CHAPTER 4

DBLOAD Procedure Reference

Introduction

The DBLOAD procedure enables you to create and load PC files. This chapter provides general reference information for the DBLOAD procedure.

File-specific information for the SAS/ACCESS interface to your PC file format is included in Chapter 5 and later. This chapter presents the PROC DBLOAD options and statements common to all formats.

For a quick summary of the tasks that are described in this chapter, see the tab page enclosed with your PC file format chapter.

Refer to *SAS Language Reference: Dictionary* and to the SAS documentation for your operating environment for more information on SAS data sets and SAS data libraries and their naming conventions or for help with the terminology used in this procedure description.

Remember that help is available when you select the **Help** menu.

Note to UNIX and OS/390 Users

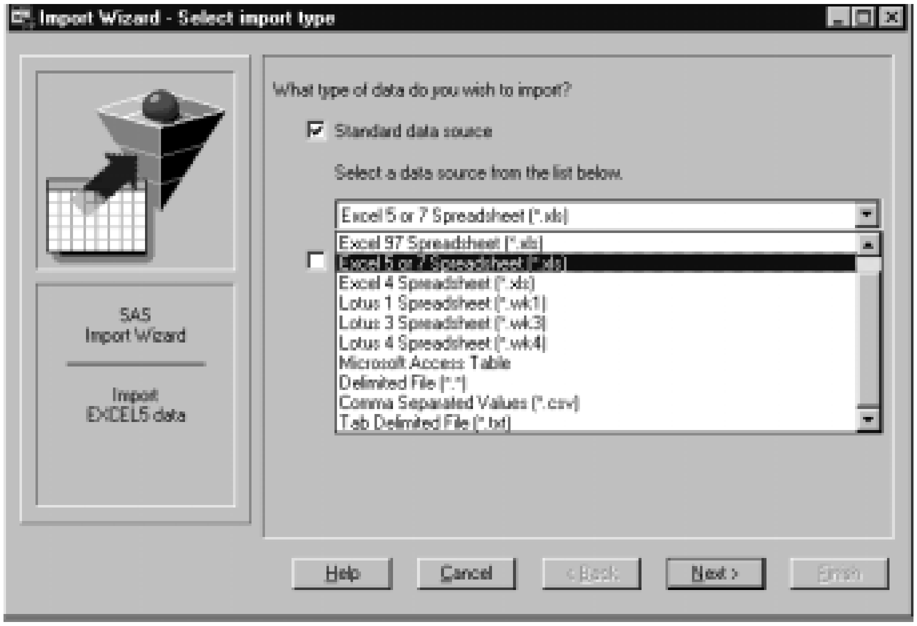
If you are running this SAS/ACCESS interface under the UNIX or OS/390 operating environment, this chapter does not apply to you. Instead, see Chapter 3, “DBF and DIF Procedures,” on page 33. Under UNIX and PC hosts, you can use these procedures to convert a DBF or DIF file to a SAS data set or a SAS data set to a DBF or DIF file. Under OS/390, you can use PROC DBF *only* to convert a DBF file to a SAS data set or a SAS data set to a DBF file.

Import/Export Facility

UNIX and PC users can access DBF, WK*n*, Excel 4 and 5, Excel 97, and MS Access data through the Import/Export facility or by using the IMPORT and EXPORT procedures. An overview is included in Chapter 5, “Import/Export Facility and Procedures,” on page 51.

To use the point-and-click interface from a SAS PROGRAM EDITOR window, select the **`File`** menu and then select the **`Import Data`** or **`Export Data`** item. Information about how to import or export PC file data is available from the Help button. The following is a sample Import window:

Display 4.1 Import Window



Import Wizard - Select import type

What type of data do you wish to import?

☑ Standard data source

Select a data source from the list below.

Excel 5 or 7 Spreadsheet (*.xls)

□

Excel 97 Spreadsheet (*.xls)
Excel 5 or 7 Spreadsheet (*.xls)
Excel 4 Spreadsheet (*.xls)
Lotus 1 Spreadsheet (*.wk1)
Lotus 3 Spreadsheet (*.wk3)
Lotus 4 Spreadsheet (*.wk4)
Microsoft Access Table
Delimited File (*.*)
Comma Separated Values (*.csv)
Tab Delimited File (*.txt)

To write code to import or export PC file data, refer to the detailed descriptions of the IMPORT and EXPORT procedures in the *SAS Procedures Guide.* This documentation also includes several examples.

