



CHAPTER

15

Procedures

SAS Procedures Under UNIX 219

SAS Procedures Under UNIX

This chapter describes SAS procedures that have behavior or syntax that is specific to UNIX environments. Each procedure description includes a brief “UNIX specifics” section that tells which aspect of the procedure is specific to UNIX. If the information under the “UNIX specifics” is “all,” then the procedure is described only in this documentation. Otherwise, the procedure is described in both this documentation and in *SAS Procedures Guide*.

BMDP

Calls any BMDP program to analyze data in a SAS data set

UNIX specifics: all

Syntax

```
PROC BMDP < options>;
  VAR variable-list;
  BY variable-list;
  PARMCARDS;
  ...BMDP control statements...
;
```

Details

BMDP is a library of statistical analysis programs maintained by BMDP Statistical Software Inc. Use the BMDP procedure in SAS programs to

- call a BMDP program to analyze data in a SAS data set
- convert a SAS data set to a BMDP save file.

Note: The BMDP procedure is available for the AIX, HP-UX, and Solaris operating environments only. Δ

You can analyze SAS data sets with BMDP programs by invoking this procedure. You can use the BMDP procedure any number of times to invoke BMDP. Before using the BMDP procedure, you must have access to the BMDP programs from your host because the procedure invokes them. Ask your system administrator for the location of the BMDP programs. To use the BMDP procedure, first specify the name of the BMDP program you want to invoke in the PROC BMDP statement. The VAR and BY statements can follow, although they are optional. The BMDP control statements follow the PARMCARDS statement.

To analyze BMDP data with the SAS System, create a BMDP save file in a BMDP program and then convert the save file to a SAS data set with the CONVERT procedure or use the BMDP engine. See “The BMDP Engine” on page 98 for more information on the BMDP engine.

PROC BMDP Statement

PROC BMDP *<options>*;

The following options can be used in the PROC BMDP statement:

CODE=*save-file*

assigns a name to the BMDP save file that the BMDP procedure creates from a SAS data set. The *save-file* corresponds to the CODE sentence in the BMDP INPUT paragraph. For example, if you use the following statement the BMDP INPUT paragraph must contain **CODE= 'JUDGES'**:

```
proc bmdp prog=bmdp3s code=judges;
```

CODE= is usually not necessary in the PROC BMDP statement. When CODE= is not specified, the name of the BMDP save file is the SAS data set name.

If you are converting a SAS data set to a BMDP save file, name the save file in the CODE sentence of the BMDP INPUT paragraph. To use the name of the SAS data set, specify that name in the CODE sentence. If you use a different name, it must match the name supplied in the CODE= option.

CONTENT=DATA | CORR | FREQ | MEAN

lets BMDP know if your SAS data set is a standard SAS data set (CONTENT=DATA) or if it contains a correlation matrix (CORR), variable means (MEAN), or frequency counts (FREQ). You need not specify the CONTENT= option for specially structured SAS data sets that were created by other SAS procedures. If you omit the CONTENT= option, the data set's TYPE value is used.

Note: BMDP may use a structure for special data sets (for example, a correlation matrix) that is different from the SAS structure. Be sure that the input SAS data set is in the form that BMDP expects. Δ

DATA=*SAS-data-set*

specifies the SAS data set that you want the BMDP program to process. If you do not specify the DATA= option, PROC BMDP uses the most recently created SAS data set.

INPUTFILE=*filename*

indicates the file in which the procedure will place the converted BMDP save file. It should match the name used in the INPUT paragraph of the BMDP control language.

LABEL=*variable*

specifies a variable whose values are to be used as case labels for BMDP. Only the first four characters of the values are used. The LABEL= variable must also be included in the VAR statement, if you use one.

LABEL2=*variable*

specifies a variable whose values are to be used as second case labels for BMDP; as with the LABEL= option, only the first four characters are used. The LABEL2= variable must also be given in the VAR statement, if you use one.

NOMISS

specifies that you want the BMDP program or save file to exclude observations with missing values.

