## PROC EXPORT

**PROC EXPORT reads data from a SAS data set and writes it to an external data source.** External data sources can include such files as Microsoft Access Databases, Microsoft Excel Workbooks, Lotus spreadsheets, and delimited files. In delimited files, A delimiter such as a blank, comma, or tab separates columns of data values.

**Syntax:-**

```
    PROC EXPORT DATA=<LIBREF.>SAS DATA-SET <(SAS DATA-SET-OPTIONS)>
                    OUTFILE="FILENAME" | OUTTABLE="TABLENAME"
                    <DBMS=IDENTIFIER> <LABEL><REPLACE>;
    PUTNAMES=YES/NO;
    SHEET=”SHEET_NAME”;
    RUN;
```

**Examples:-**

**Data=Dataset**

Specify the dataset name which we are exporting into external files.

**Outfile=File location**

Specify the file location to where we are exporting data

**DBMS=File identifier**

Specify the DBMS (type of data) of external file.

```
PROC EXPORT DATA=SASHELP.CARS
            OUTFILE='E:\MYFILES\CLASS.XLS'
            DBMS=EXCEL;
RUN;
PROC EXPORT DATA=SASHELP.CLASS
            OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.TXT'
            DBMS=TAB;
RUN;
```

| IDENTIFIER | OUTPUT DATA SOURCE | EXTENSION | HOST AVAILABILITY |
|---|---|---|---|
| CSV | delimited file (comma-separated values) | .csv | OpenVMS, UNIX, Microsoft Windows |
| DLM | delimited file (default delimiter is a blank) | .* | OpenVMS, UNIX, Microsoft Windows |
| TAB | delimited file (tab-delimited values) | .txt | OpenVMS, UNIX, Microsoft Windows |

| IDENTIFIER | OUTPUT DATA SOURCE | EXTENSION |
|---|---|---|
| Access* | Microsoft Access 2000, 2002, 2003, or 2007 table (using the LIBNAME statement) | .mdb<br>.accdb |
| Access97 | Microsoft Access table (using the LIBNAME statement) | .mdb |
| ACCESSCS | Microsoft Access table connecting remotely through PC Files Server | .mdb<br>.accdb |
| CSV | delimited file (comma-separated values) | .csv |
| DBF | dBASE 5.0, IV, III+, and III files | .dbf |
| dbfMemo | dBASE 5.0, IV, III+, and III files with memos<br>FoxPro and VisualPro with memos | .dbf<br>.fpt |
| DLM | delimited file (default delimiter is a blank) | .* |
| DTA | Stata file | .dta |
| Excel* | Excel 97, 2000, 2002, 2003, or 2007 spreadsheet (using the LIBNAME statement) | .xls<br>.xlsb<br>.xlsx |
| Excel4 | Excel 4.0 spreadsheet (using PROC DBLOAD) | .xls |
| Excel5 | Excel 5.0 or 7.0 (95) spreadsheet (using PROC DBLOAD) | .xls |
| EXCELCS | Excel spreadsheet connecting remotely through PC FIles Server | .xls<br>.xlsb |
| JMP | JMP files | .jmp |
| Paradox | Paradox DB files | .db |
| PCFS | JMP files, Stata files, and SPSS files connecting remotely through PC FIles Server | .jmp, .dta, .sav |
| SAV | SPSS files, compressed and uncompressed binary files | .sav |
| TAB | delimited file (tab-delimited values) | .txt |
| WK1 | Lotus 1-2-3 Release 2 spreadsheet | .wk1 |
| WK3 | Lotus 1-2-3 Release 3 spreadsheet | .wk3 |
| WK4 | Lotus 1-2-3 releases 4 and 5 spreadsheet | .wk4 |
| XLS | Excel 97, 2000, 2002, or 2003 spreadsheet (using file formats) | .xls |

**Label**

Writes SAS label names as column names to the exported table. If SAS label names do not exist, then the variable names are used as column names in the exported table.

```
DATA DS;
SET SASHELP.CLASS;
LABEL SEX=GENDER;
RUN;
```

```
PROC EXPORT DATA=ds
        OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.XLS'
        DBMS=EXCEL
        LABEL;                /*Check once without label*/
RUN;
```

**Replace**

If already external file exists it replaces on that. Default is noreplace.

```
PROC EXPORT DATA=ds
        OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.XLS'
        DBMS=EXCEL
        LABEL
        REPLACE;
RUN;
```

**Putnames=Yes/No**

Default is **Yes,** It means it writes column names in external file when we export data.
But if we specify **Putnames=No,** It won't write columns in external file.

```
PROC EXPORT DATA=DS
                  OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.TXT'
                  DBMS=TAB
                  REPLACE;
PUTNAMES=NO;
RUN;
```

**Sheet="Sheet Name"**

Specify the Sheet name in excel file, in which sheet we are writing the data.
In same excel file we are exporting data into multiple sheets

```
PROC EXPORT DATA=SASHELP.CLASS
                 OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.XLS'
                 DBMS=EXCEL
                 LABEL
                 REPLACE;
SHEET="KRISHNA";
RUN;
```

```
PROC EXPORT DATA=SASHELP.CARS
            OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.XLS'
            DBMS=EXCEL
            LABEL
            REPLACE;
SHEET="STANSYS";
RUN;
```

**OUTTABLE=*table-name***

Specifies the DBMS output table. If the name does not include special characters (such as question marks), lowercase characters, or spaces, omit the quotation marks. The DBMS table name might be case sensitive.

```
PROC EXPORT OUTTABLE=CLASS
            DATA=SASHELP.CLASS
            DBMS=ACCESS
            REPLACE;
DATABASE="C:\Documents and Settings\Administrator\Desktop\New
Folder\STANSYS\HYDERABAD\SAS\SOURCE DATA\MS-ACCESS\DATA.MDB";
RUN;
```

**Exporting data into CSV file**

```
PROC EXPORT DATA=SASHELP.CLASS
            OUTFILE='E:\MYFILES\CLASS.CSV'
            DBMS=CSV
            REPLACE;
RUN;
```

**Exporting data into EXCEL file**

```
PROC EXPORT DATA=SASHELP.CLASS
            OUTFILE='E:\MYFILES\CLASS.XLS'
            DBMS=EXCEL
            REPLACE;
PUTNAMES=YES;
SHEET="SAMPLE";
RUN;
```

```
PROC EXPORT DATA=SASHELP.CLASS (WHERE= (SEX='F'))
            OUTFILE='E:\MYFILES\CLASS.XLS'
            DBMS=EXCEL
            REPLACE;
RUN;
```

## Exporting data into TAB and DLM delimiter files

```
PROC EXPORT DATA=SASHELP.CLASS
                    OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.txt'
                    DBMS=TAB;
RUN;

PROC EXPORT DATA=SASHELP.CLASS
                    OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.txt'
                    DBMS=TAB
                    REPLACE;
PUTNAMES=NO;
RUN;

PROC EXPORT DATA=SASHELP.CLASS
                    OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.txt'
                    DBMS=TAB
                    REPLACE;
PUTNAMES=NO;
DELIMITER=',';     /*Take any special char which you want assign as a delimiter*/
RUN;

PROC EXPORT DATA=SASHELP.CLASS
                    OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.txt'
                    DBMS=TAB
                    REPLACE;
PUTNAMES=NO;
DELIMITER='  ';
RUN;

PROC EXPORT DATA=SASHELP.CLASS
                    OUTFILE='E:\MYFILES\CLASS.XLS\CLASS.txt'
                    DBMS=TAB
                    REPLACE;
PUTNAMES=NO;
DELIMITER='20'X;
RUN;

PROC EXPORT DATA=SASHELP.CLASS (keep=name height)
         OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.txt'
         DBMS=dlm
         REPLACE;
PUTNAMES=NO;
DELIMITER='*';
RUN;
```

```
PROC EXPORT DATA=SASHELP.CLASS (keep=name height)
        OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.txt'
        DBMS=dlm
        REPLACE;
PUTNAMES=NO;
DELIMITER='char';
RUN;
```

```
PROC EXPORT DATA=SASHELP.CLASS
                 OUTFILE='C:\Documents and Settings\Administrator\Desktop\sas\CLASS.TXT'
                 DBMS=DLM
                 REPLACE;
                 DELIMITER='*';
RUN;
```

