

## CHAPTER

## 8

## DIF Chapter

<i>Note to UNIX and OS/390 Users</i>	107
<i>Import/Export Facility</i>	107
<i>Understanding DIF Essentials</i>	108
<i>DIF Files</i>	108
<i>DIF File Naming Conventions</i>	109
<i>DIF File Data Types</i>	109
<i>How the SAS System Handles Date and Time Values</i>	110
<i>Datetime Conversions in the ACCESS Procedure</i>	110
<i>Datetime Conversions in the DBLOAD Procedure</i>	111
<i>ACCESS Procedure Data Conversions</i>	111
<i>ACCESS Procedure: DIF Specifics</i>	112
<i>ACCESS Procedure Statements for DIF Files</i>	112
<i>DBLOAD Procedure: DIF Specifics</i>	114
<i>DBLOAD Procedure Statements for DIF Files</i>	114
<i>How the SAS/ACCESS Interface Works</i>	115

## Note to UNIX and OS/390 Users

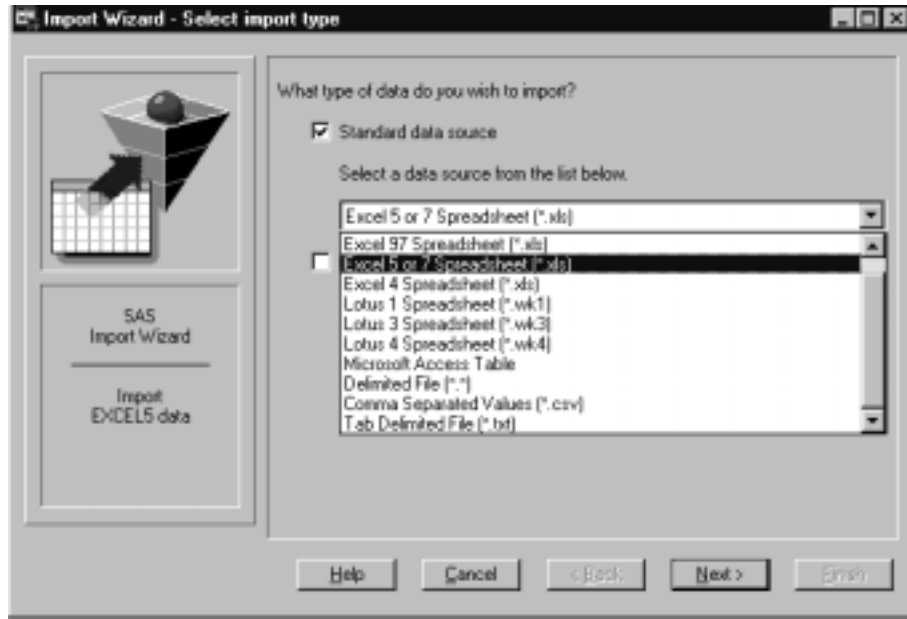
If you are running this SAS/ACCESS interface under the UNIX or OS/390 operating environment, this chapter does not apply to you. Instead, see Chapter 3, “DBF and DIF Procedures,” on page 33. Under UNIX and PC hosts, you can use only PROC DIF to convert a DIF file to a SAS data set or a SAS data set to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

## Import/Export Facility

UNIX and PC users can access most PC file data through the Import/Export facility or by using the IMPORT and EXPORT procedures. However, you cannot access DIF data by these methods. An overview of importing and exporting data 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 data is available from the [Help](#) button. The following is a sample Import window:

Display 8.1 Import Window



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

---

## Understanding DIF Essentials

This chapter introduces SAS System users to DIF files that can be created using software under a variety of microcomputer software packages.

This chapter focuses on the terms and concepts that help you access DIF files with SAS/ACCESS software. Then it describes DIF-specific statements you use in the ACCESS and DBLOAD procedures. Finally, it contains a section on how the SAS/ACCESS interface works.

Data Interchange Format (DIF) files are used by SAS/ACCESS software to access data indirectly from other software products, such as data in Lotus 1-2-3 spreadsheets and database files.

---

### DIF Files

Various software products such as Lotus 1-2-3 enable you to use spreadsheet or database files to enter, organize, and perform calculations on data. Spreadsheets are most often used for general ledgers, income statements, and other types of financial record keeping. Database files also enable you to organize related information, such as, the data in an accounts-receivable journal.