PROG=BMDP*nn*

specifies the BMDP program that you want to run. For example, the following PROC BMDP statement runs the BMDP3S program:

```
proc bmdp prog=bmdp3s;
```

Note: If you only want to convert a SAS data set to a BMDP save file and do not want to run a BMDP program, omit the PROG= option and include the UNIT= option. Δ

UNIT=*n*

specifies the FORTRAN logical unit number for the BMDP save file that the BMDP procedure creates. The value that you specify for *n* must correspond to the UNIT value specified in the INPUT paragraph of the BMDP control language.

If you omit this option, *n* defaults to 3 and FT03F001 is used as the fileref for the save file. The following message is also printed:

Note: The UNIT= option was not specified. Unlike Version 5, the UNIT= option is required. Therefore, UNIT=3 is assumed. Ensure that the INPUT paragraph uses UNIT=3, or explicitly specify the correct UNIT value. Δ

WRKSPCE=*nn* | **PARM=***nn*

controls the allocation of a workspace in BMDP. The WRKSPCE= or PARM= value is passed as a parameter to BMDP programs and corresponds to the WRKSPCE= feature in BMDP MVS cataloged procedures. The default value for *nn* is 30.

VAR Statement

VAR *variable-list*

The VAR statement specifies the variables to be used in the BMDP program. When you do not include a VAR statement, the BMDP program uses all the numeric variables in the SAS data set.

BY Statement

BY *variable-list*

Use the BY statement with the BMDP procedure to obtain separate analyses on observations in groups that are defined with the BY variables. When you use a BY statement, the procedure expects the input data set to be sorted in order of the BY

variables or to have an appropriate index. If your input data set is not sorted in ascending order, you can do one of the following:

- Use the SORT procedure with a similar BY statement to sort the data.
- If appropriate, use the BY statement options NOTSORTED or DESCENDING.
- Create an index on the BY variables you want to use. For more information on creating indexes and using the BY statement with indexed data sets, see the *SAS Procedures Guide*.

If a BY statement is used, it is printed with the BMDP printed output to distinguish the BY group output.

For more information on the BY statement, see *SAS Language Reference: Dictionary*.

PARMCARDS Statement

PARMCARDS;

The PARMCARDS statement signals that the BMDP control language follows.

BMDP Control Statements

Put your BMDP control language after the PARMCARDS statement. These are similar for all BMDP programs; see the most current BMDP manual for information on their forms and functions.

The BMDP INPUT paragraph must include UNIT and CODE sentences. Their values must match the UNIT= and CODE= values given in the PROC BMDP statement. (If the PROC BMDP statement does not specify a UNIT= value, use 3 as the UNIT= value in the BMDP statements.) Use the SAS data set name as the CODE value unless you have specified a different name with the CODE= option in the PROC BMDP statement. Omit the VARIABLES paragraph from the BMDP statements because it is not needed when your input is a save file.

How Missing Values Are Handled

Before the BMDP procedure sends data to BMDP, it converts missing SAS values to the standard BMDP missing value. When you use the NOMISS option in the PROC BMDP statement, observations with missing values are not sent to the BMDP program.

Using BMDP Programs that Need FORTRAN Routines

Some BMDP programs, such as the ones for nonlinear regression, need to invoke the FORTRAN compiler and linkage editor before executing. All BMDP compilation and link editing must be completed before you use PROC BMDP.

Example of Creating and Converting a BMDP Save File

The following code converts the BMDP save file BOUT to the SAS data set FROMBMDP.

```
data temp;❶
  input a b c;
  cards;
datalines;
1 2 3
4 5 6
7 8 9
```

```

;
run;

proc contents;②
  title 'CONTENTS OF SAS DATA SET TO BE
        RUN THROUGH BMDP1D';
run;

options parmcards=bmdpcntl;③
filename bmdpcntl 'bmdp.inp';
proc bmdp prog=bmdp1d data=temp
  inputfile='temp.dat';④
  parmcards;
  /prob title='SHOW SAS/BMDP INTERFACE'.
  /input file='temp.dat'. code='TEMP'.
  /save code='BOUT'. new. file='bout.dat'.
  /end
  /finish
;
run;

filename fromfile 'bout.dat';⑤
proc convert bmdp=fromfile out=frombmdp;
run;

proc contents;⑥
title 'SAS DATA SET CONVERTED FROM BMDP
      SAVE FILE';
run;

proc print;
run;

