



THE
POWER
TO KNOW.



Using SAS[®] 9.1.3 (and higher) to Read and Write to Microsoft Access Database Files in a Windows Operating Environment

Table of Contents

Using SAS 9.1.3 (or higher) to Read and Write to Microsoft Access Database

Files in a Windows Operating Environment.....	1
Using the SAS/ACCESS® Interface to PC Files.....	1
Reading Microsoft Access Data into SAS.....	1
Writing SAS Data to Microsoft Access.....	3
Using the SAS/ACCESS® Interface to OLE DB	4
Reading Microsoft Access Data into SAS.....	4
Writing SAS Data to Microsoft Access.....	5
Using the SAS/ACCESS® Interface to ODBC	5
Reading Microsoft Access Data into SAS.....	5
Writing SAS Data to Microsoft Access.....	7

Using SAS 9.1.3 (or higher) to Read and Write to Microsoft Access Database Files in a Windows Operating Environment

When using SAS in a Windows operating environment, there are three approaches to reading and writing Microsoft Access .mdb files.

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

Using the SAS/ACCESS® Interface to PC Files

Reading Microsoft Access 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 this file:

```
sasimode.dll
```

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

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

Using the LIBNAME – SAS/ACCESS Engine Statement

```
libname mylib "c:\documents and settings\mydata.mdb";

proc print data=mylib.accesstable; run;

data work.test;
  set mylib.accesstable;
  run;

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

Using the PROC SQL Pass-Through Facility

```
proc sql;
  connect to access(path="c:\documents and settings\mydata.mdb");
  create table work.test as
    select * from connection to access
      (select * from accesstable);
disconnect from access;
quit;
```

Using the Import Wizard

1. Choose the File–Import Data menu to invoke the Import Wizard.
2. Select the appropriate Microsoft Access database version from the drop-down list of Standard Data Sources, and click **Next**.
3. On the Connect to Microsoft Access dialog box, provide the name of your Microsoft Access database file (.mdb), or browse to locate your .mdb file. If required, provide the appropriate user authentication and workgroup filename and click **OK**.
4. Select the Microsoft Access table (or query) that you want to read from the drop-down list.

Note: Use the **Options** button to modify default behaviors for Memo Fields, Date, and Time values.

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
  datatable="mytable"
  dbms=access replace;
  database="c:\documents and settings\mydata.mdb";
run;
```

Writing SAS Data to Microsoft Access

Using the LIBNAME – SAS/ACCESS Engine Statement

```
libname mylib "c:\documents and settings\mydata.mdb";

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

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

Using the Export Wizard

To use the Export Wizard to write a SAS data set to a table in a Microsoft Access database do the following:

1. Choose the File - Export Data menu to invoke the Export Wizard.
2. Choose the Library and name of the data set that you want to write to Microsoft Access, and click **Next**.
3. Under Standard Data Source, click the **drop-down arrow** and select the version of Microsoft Access to which you are writing and click **Next**.
4. On the 'Connect to Microsoft Access' window, browse to locate the Microsoft Access .mdb file in which you want to add this new table. If the Microsoft Access database requires a user ID and password, specify that information here, and click **OK**.

Note: You may also specify a new Microsoft Access .mdb filename and the export process will create the .mdb file.

5. In the Export Wizard - Select Table window under **Assign a name to the exported table**, specify the name of the new table you are creating 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 Statements

```
proc export data=sashelp.shoes
  outtable="shoes"
  dbms=Access replace;
  database="C:\my documents\mydata.mdb"
```

Using the SAS/ACCESS® Interface to OLE DB

Reading Microsoft Access 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 SAS installation directories for this 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 Microsoft Access data:

Using the LIBNAME – OLE DB Engine Statement

```
libname mylib oledb init_string="provider=microsoft.jet.oledb.4.0;password="";
data source=c:\documents and settings\mydata.mdb;
persist security info=true";

proc print data=mylib.accesstable; run;

data work.test;
  set mylib.accesstable;
  run;

proc sql;
  create table work.test as select * from mylib.accesstable;
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\mydata.mdb;
persist security info=true");
create table work.test as
select * from connection to oledb
(select * from accesstable);
disconnect from oledb;
quit;
```

Writing SAS Data to Microsoft Access

```
libname mylib oledb init_string="provider=microsoft.jet.oledb.4.0;password="";
data source=c:\documents and settings\mydata.mdb;
persist security info=true";

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 Microsoft Access 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 SAS installation directories for this file:

```
sasioodb.dll
```

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

To configure your Microsoft Access 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 Microsoft Access ODBC Driver is installed, a data source called 'MS Access Database' 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 Access Setup dialog, in the Database section, use the **Select** button to browse to locate your .mdb file.
4. Click **OK** to select that .mdb file, and click **OK** again to close the ODBC Microsoft Access 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: **Start ► Settings ► Control Panel ► Administrative Tools ► Data Sources ODBC.**
2. From the User Data Sources window, click **Add**.
3. Choose the Microsoft Access Driver (*.mdb) from the list of odbc drivers, and click **Finish**.
4. In the ODBC Microsoft Access Setup dialog, specify a name for your data source (for example, MyMSA).
5. In the Database section, use the **Select** button to browse to locate your .mdb file. Then click **OK** to select that .mdb file.
6. Click **OK** again to close the ODBC Microsoft Access 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 Microsoft Access data:

Using the LIBNAME – ODBC Engine Statement

```
libname mylib odbc dsn="ms access database";

proc print data=mylib.accesstable; run;

data work.test;
  set mylib.accesstable;
  run;

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

Using the PROC SQL Pass-Through Facility

```
proc sql;
  connect to odbc(dsn="ms access database");
  create table work.test as
  select * from connection to odbc
  (select * from accesstable);
  disconnect from odbc;
quit;
```

Writing SAS Data to Microsoft Access

Using the SAS/ACCESS Interface to ODBC LIBNAME statement:

```
libname mylib odbc dsn="ms access database";

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

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

SAS 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. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.
Copyright © 2008 SAS Institute Inc., Cary, NC, USA. All rights reserved.