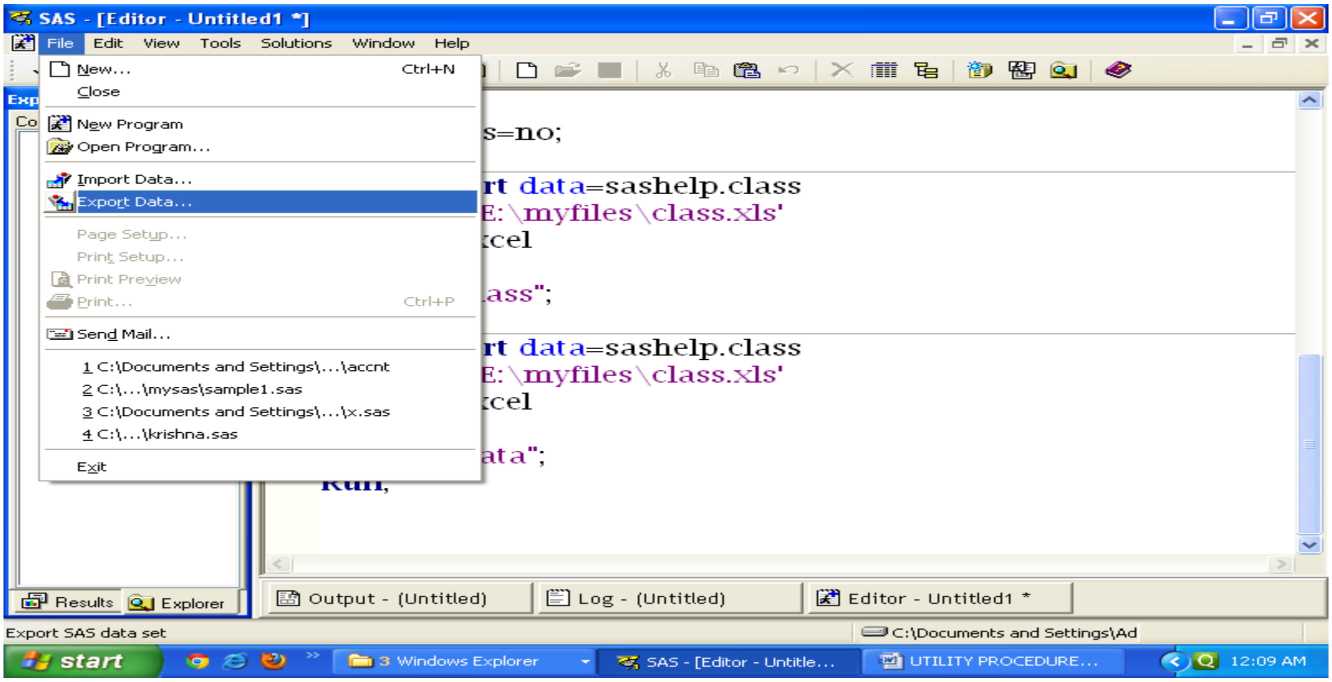
**Data step code for exporting**

```
Data _null_;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
%let _EFIREC_ = 0;    /* clear export record count macro variable */
File 'C:\Documents and Settings\Administrator\Desktop\sas\CLASS.TXT' delimiter='*' DSD
DROPOVER lrecl=32767;
If _n_ = 1 then      /* write column names or labels */ Do;
Put "Name" '*' "Sex" '*' "Age" '*' "Height" '*' "Weight”;
End;
Set SASHELP.CLASS   end=EFIEOD;
Format Name $8. ;
Format Sex $1. ;
Format Age best12. ;
Format Height best12. ;
Format Weight best12. ;
Do;
EFIOUT + 1;
Put Name $ @;
Put Sex $ @;
Put Age @;
Put Height @;
Put Weight;
;
End;
If _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
If EFIEOD then call symputx('_EFIREC_',EFIOUT);
Run;
```

We can also use the Export Wizard to guide you through the steps to export a SAS data set. The Export Wizard can generate EXPORT procedure statements, which you can save to a file for subsequent use. To open the Export Wizard, from the SAS windowing environment,

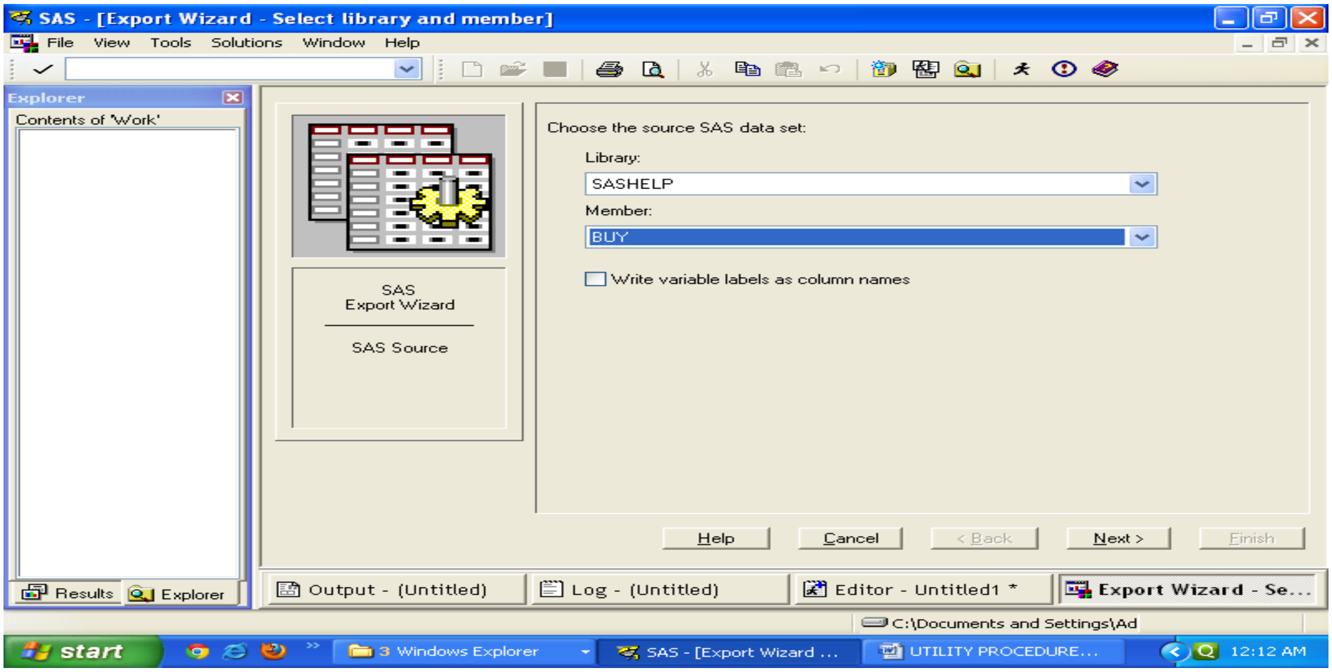
**Select File ➨ Export Data.**

**Exporting data from SAS to Excel through import wizard**



```
s=no;

rt data=sashelp.class
E:\myfiles\class.xls'
xcel

ass";

rt data=sashelp.class
E:\myfiles\class.xls'
xcel

ata";
Run;
```