```

- ① DATA TEMP creates a SAS data set called TEMP.
- ② The CONTENTS procedure shows the descriptive information for the data set TEMP.
- ③ The PARMCARDS= option gives a fileref for the parmcards file. The FILENAME statement indicates what file corresponds to that fileref for the PARMCARDS file. PROC BMDP gives the PARMCARDS file to the BMDP program.
- ④ PROC BMDP calls the BMDP program BMDP1D to analyze the data set TEMP.

The input BMDP save file, **temp.dat**, is specified both in the INPUTFILE= option and in the FILE sentence of the INPUT paragraph. The CODE sentence specifies the TEMP data set. The results are saved in the BMDP save file BOUT, which is stored in **temp.dat**, as specified in the SAVE paragraph.

- ⑤ The FILENAME statement associates the FROMFILE fileref with the **bout.dat** file, which contains the BOUT save file.

The CONVERT procedure converts the BMDP save file BOUT to the SAS data set FROMBMDP. The BMDP= option is set to

FROMFILE, since PROC CONVERT requires a fileref with the BMDP= option.

- ⑥ The CONTENTS and PRINT procedures show the new SAS data set, FROMBMDP.

See Also

- “The BMDP Engine” on page 98
- “CONVERT” on page 225
- Dixon, W.D.; Brown, M.B.; Engleman, L.; Frane, J.W.; Hill, M.A.; Jennrich, R.I.; and Toporek, J.D., eds. (1981), *BMDP Statistical Software*, Los Angeles: University of California Press.

CATALOG

Manages entries in SAS catalogs

UNIX specifics: FILE= option in the CONTENTS statement

Details

The FILE= option in the CONTENTS statement of the CATALOG procedure accepts a fileref. If the name specified does not correspond to a fileref, a file with that name and an extension of `.lst` is created in the current directory. For example, if MYFILE is not a fileref, the following code creates file `myfile.lst` in your current directory:

```
proc catalog catalog=sasuser.profile;
  contents file=myfile;
run;
```

Note: The filename that is created is always in lowercase, even if you specified it in uppercase. Δ

See Also

- *SAS Procedures Guide*

CONTENTS

Prints descriptions of one or more files from a SAS data library

UNIX specifics: information displayed in the SAS output

Details

The CONTENTS procedure produces the same information as the CONTENTS statement in the DATASETS procedure. See “DATASETS” on page 228 for more information.

See Also

- “The CONTENTS Procedure” in *SAS Procedures Guide*

CONVERT

Converts BMDP, OSIRIS, and SPSS system files to SAS data sets

UNIX specifics: all

Syntax

PROC CONVERT *product-specification* <*option-list*>;

Details

The CONVERT procedure converts BMDP, OSIRIS, and SPSS system files to SAS data sets. The procedure is supplied for compatibility with Version 5. The procedure invokes the appropriate engine to convert files.

CONVERT produces one output data set, but no printed output. The new data set contains the same information as the input system file; exceptions are noted in “How Missing Values Are Handled” on page 226.

The procedure converts system files from these three products:

- BMDP save files up to and including the most recent release of BMDP (available for AIX, HP-UX, and Solaris only).
- SPSS save files in Portable File Format (PDF).
- OSIRIS files through and including OSIRIS IV. (Hierarchical file structures are not supported.)

Because the BMDP, OSIRIS, and SPSS products are maintained by other organizations, changes may be made that make new system files incompatible with the current version of PROC CONVERT. SAS Institute upgrades PROC CONVERT to support changes to these products only when a new version of the SAS System is available.