In both spreadsheets and database files, the data are organized according to certain relationships among data items. These relationships are expressed by files in a tabular form, that is, in columns and rows. DIF files allow both character and numeric data in the same column. See “DIF File Data Types” on page 109 in this chapter for more information. Each *row* can hold one data value for each column. The spreadsheet and database files can be translated to DIF files that the SAS/ACCESS interface can process.

A spreadsheet consists of columns and rows, and their intersection is called a *cell*. Figure 8.1 on page 109 illustrates four columns from the spreadsheet CUSTOMERS and highlights a column and a row.

**Figure 8.1** Columns and Rows of Data in a DIF File

	column			
CUSTOMER	CITY	STATE	COUNTRY	
14324742	San Jose	CA	USA	
14569877	Memphis	TN	USA	row
14898029	Rockville	MD	USA	
26422096	La Rochelle		France	
38763919	Buenos Aires		Argentina	
46783280	Singapore		Singapore	

## DIF File Naming Conventions

DIF filenames must follow operating-environment specific conventions, so check the documentation that comes with your application or operating system software for further information.

- Under Windows 95, Windows 98, Windows NT, and OS/2, the ACCESS and DBLOAD procedures support long names that are specified in the PATH= statement (such as `path='c:\sasdemo\library\new_customers_1999.dif'`);). However, some applications that support DIF files might not accept files with long names.
- Filenames start with a letter, and they can contain any combination of the letters A through Z, the digits 0 through 9, and the underscore (\_).

---

## DIF File Data Types

Every column in a DIF file has a name and one or two data types. A DIF file allows columns that include both character and numeric data.

*Character data* are generally considered text and can include dates and numbers if prefixes are used to indicate character data and to align the data in the cell. For example, in Lotus 1-2-3, the value "**110 Maple Street**" uses the double quote prefix and aligns the label on the right side of the cell.

*Numeric data* include numbers (0 through 9), formulas, and cell entries that begin with one of the following symbols: +, \$, @, -, or #. When you create and load a DIF file with PROC DBLOAD, the SAS/ACCESS engine supplies **NA** for a missing, numeric value. For decimal numbers, the SAS/ACCESS engine queries the operating environment for the current setting of the decimal separator and uses it when reading or creating DIF files.

Numeric data also include *date and time values*. In Lotus software, a date value is the integer portion of a number that can range from 01 January 1900 to 31 December 2099, that is, 1 to 73050. A Lotus software time value is the decimal portion of a number that represents time as a proportion of a day. For example, 0.0 is midnight, 0.5 is noon, and 0.999988 is 23:59:59 (on a 24-hour clock). While a number can have both a date and a time portion, the formats in Lotus 1-2-3 display a number only in a date format or a time format. For information about how the SAS/ACCESS interface handles date and time values and formats, see "How the SAS System Handles Date and Time Values" on page 110.

When you create an access descriptor, the interface software determines the column type by the value in the first row of data (excluding any rows that are defined for column names, blank rows for readability, and so on). If the first row in the column has no data value, the column type defaults to character data.

By default, any data value in a column that does not match the type is treated as a missing value. However, if you set the DIFNUMS environment variable to **YES** in your SAS configuration file, any numeric data values in a character column are converted to the character representation of the number and are not treated as missing values. Add the following line to your SAS configuration file to set the DIFNUMS environment variable to **YES**:

```
-SET DIFNUMS YES
```

The default for the DIFNUMS environment variable is **NO**. Refer to the SAS companion documentation for your operating system for more information about environment variables.

You can change the column type from the type determined by SAS/ACCESS software when you create an access descriptor.

---

## How the SAS System Handles Date and Time Values

As an introduction to how the SAS System handles date and time values in DIF files, this section describes the date and time values in DIF files created from Lotus.

Lotus date and time datetime values and formats are described earlier in this chapter in "DIF File Data Types" on page 109.\* The following sections describe how the ACCESS and DBLOAD procedures handle the conversions between the SAS values and the Lotus values and their respective formats.

### Datetime Conversions in the ACCESS Procedure

When you create an access descriptor, the SAS System cannot distinguish a Lotus datetime value from other numeric data. The SAS System stores the Lotus datetime value as a number and displays it like other Lotus numeric data by using the SAS variable format 15.2 (the default format for this interface).

To convert a Lotus datetime value to a SAS datetime value, you must specify a SAS datetime format in the access descriptor. A Lotus datetime value is a number that represents the number of days between January 1, 1900, and a specified date; changing the default SAS format (15.2) to a datetime format in the descriptor causes the Lotus value to be converted to an equivalent SAS datetime value based on January 1, 1960. In other words, the Lotus numeric value for January 1, 1960 (which is 21916) is converted to the equivalent SAS representation of January 1, 1960 (which is 0) only if a SAS datetime format is stored in the descriptor for that column. Otherwise, the Lotus value of 21916 is treated as a SAS numeric value of 21916.