**Specify the library name and dataset name which you want to export**



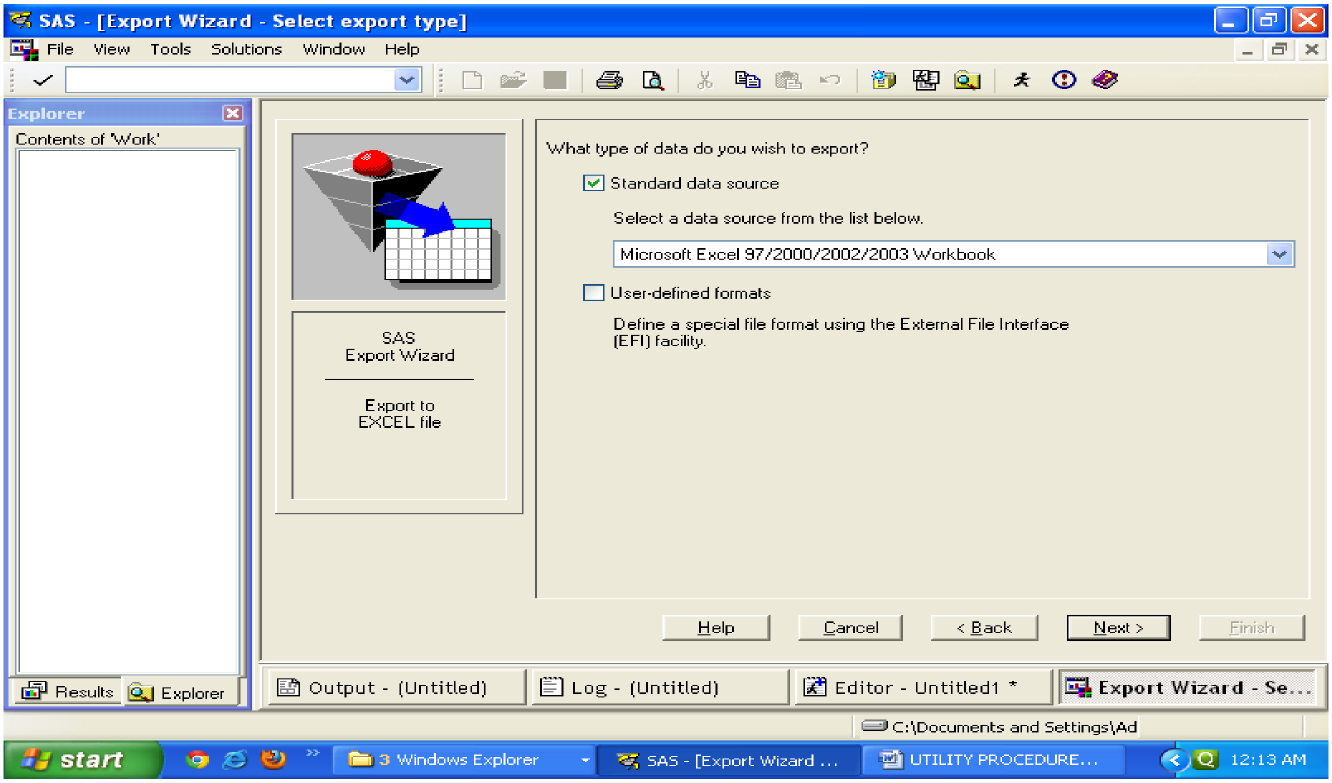
Choose the source SAS data set:

Library: SASHELP

Member: BUY

□ Write variable labels as column names

**Select format of export data like excel, csv or notepad.**



What type of data do you wish to export?

☑ Standard data source

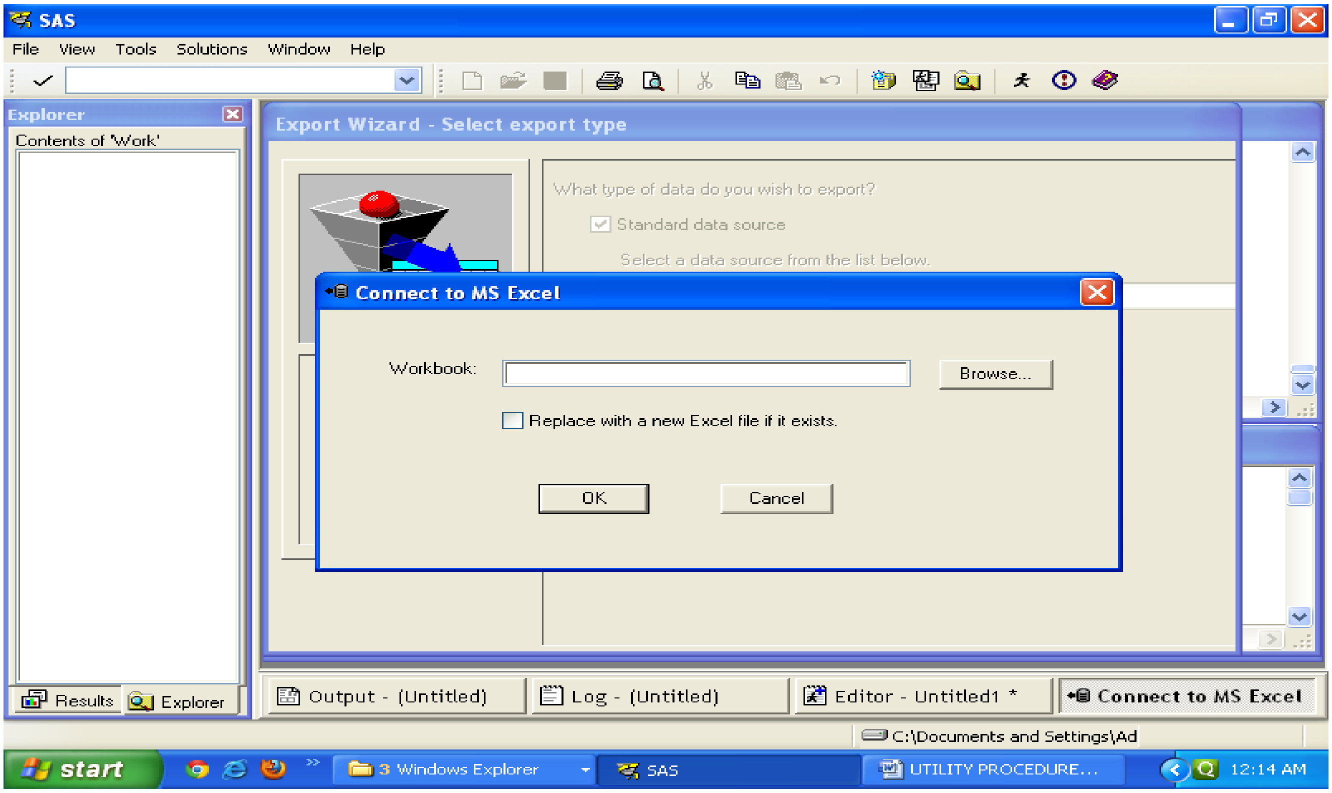
Select a data source from the list below.

Microsoft Excel 97/2000/2002/2003 Workbook

□ User-defined formats

Define a special file format using the External File Interface (EFI) facility.