In the PROC CONVERT statement, *product-specification* is required and can be one of the following:

BMDP=*fileref* <(CODE=*code* CONTENT=*content-type*)>

converts the first member of a BMDP save file created under UNIX (AIX) into a SAS data set. Here is an example:

```
filename save '/usr/mydir/bmdp.dat';
proc convert bmdp=save;
run;
```

If you have more than one save file in the BMDP file referenced by the *fileref* argument, you can use two options in parentheses after *fileref*. The CODE= option specifies the code of the save file that you want, and the CONTENT= option specifies the content of the save file. For example, if a file with CODE=JUDGES has a content of DATA, you can use the following statements:

```
filename save '/usr/mydir/bmdp.dat';
proc convert bmdp=save(code=judges
                      content=data);
run;
```

OSIRIS=*fileref* | *libref*

specifies a fileref or libref for the OSIRIS file to be converted into a SAS data set. You must also include the DICT= option.

SPSS=*fileref* | *libref*

specifies a fileref or libref for the SPSS file to be converted into a SAS data set. The SPSS file must be in Portable File Format, but it can be from any operating system.

Option-list can be one or more of the following:

DICT=*fileref* | *libref*

specifies a fileref or libref of the dictionary file for the OSIRIS file. DICT= is valid only when used with the OSIRIS product specification.

FIRSTOBS=*n*

gives the number of the observation where the conversion is to begin, so that you can skip observations at the beginning of the BMDP, OSIRIS, or SPSS file.

OBS=*n*

specifies the number of the last observation to be converted. This allows you to exclude observations at the end of the file.

OUT=*SAS-data-set*

names the SAS data set that will hold the converted data. If OUT= is omitted, the SAS System still creates a WORK data set and automatically names it DATA*n*, just as if you had omitted a data set name in a DATA statement. See Chapter 4, "Using SAS Files," on page 83 for more information.

How Missing Values Are Handled

If a numeric variable in the input data set has no value or a system missing value, CONVERT assigns it a missing value.

How Variable Names Are Assigned

The following sections explain how names are assigned to the SAS variables created by the CONVERT procedure.

CAUTION:

Be sure that the translated names will be unique. Variable names are translated as indicated in the following sections. Δ

Variable Names in BMDP Output Variable names from the BMDP save file are used in the SAS data set, but nontrailing blanks and all special characters are converted to underscores in the SAS variable names. The subscript in BMDP variable names, such as x(1), becomes part of the SAS variable name with the parentheses omitted: X1. Alphabetic BMDP variables become SAS character variables of corresponding length. Category records from BMDP are not accepted.

Variable Names in OSIRIS Output For single-response variables, the V1 through V9999 name becomes the SAS variable name. For multiple-response variables, the suffix R*n* is added to the variable name where *n* is the response. For example, V25R1 would be the first response of the multiple-response V25. If the variable after V1000 has 100 or more

responses, responses above 99 are eliminated. Numeric variables that OSIRIS stores in character, fixed-point binary, or floating-point binary mode become SAS numeric variables. Alphabetic variables become SAS character variables; any alphabetic variable of length greater than 200 is truncated to 200. The OSIRIS variable description becomes a SAS variable label, and OSIRIS print format information becomes a SAS format.

Variable Names in SPSS Output SPSS variable names and variable labels become variable names and labels without change. SPSS alphabetic variables become SAS character variables of the same length. SPSS blank values are converted to SAS missing values. SPSS print formats become SAS formats, and the SPSS default precision of no decimal places becomes part of the variables' formats. The SPSS DOCUMENT data is copied so that the CONTENTS procedure can display it. SPSS value labels are not copied.

File Conversion Examples

These three examples show how to convert BMDP, OSIRIS, and SPSS files to SAS data sets.

Converting a BMDP save file

The following statements convert a BMDP save file and produce the temporary SAS data set TEMP, which contains the converted data:

```
filename bmdpfile 'bmdp.savefile';
proc convert bmdp=bmdpfile out=temp;
run;
```

Converting an OSIRIS file

The following statements convert an OSIRIS file and produce the temporary SAS data set TEMP, which contains the converted data:

```
filename osirfile 'osirdata';
filename dictfile 'osirdict';
proc convert osiris=osirfile dict=dictfile
            out=temp;
run;
```

Converting an SPSS file

The following statements convert an SPSS Release 9 file and produce the temporary SAS data set TEMP, which contains the converted data:

```
filename spssfile 'spssfile.num1';
proc convert spss=spssfile out=temp;
run;
```