The following table shows how the SAS System uses a Lotus datetime value to convert to a SAS datetime format.

---

\* In this description, datetime (in lowercase) refers to any value or format that represents a date, a time, or both a date and time.

**Table 8.1** Value-to-Format Conversions

For a SAS format	SAS System uses
date	integer portion of the Lotus number
time	decimal portion of the Lotus number
date-and-time	integer and decimal portion of the Lotus number

### Datetime Conversions in the DBLOAD Procedure

If a SAS variable is specified with a date, time, or datetime format in the FORMAT statement, the interface view engine converts that datetime value into the equivalent Lotus datetime value when the new DIF file is created. However, the DIF file has no way of relating this formatting information to Lotus products. Therefore, when you load the DIF file into a 1-2-3 worksheet, the datetime values are represented as numbers. You should assign (from within Lotus) a Lotus datetime format to any datetime column that you load from a DIF file.

If a SAS variable represents a date, time, or datetime value, but it has not been assigned a SAS datetime format—the SAS datetime value is represented as a number—the number is *not converted* into an equivalent Lotus datetime value in the DIF file. Rather, the number is written to the new DIF file as is. Because SAS dates are based on January 1, 1960, and Lotus dates are based on January 1, 1900, if you assign a Lotus datetime format to an unconverted Lotus column, the datetime values in that column are inaccurate.

To maintain a SAS variable format in the input data set, yet change it only while the DBLOAD procedure is in progress, use the DBLOAD FORMAT statement. This statement enables you to assign a temporary format to a SAS variable for the duration of the procedure without affecting how the SAS System stores the variable.

For example, if the SAS format for the BIRTHDAT variable in the MYDATA.SASEMPS data set is left at the default 15.2 format, you can specify the FORMAT statement to change the variable's format to DATE7. while you are creating and loading the DIF file. When you load the DIF file into a 1-2-3 worksheet, you can specify an equivalent Lotus date format. When the DBLOAD procedure has completed, the SAS format for the BIRTHDAT variable reverts to the 15.2 format.

You can specify the FORMAT statement in the PROC DBLOAD statement when you invoke the procedure using any of the methods of processing.

*Note:* There are certain display restrictions on the SAS datetime values that are loaded into Lotus 1-2-3 worksheets through DIF files. If you load a SAS variable with a DATETIME $w.d$  format into a DIF file, Lotus stores the number with both integer and decimal portions. However, when you load the DIF file into a 1-2-3 worksheet and specify a format for the column, you can only specify a date format (that uses the integer portion) or time format (that uses the decimal portion) for that column, not both at the same time.  $\Delta$

---

## ACCESS Procedure Data Conversions

The table below shows the default SAS System variable formats that the ACCESS procedure assigns to each type of DIF file data. DIF file numeric data include date and time values. See “How the SAS System Handles Date and Time Values” on page 110 for more information.

**Table 8.2** Default SAS System Variable Formats for DIF File Data

DIF File Data	SAS Variable Format
C (Character)	\$20.
N (Numeric)	15.2

If DIF file data fall outside of the valid SAS data ranges, you get an error message in the SAS log when you try to read the data.

---

## ACCESS Procedure: DIF Specifics

Chapter 2, “ACCESS Procedure Reference,” on page 11 describes the generic options and procedure statements that enable you to create access descriptors, view descriptors, and SAS data files from PC file format data. The following section describes the PC file-specific statements you use in the SAS/ACCESS interface to DIF files.

---

### ACCESS Procedure Statements for DIF Files

To create an access descriptor, you use the DBMS=DIF option and the database-description statements PATH=, DIFLABEL, and SKIPROWS. These statements supply DIF-specific information to the SAS System, and must immediately follow the CREATE statement. In addition to the database-description statements, you can use optional editing statements when you create an access descriptor. These editing statements must follow the database-description statements.

Database-description statements are only required when you create access descriptors. Because the DIF information is stored in an access descriptor, you do not need to repeat this information when you create view descriptors.