## Browse the location where you want to export



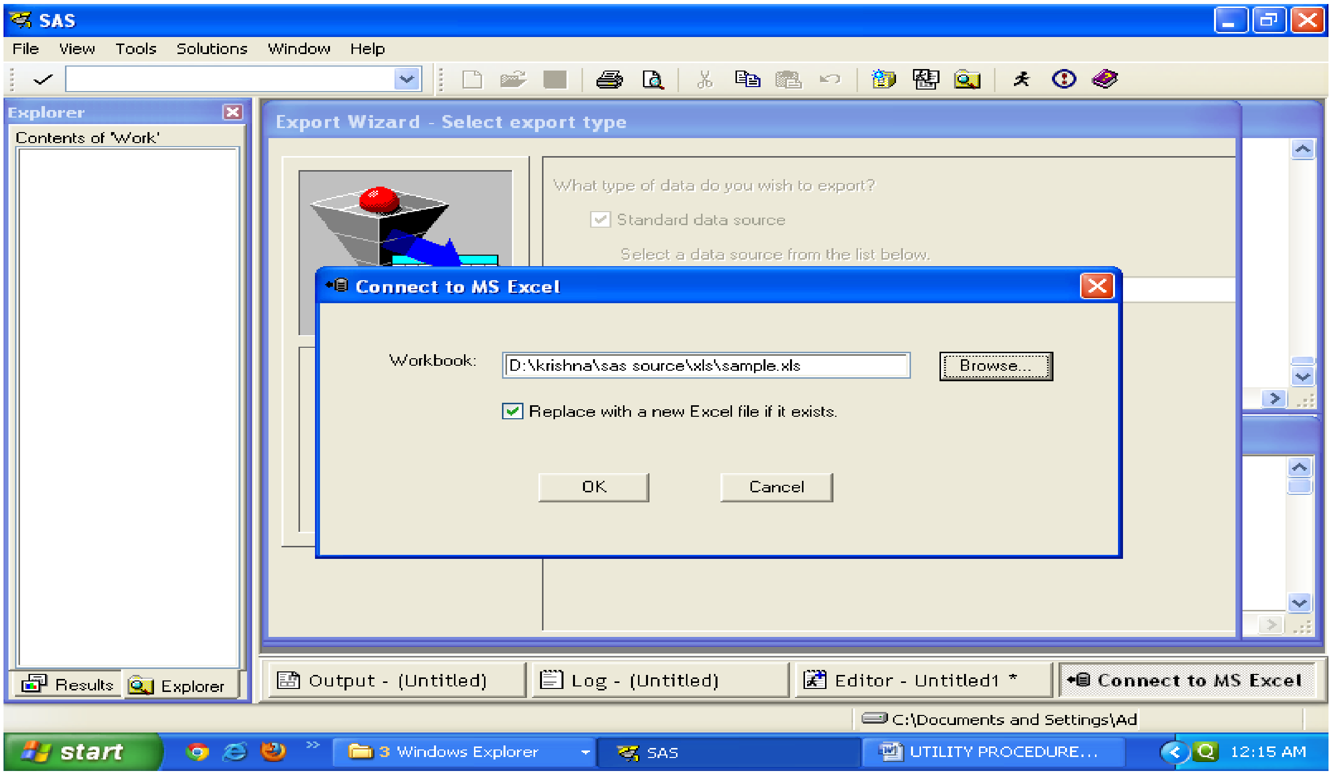
What type of data do you wish to export?

☑ Standard data source

Select a data source from the list below.

Workbook:

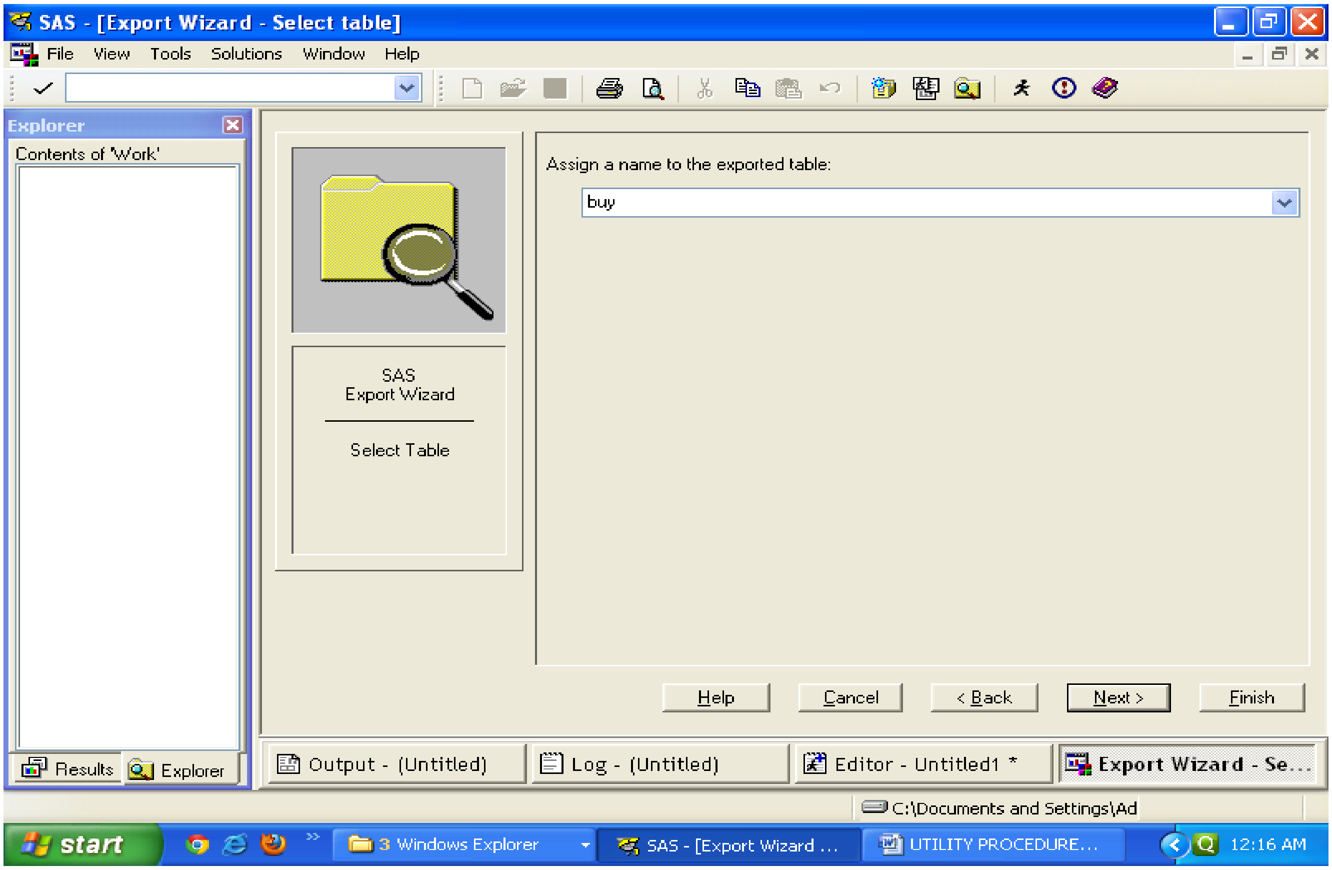
□ Replace with a new Excel file if it exists.



Workbook: D:\krishna\sas source\xls\sample.xls

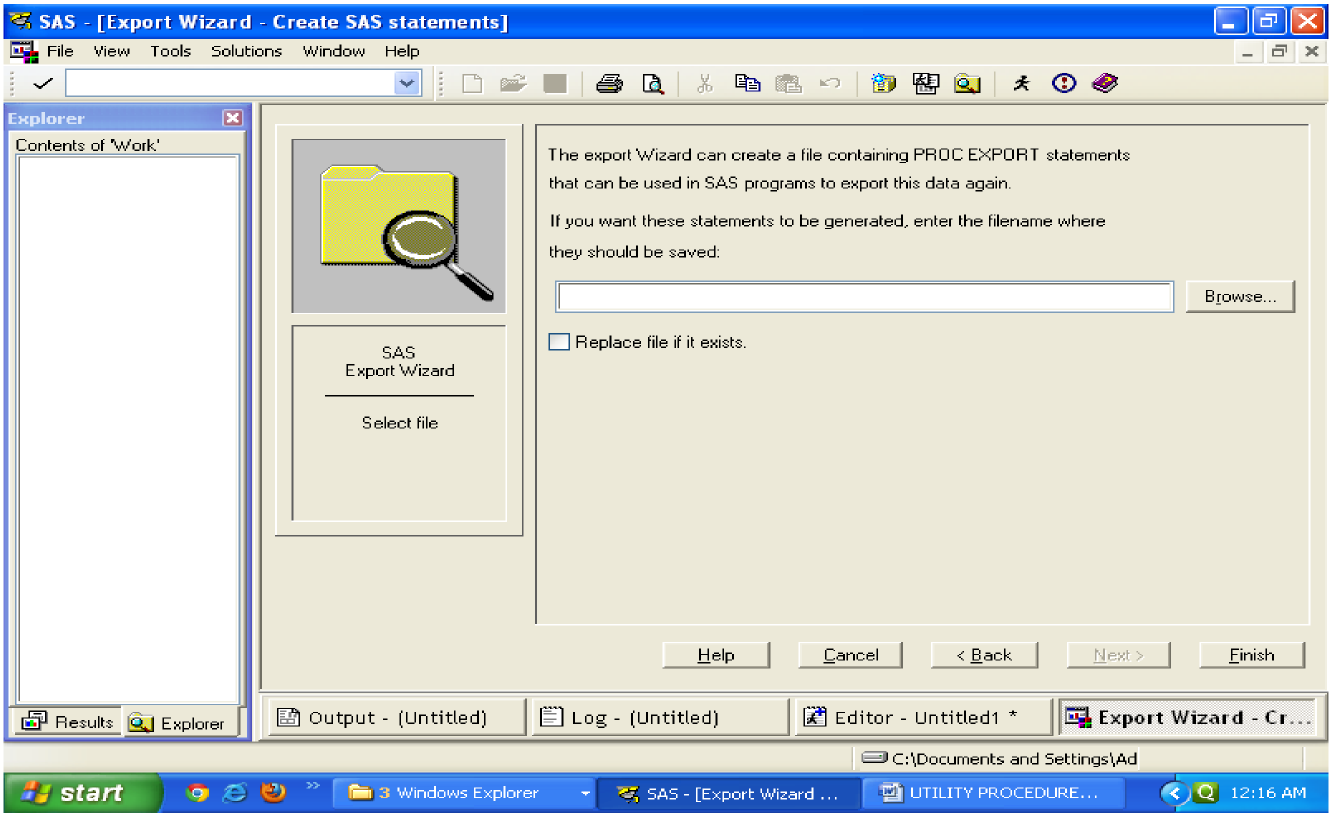
☑ Replace with a new Excel file if it exists.