Comparison with Interface Library Engines

The CONVERT procedure is closely related to the interface library engines BMDP, OSIRIS, and SPSS. (In fact, the CONVERT procedure uses these engines.) For example, the following two sections of code provide identical results:

```
filename myfile 'mybmdp.dat';
proc convert bmdp=myfile out=temp;
run;

libname myfile bmdp 'mybmdp.dat';
data temp;
```

```

        set myfile._first_;
run;

```

However, the BMDP, OSIRIS, and SPSS engines provide more extensive capability than PROC CONVERT. For example, PROC CONVERT converts only the first BMDP member in a save file. The BMDP engine, in conjunction with the COPY procedure, copies all members.

See Also

- “Accessing BMDP, OSIRIS, or SPSS Data Files” on page 98

DATASETS

Lists, copies, renames, and deletes SAS files, manages indexes for and appends SAS data sets in a SAS data library

UNIX specifics: information displayed on the SAS log and output

Details

Some of the information, such as the filenames and access permissions, that is generated by the DATASETS procedure depends on your operating environment. The information generated by the CONTENTS statement also varies according to the device type or access method associated with the data set. If you specify the DIRECTORY option in the CONTENTS statement, the directory information is displayed in both the log and output windows.

Example

The following SAS code creates two data sets, `grades.sas7bdat` and `majors.sas7bdat`, and runs PROC DATASETS on `majors.sas7bdat`.

```

options nodate pageno=1;
libname classes '.';

data classes.grades (label='First Data Set');
  input student year state $ gradel grade2;
  label year='Year of Birth';
  format gradel 4.1;
  datalines;
1000 1980 NC 85 87
1042 1981 MD 92 92
1095 1979 PA 78 72
1187 1980 MA 87 94
;
run;

data classes.majors(label='Second Data Set');
  input student $ year state $ gradel grade2 major $;
  label state='Home State';

```

```

        format grade1 5.2;
        datalines;
1000 1980 NC 84 87 Math
1042 1981 MD 92 92 History
1095 1979 PA 79 73 Physics
1187 1980 MA 87 74 Dance
1204 1981 NC 82 96 French
;
run;

proc datasets library=classes;
    contents data=majors directory;
run;

```

The output of this example is shown in Output 15.1 on page 229. The first page of output from this example SAS code is produced by the DIRECTORY option in the CONTENTS statement. This information also appears on the SAS log. Pages 2 and 3 in this output describe the data set `classes.majors.sas7bdat` and appear only on the SAS output.

Output 15.1 PROC DATASETS Example

```

                                The SAS System

                                The DATASETS Procedure

                                -----Directory-----

Libref:                          CLASSES
Engine:                           V8
Physical Name:                     /remote/u/yourid
File Name:                         /remote/u/yourid
Inode Number:                      283433
Access Permission:                rwxrwxrwx
Owner Name:                       yourid
File Size (bytes):                1024

#  Name      Memtype      File
                                size  Last modified
-----
1  MAJORS    DATA             16384  01JUN1999:15:50:11
2  GRADES    DATA             16384  01JUN1999:15:50:12

```

```

                                The SAS System

                                The DATASETS Procedure

Data Set Name: CLASSES.MAJORS                Observations:      5
Member Type:   DATA                        Variables:         6
Engine:        V8                            Indexes:           0
Created:       15:50 Tuesday, June 1, 1999  Observation Length: 48
Last Modified: 15:50 Tuesday, June 1, 1999 Deleted Observations: 0
Protection:                               Compressed:        NO
Data Set Type:                               Sorted:            NO
Label:         Second Data Set

                                -----Engine/Host Dependent Information-----

Data Set Page Size:      8192
Number of Data Set Pages: 1
First Data Page:        1
Max Obs per Page:       169
Obs in First Data Page: 5
Number of Data Set Repairs: 0
File Name:               /remote/u/yourid/majors.sas7bdat
Release Created:         8.00.00B
Host Created:            HP-UX
Inode Number:           2834581
Access Permission:      rw-r--r--
Owner Name:              yourid
File Size (bytes):      16384
    
```

```

                                The SAS System

                                The DATASETS Procedure

                                -----Alphabetic List of Variables and Attributes-----

#   Variable   Type   Len   Pos   Format   Label
-----
4   grade1     Num    8     8     5.2
5   grade2     Num    8    16
6   major      Char    8    40
3   state      Char    8    32           Home State
1   student    Char    8    24
2   year       Num    8     0
    
```

