

## TS589A: Using PC SAS/ACCESS Software to Read and Write to the DBF File Format

*There are 4 methods that allow you to read DBF files into SAS*

### Method I: Use SAS/ACCESS to PC File Formats

Base SAS and SAS/Access Interface to PC File Formats must be licensed and installed at your site. To verify that your site license includes these products, submit the following code and check the list of products that is returned to your SAS Log:

```
proc setinit noalias; run;
```

To verify that the product is installed, check your SAS installation for the following file:

```
sasimode.dll.
```

\*\*\*\*\*

Note: SAS/Access Interface to PC File Formats cannot read a Visual FoxPro DBF file. From within Visual Foxpro, the file must be saved as a Foxpro version 2.x dbf file.

\*\*\*\*\*

Use the following code to read a DBF file into a SAS data set:

```
proc import out= work.test datafile= "c:\my documents\houses.dbf"  
  dbms=dbf replace;  
  getdeleted=no;  
run;
```

Use the Import Wizard to read a DBF file into a SAS data set:

1. Choose File>Import Data.
2. Select DBF from the list of file types.
3. Press NEXT.
4. Specify the full path to the DBF file that you are trying to import into a SAS data set.
5. Press NEXT.
6. Pick the Library where the newly-created data set is to be saved.  
Note: If library you want to use is not listed, you have to assign the LIBNAME statement to create the library before you use the Import Wizard.
7. Name the data set by using the list of data sets that resides in that library to overwrite an existing SAS data set, or provide a new data set name.
8. Specify a path/filename for saving the Wizard-generated PROC IMPORT code.
9. Choose FINISH.

### Method II: Use PROC DBF (this procedure is included in SAS/ACCESS Software for PC File Formats)

Note: This is an older V6 SAS procedure.

Use the following code to read a DBF file into SAS.

```
filename houses 'c:\houses.dbf';
/* dbn refers to the version of dbf file you are importing. */
proc dbf db4=houses out=sasuser.houses;
run;
```

### **Method III: Use SAS/ACCESS Interface to ODBC to access a DBF file via the appropriate vendor-supplied DBF ODBC driver**

To use SAS/ACCESS to ODBC, BASE SAS and SAS/ACCESS Interface to ODBC must be licensed and installed at your site. A DBF ODBC driver must be installed and configured on your pc.

Use the following code to read the DBF file into SAS:

```
libname mylib odbc dsn=yourdsn;
proc print data=mylib.dbftable; run;
```

OR

```
proc sql;
connect to odbc (prompt);
  create table sasuser.test as select * from connection to odbc (select *
  from tablename);
disconnect from odbc;
quit;
```

Notes:

- 'Prompt' activates a dialog box where you can choose your data source name. Alternatively, you can use 'DSN=' in the parenthesis and specify your data source name.
- 'sasuser.TEST' is the name of the SAS data set that you create from this query.
- 'tablename' refers to the name of the DBF file.

### **Method IV: Use SAS/ACCESS Interface to OLEDB to access the DBF table via the vendor-supplied OLEDB provider**

BASE SAS and SAS/ACCESS Interface to OLEDB must be licensed and installed at your site. An OLE DB Provider that supports DBF must be installed on your PC.

Use the following code to read the DBF file into SAS:

```
libname mylib oledb;
%put &sysdbmg;
```

This is a prompted connection—choose your OLEDB Provider and your source dbf file. After the LIBNAME is assigned, you can reference the table with the LIBNAME, for example

```
proc print data=mylib.dbftable; run;
```

OR

```
proc sql;
connect to oledb(prompt);
create table sasuser.test as select * from connection to oledb
    (select * from tablename);
disconnect from oledb;
quit;
```

***There are 4 methods that allow you to write your SAS data sets to a DBF file.***

### **Method I: Use SAS/ACCESS Interface to PC File Formats**

Base SAS and SAS/ACCESS Interface to PC File Formats must be licensed and installed on your PC.

To write a SAS data set out to a DBF file, you can use PROC EXPORT code, similar to the following:

```
proc export data= sashelp.class outfile= "c:\my documents\class.dbf"
    dbms=dbf replace;
run;
```

To write a SAS data set out to a DBF file, you can also use the Export Wizard. From the File-Export Data menu:

1. Invoke the Export Wizard by choosing File-Export Data.
2. Choose your source SAS data set and select Next.
3. From the drop-down list in the field for Standard Data Source, select dBASE file (\*.dbf) and select Next.
4. Type in the full path/filename of the file you want to create, or browse to the drive/directory where you want to save this new file and enter the filename and extension, select Save, and select Next.
5. Specify a path/filename for the saved PROC EXPORT code and select Finish.

OR

You can use PROC DBLOAD:

```
proc dbload dbms=dbf data=sashelp.class;
path="c:\my documents\my new file.dbf";
version=3; /* optional-default is 5 */
label; /* optional-writes var labels as field names */
reset all; /* optional but required with label stmt */
limit=0;
load;
run;
```

## **Method II: Use PROC DBF (this procedure is included in SAS/ACCESS Software for PC File Formats)**

Note: is an older V6 SAS procedure.

```
/* dbn refers to the version of dbf file you are creating. */
filename class 'c:\class.dbf';
proc dbf dbf5=class data=sashelp.class;
run;
```

## **Method III: Use SAS/ACCESS Interface to ODBC to access a DBF file via the appropriate vendor-supplied DBF ODBC driver**

SAS/BASE and SAS/ACCESS Interface to ODBC must be licensed and installed at your site. A DBF ODBC driver must be installed and configured on your pc.

To write SAS data to the DBF file format, if you have used the LIBNAME ODBC engine statement, you can use SAS DATA step or PROC SQL statements. For example, see the following:

```
libname mylib odbc dsn="dbf datasource";
data mylib.new;
set sasuser.data set;
run;
```

OR

```
proc sql;
create table mylib.new as select * from sasuser.data set;
quit;
```

This product also supports use of the DBLOAD procedure:

```
proc dbload dbms=dbf data=sasuser.data set;
path="c:\my documents\newfile.dbf";
version=3; /* optional-default is v */
label;
reset all;
limit=0;
load;
run;
```

## **Method IV: Use SAS/ACCESS Interface to OLEDB to access the DBF table via the vendor-supplied OLEDB provider**

SAS/BASE and SAS/ACCESS Interface to OLEDB must be licensed and installed at your site. An OLE DB Provider that supports DBF must be installed on your pc.

If you use OLEDB to write back to the DBF file format, you must use the LIBNAME OLEDB engine statement and either a SAS DATA step or PROC SQL statements (the DBLoad procedure is not supported by this product):

```
libname mylib oledb;  
data mylib.new;  
set sasuser.data set;  
run;
```

OR

```
proc sql;  
create table mylib.new as select * from sasuser.data set;  
quit;
```

**Important note:** If you *cannot perform* any reads or writes because you do not have the SAS/ACCESS products installed on your PC, you can save the DBF file as a delimited text file (that is, csv, or txt) and then use the Import Wizard to read that file into SAS, or you can use Dynamic Data Exchange (DDE). Both of these methods are supported by Base SAS. See the "SAS Companion for the Microsoft Windows Environment" or the Tech Support document TS-325 for more examples for using DDE.