**Specify a sheet name if you are exporting into excel format**



Assign a name to the exported table: buy

**Browse the location if you want to get code for exporting**



The export Wizard can create a file containing PROC EXPORT statements that can be used in SAS programs to export this data again.

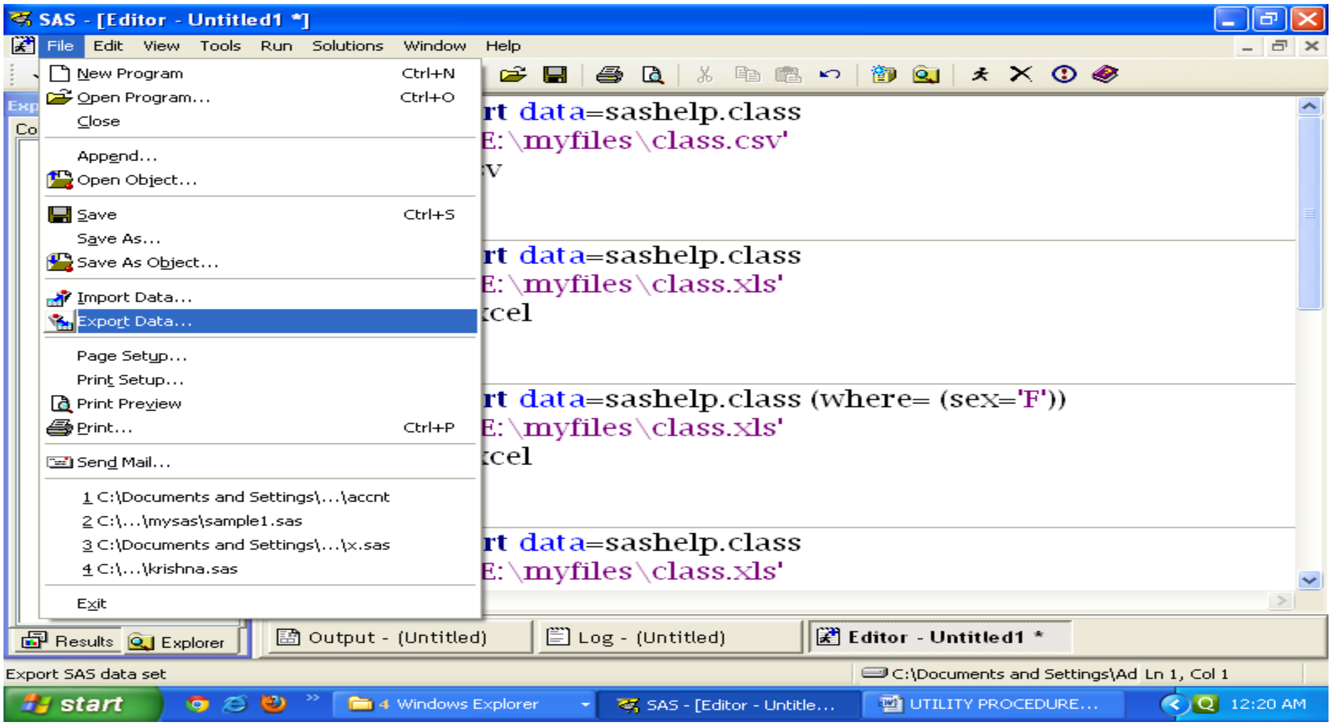
If you want these statements to be generated, enter the filename where they should be saved:

□ Replace file if it exists.

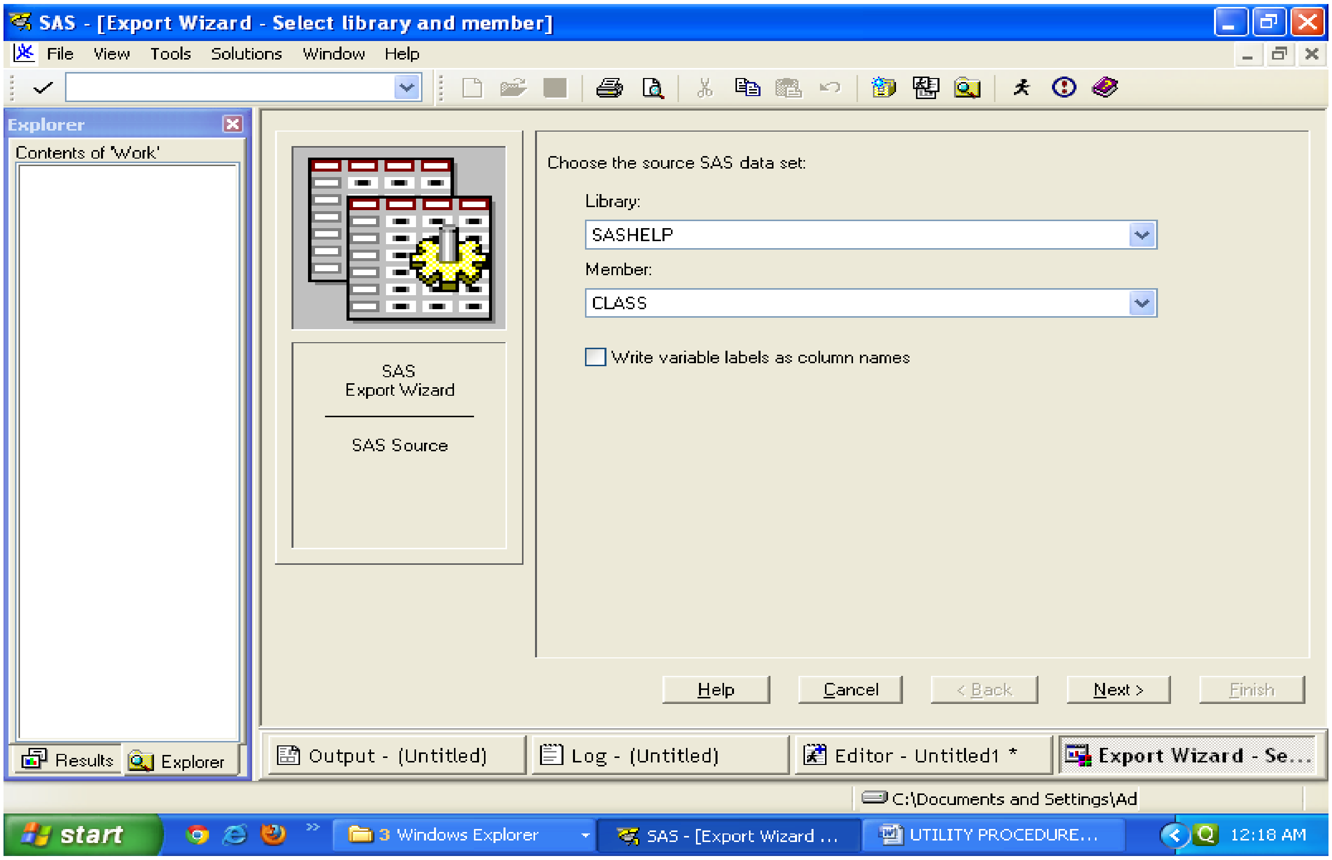
**Finally click Finish**

**CSV files also we can export in same way.**

## Exporting data from SAS to MS-ACCESS through import wizard



```
rt data=sashelp.class
E:\myfiles\class.csv'
;v

rt data=sashelp.class
E:\myfiles\class.xls'
:cel

rt data=sashelp.class (where= (sex='F'))
E:\myfiles\class.xls'
:cel

rt data=sashelp.class
E:\myfiles\class.xls'
```