DBLOAD Naming Conventions

In Version 7 of SAS software, SAS data set names and variable names can be up to 32 characters in length. However, when you use the DBLOAD procedure to load a SAS data set into a PC file, the SAS variable names cannot exceed 8 characters. This restriction is applied in order to have compatibility with Version 6 naming conventions.

DBLOAD Procedure Syntax

PROC DBLOAD <DBMS=*pc-file-format*>
<DATA=<*libref.*>*SAS-data-set*>;

Database-Description Statement

PATH='*path-and-filename*<*.PC-file-extension*>' |
<'>*filename*<'> | *fileref* ;

Editing Statements

ACCDESC=<*libref.*>*access-descriptor*;

DELETE *variable-identifier-1*
<*...variable-identifier-n*;>

ERRLIMIT=*error-limit*;

LABEL;

LIMIT=*load-limit*;< >

LIST <ALL | COLUMNS | FIELDS |
variable-identifier>;

QUIT;

RENAME *variable-identifier-1*=<'>*column-name-1*<'>
<*...variable-identifier-n*<'>*column-name-n*<'>>;

RESET ALL | *variable-identifier-1* <*...variable-identifier-n* >;

WHERE *SAS-where-expression*;

Creating and Loading Statement

LOAD;

RUN;

Additional statements may be used with your PC file format. See your PC file format chapter for more information.

Description

The DBLOAD procedure creates and loads (that is, transfers data to) PC files using data from a SAS data set. This data can be from any of the following: a SAS data file, a PROC SQL view, a DATA step view, or a view descriptor from any SAS/ACCESS interface product. The DBLOAD procedure associates each SAS variable with a PC file column and assigns a default name and data type to each column. You can use the default information or change it as necessary. When you are finished customizing the columns, the procedure creates the PC file and loads it with the input data.

PROC DBLOAD Statement Options

The following options can be used in the PROC DBLOAD statement.

DBMS=*pc-file-format*
specifies which PC file format that you want to access. Specify DBMS=DBF for .DBF files, DBMS=DIF for .DIF files, DBMS=WK1 | WK3 | WK4 for WK*n* files, or DBMS=XLS for .XLS files. The DBMS= option is required.

DATA=<*libref.*>*SAS-data-set*
specifies the input data set. The input data can be retrieved from a SAS data file, a PROC SQL view, a DATA step view, or a SAS/ACCESS view descriptor. If the data set is permanent, you must use its two-level name, *libref.SAS-data-set*. If you omit the DATA= option, the default is the last SAS data set that was created.

Procedure Statements

To invoke PROC DBLOAD, you use the options listed in "Options" along with certain statements. The statements that you choose are determined by your task and your PC file format. These statements vary per PC file and may be optional; see your PC file format chapter for more information.

Dictionary

ACCDESC=

Creates an access descriptor based on the new file.

Optional statement

Syntax

ACCDESC=*libref.access-descriptor*;

Details The ACCDESC= statement creates an access descriptor based on the PC file that you are creating and loading. After the new PC file is created and loaded, the access descriptor is automatically created. You must specify an access descriptor that does not already exist.

An editing statement, such as ACCDESC=, must be specified after the database-description statements when you create and load a file. See “LOAD” on page 47 for more information.

The ACCDESC= statement has two aliases: ACCESS= and AD=.

DELETE

Prevents variables from being loaded into the new PC file.

Optional statement

Interacts with: RENAME, RESET

Syntax

DELETE *variable-identifier-1* <*...variable-identifier-n*>;

Details The DELETE statement drops the specified SAS variables from the PC file being created. The *variable-identifier* argument can be either the SAS variable name or the positional equivalent from the LIST statement. The positional equivalent is the number that represents the variable's place in the data set. For example, if you want to drop the third variable, submit the following statement:

```
delete 3;
```

You can drop as many variables as you want in one DELETE statement. If you drop more than one variable, separate the identifiers with spaces, not commas.

