



THE
POWER
TO KNOW.



**Using SAS[®] 9.1.3 (and later) to
Read and Write to Excel 97, 2000,
2002, and 2003 Files in a
Windows Operating Environment**

Table of Contents

Using SAS® 9.1.3 (and later) to Read and Write to Excel 97, 2000, 2002, and 2003 Files in a Windows Operating Environment.....	1
Using the SAS/ACCESS® Interface to PC Files.....	1
Reading Excel Data into SAS	1
Writing SAS Data to Excel	3
Using the SAS/ACCESS® Interface to OLE DB	3
Reading Excel Data into SAS	3
Writing SAS Data to Excel	4
Using the SAS/ACCESS® Interface to ODBC	5
Reading Excel Data into SAS	5
Writing SAS Data to Excel	6

Using SAS® 9.1.3 (and later) to Read and Write to Excel 97, 2000, 2002, and 2003 Files in a Windows Operating Environment

When using SAS in a Windows operating environment, there are three approaches to reading and writing Excel files:

1. Use the SAS/ACCESS® Interface to PC Files
2. Use the SAS/ACCESS® Interface to OLE DB
3. Use the SAS/ACCESS® Interface to ODBC

Using the SAS/ACCESS® Interface to PC Files

Reading Excel Data into SAS

The SAS/ACCESS Interface to PC Files is a product that is licensed and installed in addition to Base SAS software.

To confirm that this product is included in your SAS site license, submit the following code and check the list of products that is written to the SAS Log:

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

To verify that the product has been installed, check your **!SASroot\access\sasexe** directory for the following file:

```
sasimode.dll
```

This product provides the following methods for reading Excel data into SAS:

1. the LIBNAME – Excel Engine statement
2. the PROC SQL Pass-Through Facility
3. the Import Wizard
4. the Import procedure

Note: Excel worksheet names end with a dollar sign (\$) which is a character that is not allowed in SAS names. The worksheet name, therefore, must be specified using the Name Literal (n-literal).

Using the LIBNAME – Excel Engine Statement

```
libname mylib "c:\documents and settings\myfile.xls";

proc print data=mylib.'sheet1$'n;
run;

data work.test;
  set mylib.'sheet1$'n;
run;

proc sql;
  create table work.test as select * from mylib.'sheet1$'n;
quit;
```

Using the PROC SQL Pass-Through Facility

```
proc sql;
  connect to excel(path="c:\documents and settings\myfile.xls");
  create table work.test as
    select * from connection to excel
      (select * from [sheet1$]);
  disconnect from excel;
quit;
```

Using the Import Wizard

1. Select **File ► Import Data** to invoke the Import Wizard.
2. Select the appropriate Excel file version from the drop-down list of Standard Data Sources, and click **Next**.
3. On the Connect to Excel dialog box, provide the name of your Excel Workbook file (xls), or browse to locate your .xls file and click **OK**.
4. Select the specific worksheet or named range that you want to read.

Note: Use the **Options** button to modify default behaviors for Date and Time values, and so on. Click **OK** and then click **Next**.

5. From the Select Library and Member dialog box, choose the library where you want to create the new data set, enter a name for the new data set (member), and click **Next**.

Reminder: The target library must be assigned BEFORE using the Import Wizard.

6. Specify a path/filename for saving PROC IMPORT code that is generated by the Import Wizard and click **Finish**. If you do not want to generate this file, just click **Finish**.

Using PROC Import

```
proc import out=work.test datafile="c:\documents and settings\myfile.xls"
           dbms=excel replace;
  sheet="sheet1$";
  getnames=yes;
run;
```

Writing SAS Data to Excel

Using the LIBNAME – Excel Engine Statement

```
libname mylib "c:\documents and settings\myfile.xls";

data mylib.newtable;
  set sashelp.class;
run;

proc sql;
  create table mylib.newtable as select * from sashelp.shoes;
quit;
```

Using the Export Wizard

1. Select **File ► Export Data** to invoke the Export Wizard.
2. Choose the library and data set name of the data set that you want to write to Excel and click **Next**.
3. Under Standard Data Source, click the **drop-down arrow** and select the version of Excel to which you are writing and click **Next**.
4. On the Connect to Excel window, browse to locate the Excel .xls file to which you want to write and click **OK**.

Note: You can also specify a new Excel .xls file and the export process will create the .xls file.

5. In the Export Wizard - Select Table window under **Assign a name to the exported table** specify the name of the new worksheet and click **Next**.
6. From the Export Wizard - Create SAS Statements window you can have SAS produce a file that contains the statements generated from PROC EXPORT in a path/filename that you specify, and then click **Finish**. If you don't want to create this file, click **Finish**.

Using PROC Export

```
proc export data=sashelp.shoes
  outfile="c:\documents and settings\myfile.xls"
  dbms=excel2000 replace;
  sheet="shoes";
run;
```

Using the SAS/ACCESS® Interface to OLE DB

Reading Excel Data into SAS

The SAS/ACCESS Interface to OLE DB is a product that is licensed and installed in addition to Base SAS software.