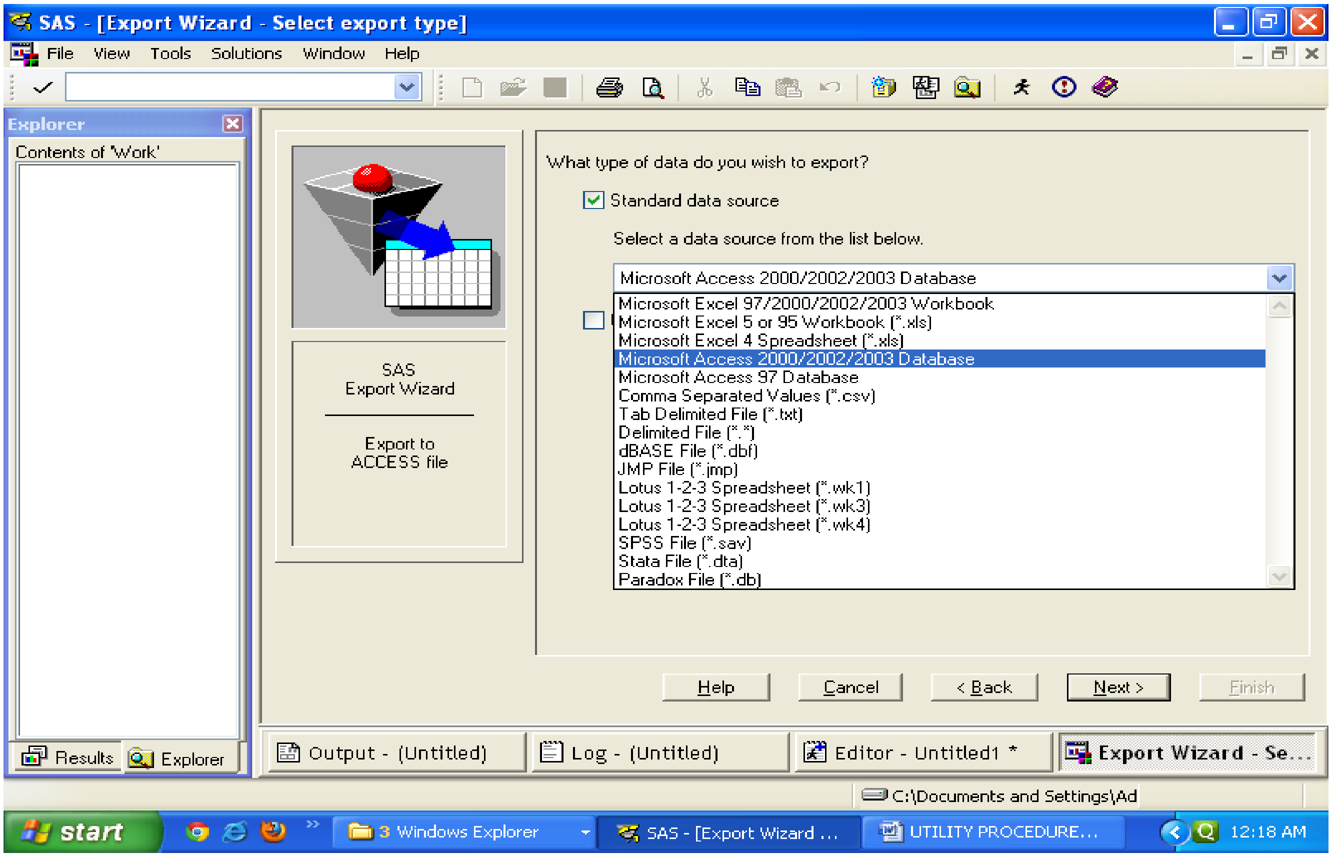


Choose the source SAS data set:

Library: SASHELP

Member: CLASS

□ Write variable labels as column names



SAS - [Export Wizard - Select export type]

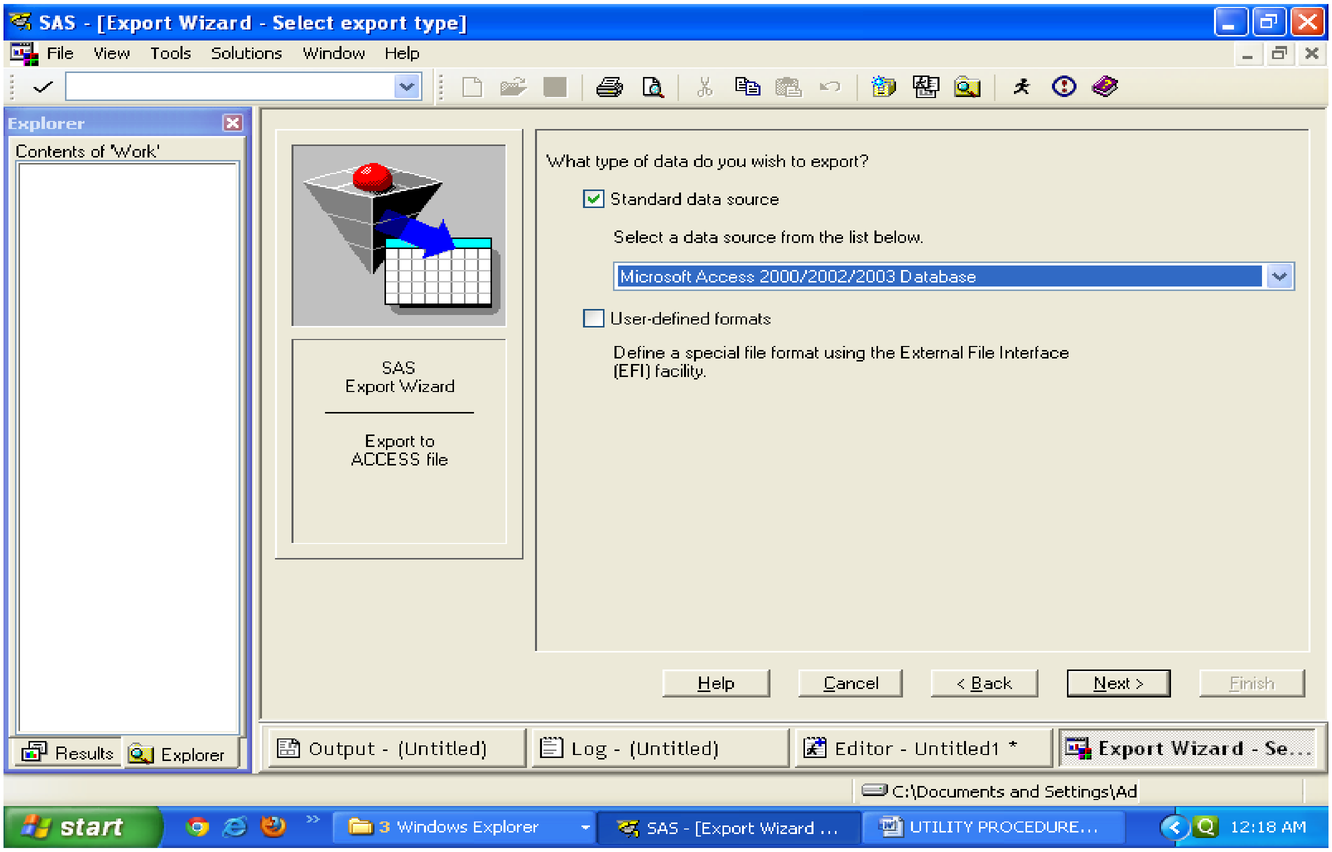
What type of data do you wish to export?