Even if you drop a variable from the list of variables, the positional equivalents of the variables do not change. For example, if you drop the second variable, the third variable is still referenced by the number 3, not 2.

An editing statement, such as DELETE, must be specified after the database-description statements when you create and load a file. See "LOAD" on page 47 for more information.

ERRLIMIT=

Stops loading data after a specified number of errors.

Optional statement

Default: 100

Syntax

ERRLIMIT=*error-limit*;

Details The ERRLIMIT= statement stops loading observations after the specified number of errors has occurred while inserting rows into the file.

The *error-limit* argument must be a nonnegative integer. Specify ERRLIMIT=0 to allow an unlimited number of errors to occur.

An editing statement, such as ERRLIMIT=, must be specified after the database-description statements when you create and load a file. See "LOAD" on page 47 for more information.

LABEL

Causes column names to default to SAS labels.

Optional statement

Interacts with: RENAME, RESET

Syntax

LABEL;

Details The LABEL statement causes the column names to default to the SAS variable labels when the new table is created. If a SAS variable has no label, the variable name is used. If the label is too long to be a valid column name, the label is truncated.

For the LABEL statement to take effect, the RESET statement must be used *after* the LABEL statement.

An editing statement, such as LABEL, must be specified after the database-description statements when you create and load PC files. See "LOAD" on page 47 for more information.

LIMIT=

Limits the number of observations loaded.

Optional statement

Default: 5000

Syntax

LIMIT= *load-limit*;

Details The LIMIT= statement places a limit on the number of observations that can be loaded into a new file. The maximum number for the limit statement varies with each PC files. The *load-limit* argument must be a nonnegative integer. To load all the observations from your input data set, specify LIMIT=0.

If you omit the LIMIT= statement, a maximum of 5,000 observations are inserted.

LIST

Lists information about the variables to be loaded.

Optional statement

Default: ALL

Syntax

LIST<ALL | COLUMNS | FIELDS | *variable-identifier*>;

Details The LIST statement lists information about all or some of the SAS variables to be loaded into the new file. By default, the list is sent to the SAS log.

The LIST statement can take one or more of the following arguments:

ALL
: lists information about all the variables in the input SAS data set, whether or not those variables are selected for the load.

COLUMNS
lists information about only the input SAS variables that are selected for the load. This argument does not apply to DBF files.

FIELDS
lists information about only the input SAS variables that are selected for the load.

variable-identifier
lists information about only the specified variable. The *variable-identifier* argument can be either the SAS variable name or the positional equivalent. The positional equivalent is the number that represents the variable's position in the data set. For example, if you want to list information for the column associated with the third SAS variable, submit the following statement:

```
list 3;
```

You can specify LIST as many times as you want while creating a file; specify LIST before the LOAD statement to see the entire table. LIST must be specified after the database-description statements. See "LOAD" on page 47 for more information.

LOAD

Creates and loads the new PC file.

Required statement

Syntax

LOAD;

Details The LOAD statement causes the interface view engine to create a file and to transfer data to it from the input SAS data set after the DBLOAD procedure is submitted for processing. This statement is required to create and load a new file.

When you create and load a file, you must place statements or groups of statements in a certain order after the PROC DBLOAD statement and its options, as follows:

1. Database-description statements: PATH= and your PC file-specific statements.
2. Editing statements: ACCDESC=, DELETE, ERRLIMIT, LABEL, LIMIT=, LIST, RENAME, RESET, and WHERE. The order within this group usually does not matter; see the individual statements for more information. QUIT is also an editing statement but using it immediately terminates PROC DBLOAD.
3. Creating and loading statement: LOAD must appear last before RUN in order to create and load the new table.
4. RUN statement: this statement is used to process the DBLOAD procedure.

PATH=

Indicates the name and path of the PC file to be created and loaded.

Required statement

Syntax

PATH=*'path-and-filename<.PC-file-extension>'* |
    <'>filename<'> | *fileref*;

Details The PATH= statement indicates the path and name of the PC file you want to create and load. The length of the filename can vary with the operating environment. See the SAS documentation for your operating environment for any restrictions.

