforms to the format that the dictionary file describes. Generally, each data file should have its own DICT file.

Examples of Accessing OSIRIS Files

Suppose you want to read the data file MY.OSIRIS.DATA, and the data dictionary is MY.OSIRIS.DICT. The following statements assign a libref to the data file and then run PROC CONTENTS and PROC PRINT on the file:

```
libname xxx osiris 'my.osiris.data'
   dict='my.osiris.dict';
proc contents data=xxx._first_;
proc print data=xxx._first_;
run;
```

The next example uses JCL. In this example, the DD statements can be omitted if the physical names are referenced in the LIBNAME statement.

```
//JOBNAME JOB
//STEP1   EXEC  SAS
//OSIR    DD  DSN=MY.OSIRIS.DATA,DISP=SHR
//DICT    DD  DSN=MY.OSIRIS.DICT,DISP=SHR
//SYSIN   DD  *
  /* Any one of the following libname */
  /* statements can be used.          */
libname osir osiris dict=dict;
libname xxx osiris 'my.osiris.data' dict=dict;
libname osir osiris dict='my.osiris.dict';
  /* Use this if the osir libref is used */
proc print data=osir._first_;
  /* Use this if the xxx libref is used  */
proc print data=xxx._first_;
//
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for the OS/390 Environment, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for the OS/390® Environment, Verison 8

1–58025–523–X

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

1st printing, November 1999