To confirm that this product is included in your SAS site license, submit the following code and check the list of products that is written to the SAS Log:

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

To verify that the product has been installed, check your **!SASroot\access\sasexe** directory for the following file:

```
sasioole.dll
```

This product supports the LIBNAME-Engine Statement and the PROC SQL Pass-Through facility to read external data into SAS. The following examples illustrate using the SAS/ACCESS Interface to OLE DB to read Excel worksheet data:

Using the LIBNAME – OLE DB Engine Statement

```
libname mylib oledb provider="microsoft.jet.oledb.4.0"
      preserve_tab_names=yes
      preserve_col_names=yes
      properties=('data source'='c:\documents and settings
                \myfile.xls")
      provider_string="excel 8.0";

proc print data=mylib.'sheet1$'n;
run;

data work.test;
  set mylib.'sheet1$'n;
run;

proc sql;
  create table work.test as select * from mylib.'sheet1$'n;
quit;
```

Using the PROC SQL Pass-Through Facility

```
proc sql;
  connect to oledb(init_string="provider=microsoft.jet.oledb.4.0;
  password="");
data source=c:\documents and settings\myfile.xls;
  extended properties=excel 8.0;
  persist security info=true");
  create table foo as
    select * from connection to oledb
      (select * from [sheet1$]);
  disconnect from oledb;
quit;
```

Writing SAS Data to Excel

Using the SAS/ACCESS® Interface to OLE DB LIBNAME statement:

```
libname mylib oledb provider="microsoft.jet.oledb.4.0"
      preserve_tab_names=yes
      preserve_col_names=yes
      properties=('data source'='c:\documents and settings\myfile.xls")
      provider_string="excel 8.0";

data mylib.newtable;
  set sashelp.class;
run;

proc sql;
  create table mylib.newtable as select * from work.test;
quit;
```

Using the SAS/ACCESS® Interface to ODBC

Reading Excel Data into SAS

The SAS/ACCESS Interface to ODBC is a product that is licensed and installed in addition to Base SAS software.

To confirm that this product is included in your SAS site license, submit the following code and check the list of products that is written to the SAS Log:

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

To verify that the product has been installed, check your **!SASroot\access\sasexe** directory for the following file:

```
sasioodb.dll
```

You must also configure the Excel ODBC data source so that it *points* to the Excel workbook file that you want to access from SAS.

To configure your Excel ODBC data source, go to your computer desktop and perform these steps:

1. Follow this path to open the ODBC Administrator. Select **Start ► Settings ► Control Panel ► Administrative Tools ► Data Sources ODBC**.
2. When the Excel ODBC Driver is installed, a data source called 'Excel Files' is added to the User Data Sources. If you want to use that data source, select it and click **Configure**.
3. From the ODBC Microsoft Excel Setup dialog, in the Database section, use the **Select Workbook** button to browse to locate your .xls file. Then click **OK** to select that .xls file.
4. Click **OK** again to close the ODBC Microsoft Excel Setup dialog.
5. Click **OK** to close the ODBC Administrator.

To create a new data source, go to your computer desktop and perform these steps:

1. Follow this path to open the ODBC Administrator. Select **Start ► Settings ► Control Panel ► Administrative Tools ► Data Sources ODBC**.
2. From the User Data Sources window, click **Add**.
3. Choose the Microsoft Excel Driver (*.xls) from the list of ODBC drivers, and click **Finish**.
4. In the ODBC Microsoft Excel Setup dialog, specify a name for your data source (for example, MyXLS).
5. In the Database section, use the **Select Workbook** button to browse to locate your .xls file. Then click **OK** to select that .xls file.
6. Click **OK** again to close the ODBC Microsoft Excel Setup dialog.
7. Click **OK** to close the ODBC Administrator.

This product supports the LIBNAME-Engine statement and the PROC SQL Pass-Through Facility to read external data into SAS. The following examples illustrate using the SAS/ACCESS Interface to ODBC to read Excel worksheet data:

Using the LIBNAME – ODBC Engine Statement

```
libname mylib odbc dsn="excel files";

proc print data=mylib.'sheet1$'n;
run;

data work.test;
  set mylib.'sheet1$'n;
run;

proc sql;
  create table work.test as select * from mylib.'sheet1$'n;
quit;
```

Using the PROC SQL Pass-Through Facility

```
proc sql;
  connect to odbc(dsn="excel files");
  create table work.test as
  select * from connection to odbc
  (select * from [sheet1$]);
  disconnect from odbc;
quit;
```

Writing SAS Data to Excel

Using the SAS/ACCESS® Interface to ODBC LIBNAME statement:

```
libname mylib odbc dsn="excel files";

data mylib.newtable;
  set sashelp.class;
run;

proc sql;
  create table mylib.newtable as select * from work.test;
quit;
```