The PATH= statement can take one of the following arguments:

'path-and-filename <.PC-file-extension>'
: specifies the fully qualified path and filename. You must enclose the entire path and filename in quotes, including the appropriate PC file extension, such as .DBF, .DIF, .WK*n*, or .XLS. If you omit the file extension, SAS/ACCESS software supplies it for you.

<'>filename <'>
: specifies the name of a file. The file must be located in your current (default) directory. If no extension is specified, the SAS/ACCESS interface supplies it for you. If the filename includes characters that are invalid in SAS names, such as the dollar sign ($) or if the filename begins with a number, you must enclose the entire filename in quotes.

fileref
: specifies a fileref that references the path and name of the file. (Assigning filerefs with the FILENAME statement is described in *SAS Language and Procedures: Usage*.)

A file with the same name must not already exist; if one does exist, it is not overwritten. An error message is written to the SAS log, and the PC file specified in this statement is not loaded.

QUIT

Exits the DBLOAD procedure.

Optional statement

Syntax

QUIT;

Details The QUIT statement exits the procedure without further processing. EXIT is an alias for the QUIT statement.

RENAME

Renames PC file columns.

Optional statement

Interacts with: DELETE, LABEL, RESET

Syntax

RENAME *variable-identifier-1*=<'>*column-name-1*<'>
    <*...variable-identifier-n* =<'>*column-name-n*<'>>;

Details The RENAME statement changes the names of the PC file's columns that are associated with the listed SAS variables. If you omit the RENAME statement, all the column names default to the corresponding SAS variable names (unless the LABEL statement is specified).

The *variable-identifier* argument can be either the SAS variable name or the positional equivalent from the LIST statement. The positional equivalent is the number that represents the variable's place in the data set. For example, if you want to rename the column associated with the third SAS variable, submit the following statement:

```
rename 3='employname';
```

The *column-name* argument must be a valid PC file column name. If the column name includes lowercase characters, special characters, or national characters, you must enclose the column name in quotes.

The RENAME statement enables you to include variables that you have previously deleted. For example, suppose you submit the following statements:

```
delete 3;
rename 3='empname';
```

The DELETE statement first drops the third variable. Then the RENAME statement includes the third variable and assigns the name EMPNAME and the default column type to it.

You can rename as many variables as you want in one RENAME statement. The RENAME statement overrides the LABEL statement for columns that are renamed. COLUMN is an alias for the RENAME statement.

An editing statement, such as RENAME, must be specified after the database-description statements when you create and load a PC file. See "LOAD" on page 47 for more information.

RESET

Resets column names and data types to their default values.

Optional statement

Interacts with: DELETE, LABEL, RENAME

Syntax

RESET ALL | *variable-identifier-1*
 <...variable-identifier-n>;

Details The RESET statement resets the columns that are associated with the listed SAS variables to the default column name, column data type, and ability to accept null values. If you specify ALL, all columns are reset to their default values, and any deleted columns are restored with their default values.

The *variable-identifier* argument can be either the SAS variable name or the positional equivalent from the LIST statement. The positional equivalent is the number that represents the variable's place in the data set. For example, if you want to reset the column associated with the third SAS variable, submit the following statement:

```
reset 3;
```

You can reset as many columns as you want in one RESET statement.

You must use the RESET statement after the LABEL statement for the LABEL statement to take effect.

An editing statement, such as RESET, must be specified after the database-description statements when you create and load a PC file. See "LOAD" on page 47 for more information.

WHERE

Loads a subset of data into the new PC file.

Optional statement

Syntax

WHERE *SAS-where-expression*;

Details The WHERE statement loads a subset of observations into the new PC file. The *SAS-where-expression* must be a valid SAS WHERE statement that uses SAS variable names (not column names) as defined in the input data set. The following example loads only the observations in which the SAS variable COUNTRY has the value **Brazil**.

```
where country='Brazil';
```

For more information on the syntax of the SAS WHERE statement, see *SAS Language Reference: Dictionary*.

An editing statement, such as WHERE, must be specified after the database-description statements when you create and load a DBMS table. See "LOAD" on page 47 for more information.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS® Software for PC File Formats: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/ACCESS® Software for PC File Formats: Reference, Verison 8

ISBN 1-58025–544–2

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.