The SAS/ACCESS interface to DIF uses the following procedure statements:

**PROC ACCESS** *options*;

**CREATE** < libref.>member-name.ACCESS | VIEW;

**UPDATE** < libref.>member-name.ACCESS | VIEW;

**PATH=** 'path-and-filename<.DIF>' | '<'>filename<'>' | fileref;

**DIFLABEL**;

**SKIPROWS** <=> number-of-rows-to-skip;

**ASSIGN** | AN <=> YES | NO;

**DROP** '<'>column-identifier-1<'><...<'>column-identifier-n<'>>;

**FORMAT** '<'>column-identifier-1<'><=> SAS-format-name-1  
<...<'>column-identifier-n<'><=> SAS-format-name-n>;

**LIST** <ALL | VIEW | '<'>column-identifier<'>>;

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

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

**SELECT** ALL | '<'>column-identifier-1<'> <...<'>column-identifier-n<'>>;

**SUBSET** *selection criteria* ;

**TYPE**  $\langle \rangle$  *column-identifier-1*  $\langle \Rightarrow \rangle$  C | N  
 $\langle \dots$  *column-identifier-n*  $\langle \Rightarrow \rangle$  C | N  $\rangle$ ;

**UNIQUE**  $\langle \Rightarrow \rangle$  YES | NO;

**RUN**;

The QUIT statement is also available in PROC ACCESS. However, its use causes the procedure to terminate. QUIT is used most often in the interactive line and noninteractive modes to exit the procedure without exiting SAS.

**DIFLABEL**

indicates whether variable names are generated from the columns in the first row. If you omit this statement, variable names that are generated are based on the columns' placement in the first row. That is, the SAS System labels each column as **COL0**, **COL1**, **COL2**, and so on. These labels become the names of SAS variables in the access descriptor.

If you specify DIFLABEL, the ACCESS procedure reads column labels from the first row of the DIF file and uses them as the SAS variable names in the access descriptor. You provide the DIF file column labels; they are not the letters (for example, A, B, and so on) that identify the columns in a worksheet. If you specify DIFLABEL, the SKIPROWS statement automatically changes to 1.

Always specify DIFLABEL after the PATH= statement and before any editing statements. When you update a descriptor, you are not allowed to specify the DIFLABEL statement.

The following example creates an access descriptor and a view descriptor based on .DIF file data.

```
options linesize=80;
libname difdliba 'SAS-data-library';
libname diflibv 'SAS-data-library';

proc access dbms=dif;
/* create access descriptor */
  create difliba.custs.access;
  path='c:\difiles\dbcusts.dif';
  diflabel;
  skiprows=2;
  assign=yes;
  rename customer = custnum;
  format firstorder date9.;
  list all;

/* create usacust view      */
  create diflibv.usacust.view;
  select customer state zipcode name
         firstorder;
run;
```

**SKIPROWS**  $\langle \Rightarrow \rangle$  *number-of-rows-to-skip*;

specifies the number of rows, beginning at the top of the DIF file, to ignore when reading data from the file. The default value for SKIPROWS is 0. The skipped (or ignored) rows often contain information such as column labels or names or underscores rather than input data.

If you specify the DIFLABEL statement, the default value of SKIPROWS automatically changes to 1. The SKIPROWS statement should always follow the PATH= statement and precede any editing statements when you are creating a

descriptor. The first row of data after SKIPROWS is used to generate the SAS variable types and formats. If there are no data in the first row of a column after SKIPROWS, the data in the rest of the column are assumed to be character, even if the data in the next row are numeric.

By default, any data value in a column that does not match the type is treated as a missing value. However, if you set the DIFNUMS environment variable to **YES** in your SAS configuration file, any numeric data values in a character column are converted to the character representation of the number and are not treated as missing values. Add the following line to your SAS configuration file to set the DIFNUMS environment variable to **YES**:

```
-SET DIFNUMS YES
```

The default for the DIFNUMS environment variable is **NO**. Refer to the SAS documentation for your operating environment for more information about environment variables.

You can change the column type from the type determined by SAS/ACCESS software when you create an access descriptor.

**TYPE** *<'>column-identifier-1 <'> <=> C | N < . . .column-identifier-n <=> C | N*>;  
changes the expected data types of SAS variables. SAS data sets have two data types: character (C) and numeric (N). Spreadsheet files have the same two data types: character (for labels and formula strings) and numeric (for numbers and formulas). Changing the default data type of a SAS variable in a descriptor file also changes its associated default format in the loaded file.

If you omit the TYPE statement, the database field types are generated from the PC file data types. You can change as many database field types as you want in one TYPE statement.

---

## DBLOAD Procedure: DIF Specifics

Chapter 4, “DBLOAD Procedure Reference,” on page 41 describes the generic options and procedure statements that enable you to create a PC data file. The following section describes the file-specific statements you use in the SAS/ACCESS interface to DIF files.