☑ Standard data source

Select a data source from the list below.

Microsoft Access 2000/2002/2003 Database

- Microsoft Excel 97/2000/2002/2003 Workbook
- Microsoft Excel 5 or 95 Workbook (*.xls)
- Microsoft Excel 4 Spreadsheet (*.xls)
- Microsoft Access 2000/2002/2003 Database
- Microsoft Access 97 Database
- Comma Separated Values (*.csv)
- Tab Delimited File (*.txt)
- Delimited File (*.*)
- dBASE File (*.dbf)
- JMP File (*.jmp)
- Lotus 1-2-3 Spreadsheet (*.wk1)
- Lotus 1-2-3 Spreadsheet (*.wk3)
- Lotus 1-2-3 Spreadsheet (*.wk4)
- SPSS File (*.sav)
- Stata File (*.dta)
- Paradox File (*.db)



SAS - [Export Wizard - Select export type]

What type of data do you wish to export?

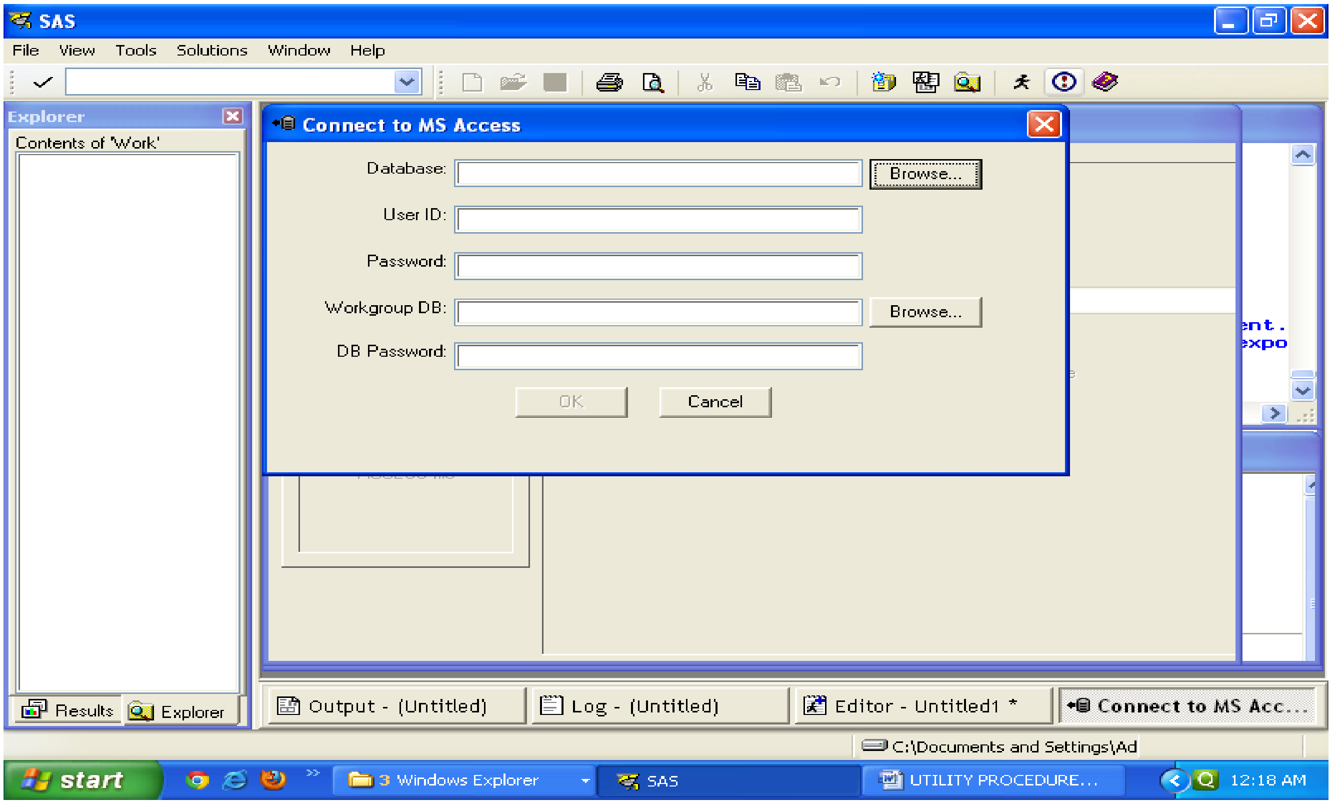
☑ Standard data source

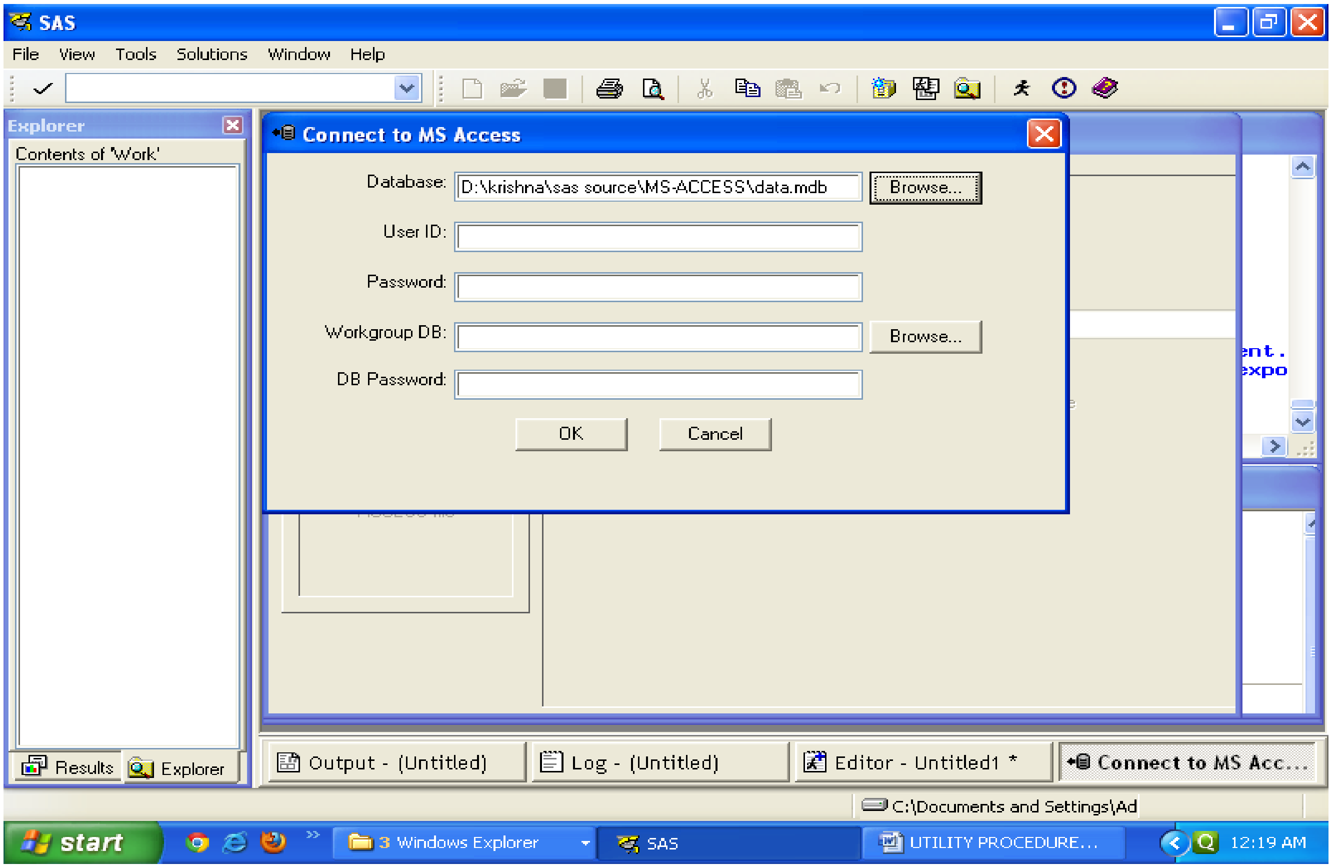
Select a data source from the list below.