See Also

- “The DATASETS Procedure” in *SAS Procedures Guide*
- “The CONTENTS Procedure” in *SAS Procedures Guide*

OPTIONS

Lists the current values of all SAS system options

UNIX specifics: options available only under UNIX

Details

PROC OPTIONS lists the current values of the system options that are available in all operating environments (portable options) and, if you specify the HOST option in the PROC OPTIONS statement, it lists those options that are available only under UNIX (host options). The option values displayed by PROC OPTIONS depend on the default values shipped with the SAS System, the default values specified by your site administrator, the default values in your own configuration file, any changes made in your current session through the System Options window or OPTIONS statement, and possibly, the device on which you are running SAS.

For more information on a specific option, refer to Chapter 17, “SAS System Options,” on page 253.

See Also

- “The OPTIONS Procedure” in *SAS Procedures Guide*

PMENU

Defines PMENU facilities for windows created with SAS software

UNIX specifics: ATTR= and COLOR= options in the TEXT statement have no effect; ACCELERATE= and MNEMONIC= options in the ITEM statement are ignored

Details

The PMENU procedure defines PMENU facilities for windows created by using the WINDOW statement in base SAS software, the %WINDOW macro statement, the BUILD procedure of SAS/AF software, or the SAS Component Language (SCL) PMENU function with SAS/AF, SAS/CALC, and SAS/FSP software.

The ATTR= and COLOR= options in the TEXT statement have no effect under UNIX. The colors and attributes for text and input fields are controlled by the CPARMS colors specified in the SASCOLOR window. See “Customizing Colors” on page 66 for more information.

The ACCELERATE= and the MNEMONIC= options in the ITEM statement are ignored under UNIX.

See Also

- “The PMENU Procedure” in *SAS Procedures Guide*

PRINTTO

Defines destinations for SAS procedure output and the SAS log

UNIX specifics: *destination*

Details

LOG=*destination*

Destination is a fully qualified pathname (in quotes), an environment variable, a fileref, or a file in the current directory (without extension).

PRINT=*destination*

Destination is a fully qualified pathname (in quotes), an environment variable, a fileref, or a file in the current directory (without extension). If you specify a fileref that is defined with the PRINTER device-type keyword, output is sent directly to the printer.

Examples

The following statements send any SAS log entries that are generated after the RUN statement to the external file that is associated with the fileref MYFILE:

```
filename myfile '/users/myid/mydir/mylog';
proc printto log=myfile;
run;
```

If MYFILE has not been defined as a fileref, PROC PRINTTO will create the file **myfile.log** in the current directory.

The following statements send any procedure output that is generated after the RUN statement to file **/users/myid/mydir/myout**:

```
proc printto print='/users/myid/mydir/myout';
run;
```

The following statements send the procedure output from the CONTENTS procedure directly to the system printer:

```
filename myfile printer;
proc printto print=myfile;
run;

proc contents data=oranges;
run;
```

To redirect the SAS log and procedure output to their original default destinations, run PROC PRINTTO without any options:

```
proc printto;  
run;
```

If MYPRINT and MYLOG have not been defined as filerefs, then the following statements send any SAS procedure output to **myprint.lst** and any log output to **mylog.log** in the current directory:

```
proc printto print=myprint log=mylog;  
run;
```

If filerefs MYPRINT and MYLOG had been defined, the output would have gone to the files associated with these filerefs.

See Also

- *SAS Procedures Guide*
- Chapter 6, “Routing Output,” on page 125

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Companion for UNIX Environments, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Companion for UNIX Environments, Version 8

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

ISBN 1-58025-502-7

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, in 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.

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.