---

### DBLOAD Procedure Statements for DIF Files

To create and load a DIF table, the SAS/ACCESS interface to PC file formats uses the following statements.

*Note:* The SAS/ACCESS interface to DIF files does not use the TYPE statement described in Chapter 4, “DBLOAD Procedure Reference,” on page 41.  $\Delta$

```
PROC DBLOAD DBMS=DIF <DATA=< libref.>SAS-data-set>;
  PATH='path-and-filename<.DIF>' | <'>filename<'> | fileref;
  DIFLABEL;
  ACCDESC=< libref.>access-descriptor;
  DELETE variable-identifier-1 <...variable-identifier-n>;
  ERRLIMIT=error-limit;
  FORMAT SAS-variable-name-1 SAS-format-1 <...SAS-variable-name-n
    SAS-format-n>;
  LABEL;
  LIMIT=load-limit;
```



```

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

RUN;

```

The QUIT statement is also available in PROC DBLOAD. However, its use causes the procedure to terminate. QUIT is used most often in the interactive line and noninteractive modes to exit the procedure without exiting SAS.

#### DIFLABEL

writes column labels to the first row of the new DIF file and follows the column labels with a blank row. The column labels can be default SAS variable names or, if you specify the LABEL statement, SAS labels. You can modify the column labels using the RENAME statement.

If this statement is omitted, data are read from the data set and written to the DIF file beginning in the first row of the DIF file, and no column labels are written to the file.

```

FORMAT SAS-variable-name-1 SAS-format-1< ...SAS-variable-name-n
sas-format-n>;

```

assigns a temporary format to a SAS variable in the input SAS data set. This format temporarily overrides any other format for the variable. The assignment lasts only for the duration of the procedure. Assign formats to as many variables as you want in one FORMAT statement.

Use FORMAT when you want to change the format, column width, or the number of decimal digits for columns being loaded into the PC file. For example, if you change the SAS variable format 12.1 to DOLLAR15.2, the column format of the loaded data changes from a fixed numeric format with a column width of 12 and one decimal digit to a currency format with a column width of 15 and two decimal digits.

The following example creates a new DIF table, EXCHANGE.DIF, from the data file DLIB.RATEOFEX. An access descriptor ADLIB.EXCHANGE is also created, based on the new DIF table. You must be granted the appropriate privileges in order to create new DIF files.

```

libname difdliba 'SAS-data-library';
libname diflibv 'SAS-data-library';

proc dbload dbms=dif data=dlib.rateofex;
  accdesc=adlib.exchange;
  path='c:\difiles\sasdemo\exchange.dif';
  diflabel;
  rename fgnindol=fgnindolar 4=dolrsinfgn;
  load;
run;

```

---

## How the SAS/ACCESS Interface Works

The SAS/ACCESS interface to DIF files accesses data in spreadsheets and databases indirectly. Spreadsheet and database data must be translated into a DIF file format

before it can be read by the SAS System. A DIF file is an ASCII text file with a file header section and a data section. DIF files, not spreadsheets or databases, are specified in the ACCESS and DBLOAD procedures. You use your software products' utilities to translate your spreadsheets and databases into DIF files. For example, you can use the Lotus 1-2-3 Translate Utility to translate a 1-2-3 worksheet or database to a DIF file. Once your spreadsheet or database is translated into a DIF file, the file is stored in a directory that you specify. You then enter this path and DIF filename with the PATH= statement in the ACCESS procedure.

If you change a spreadsheet or database file after translating the file to DIF format, re-translate the modified file and save it in a new DIF file. If you do not, the DIF file and SAS/ACCESS view based on the DIF file will not reflect your changes to the original.

The SAS/ACCESS interface to DIF files is read-only: it cannot be used to modify a DIF file.

To sort data in a DIF file, you must first extract the data into a data file. You can do this in one step with the SORT procedure's OUT= option. Or you can use the SQL procedure's SELECT statement with an ORDER BY clause.

The DBLOAD procedure translates a SAS data set into a DIF file format and stores the DIF file in the path specified by the PATH= statement. Software products such as Lotus 1-2-3 can then read data from the DIF file.

When you use a view descriptor to a DIF file in a DATA step or procedure, you provide a path to the DIF file. The DIF file interface view engine retrieves data from this file.

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

**SAS/ACCESS<sup>®</sup> Software for PC File Formats: Reference, Version 8**

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

ISBN 1-58025-544-2

All rights reserved. Produced 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 and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions 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<sup>®</sup> 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. <sup>®</sup> 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.