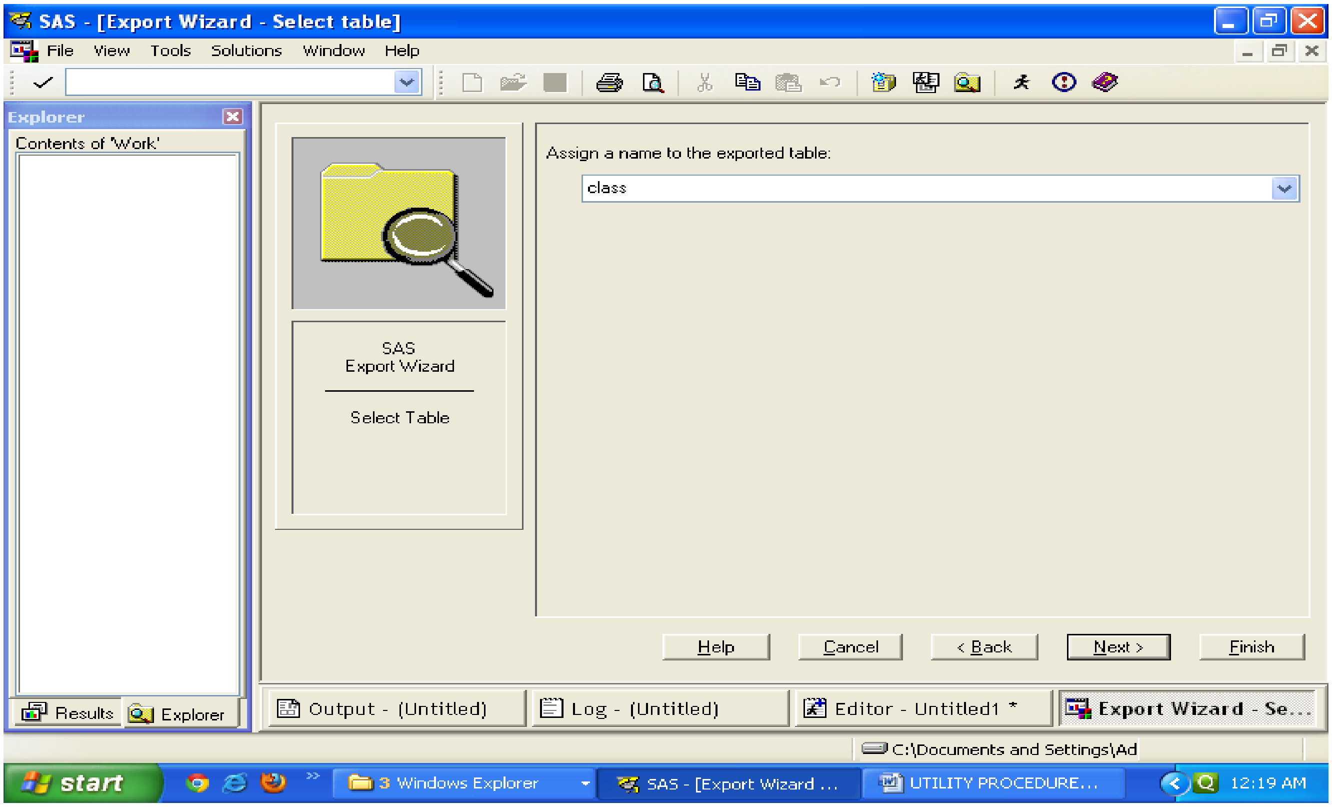
Microsoft Access 2000/2002/2003 Database

□ User-defined formats

Define a special file format using the External File Interface (EFI) facility.




Database: D:\krishna\sas source\MS-ACCESS\data.mdb



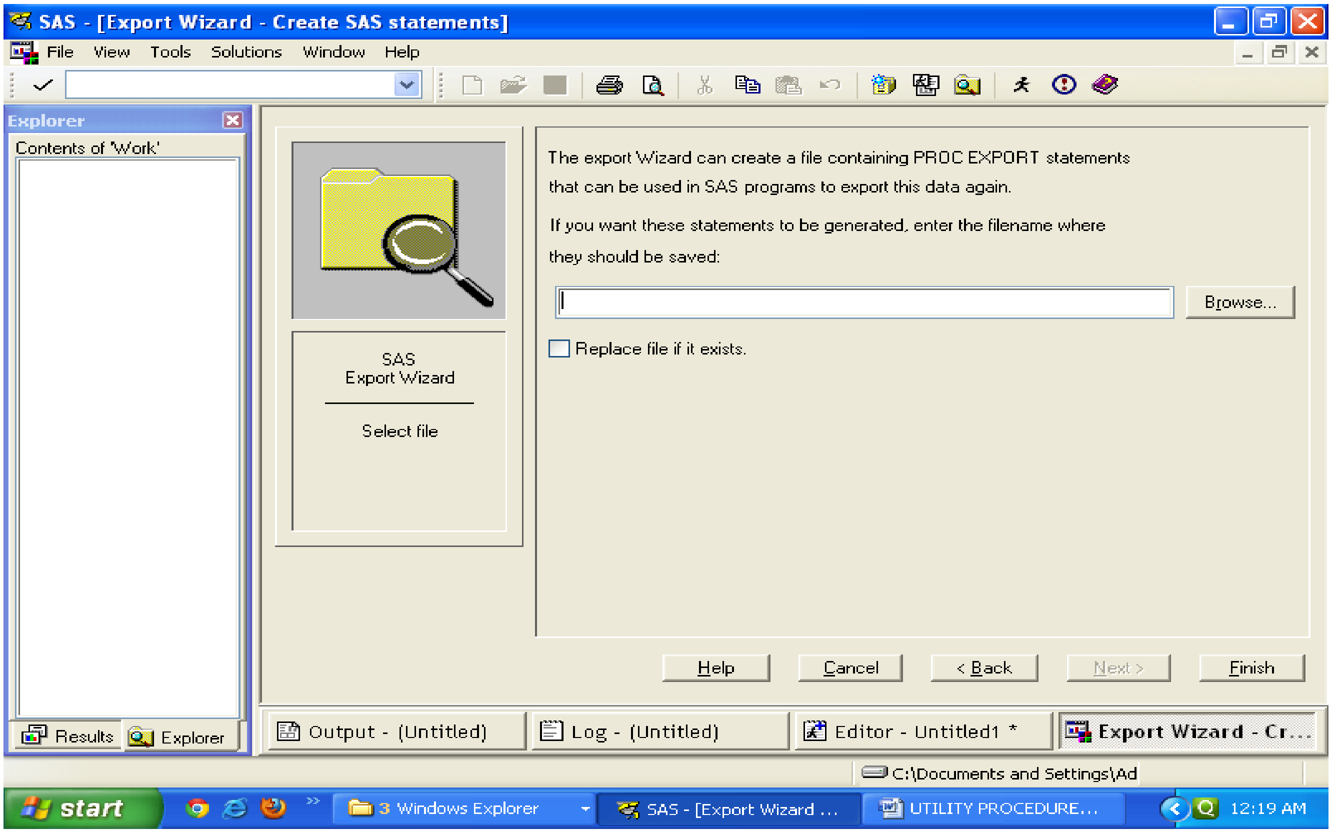
SAS - [Export Wizard - Select table]

SAS Export Wizard

Select Table

Assign a name to the exported table:

class



SAS - [Export Wizard - Create SAS statements]

SAS Export Wizard

Select file

The export Wizard can create a file containing PROC EXPORT statements that can be used in SAS programs to export this data again.

If you want these statements to be generated, enter the filename where they should be saved:

□ Replace file if it exists.

Browse location for exporting code.
Click finish.