

ODBC Data Reading



Destiny Corporation • 100 Great Meadow Rd Suite 601
Wethersfield, CT 06109-2355
Phone: (860) 721-1684 • 1-800-7TRAINING
Fax: (860) 721-9784
Email: info@destinycorp.com Web Site: www.destinycorp.com

Prepared by Dana Rafiee

Copyright ©2001 Destiny Corporation. All Rights Reserved.

Setting Up The Environment

ODBC stands for Open Database Connectivity. This is a Microsoft Standard. Its purpose is to help systems and software share different forms of data, no matter what their internal method of storage is. As long as the software subscribes to supporting ODBC, data can usually be read from and written to that software.

The first step is to register the data source on the computer. Each computer requires this registration process.

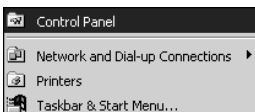
Each software that subscribes to the ODBC protocol has its own drivers that need to be installed first. For example, when ODBC information is needed from an Oracle database, Oracle's supplied driver must be installed on the machine that will access that data. Driver versions change often. Any problems encountered can usually be solved by installing the vendor's latest software driver(s).

In the following examples, Microsoft Office XP has already been installed on this Windows 2000 machine.

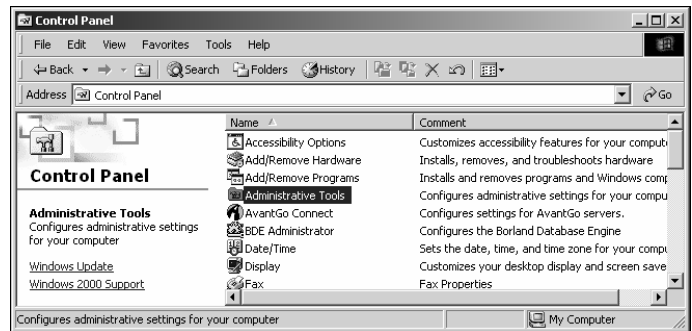
After the driver for the desired data source has been installed, the data source must be defined to the operating system.

Defining The Data Source

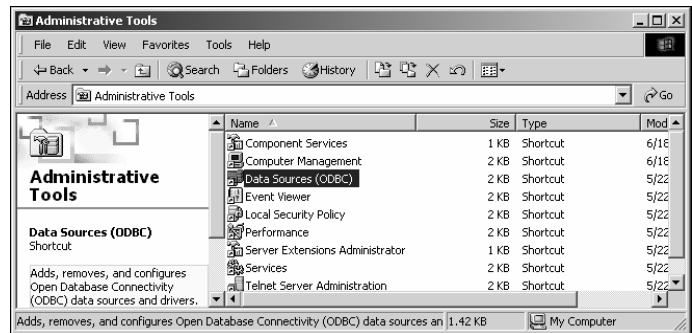
From the start menu, select Control Panel.



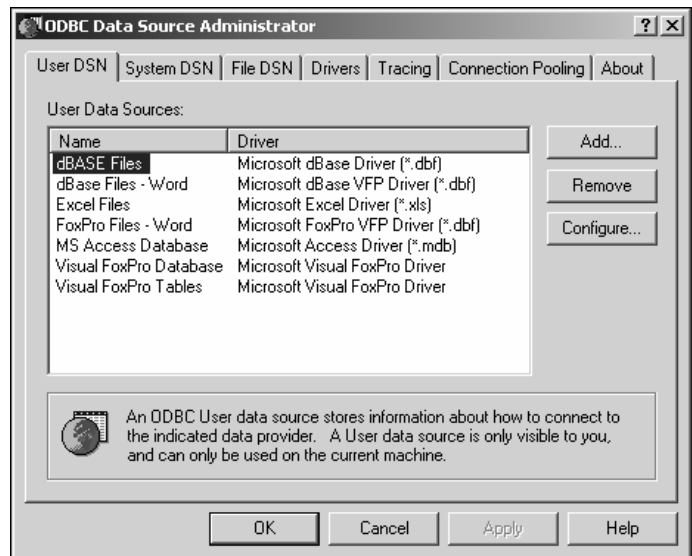
Select Administrative Tools.



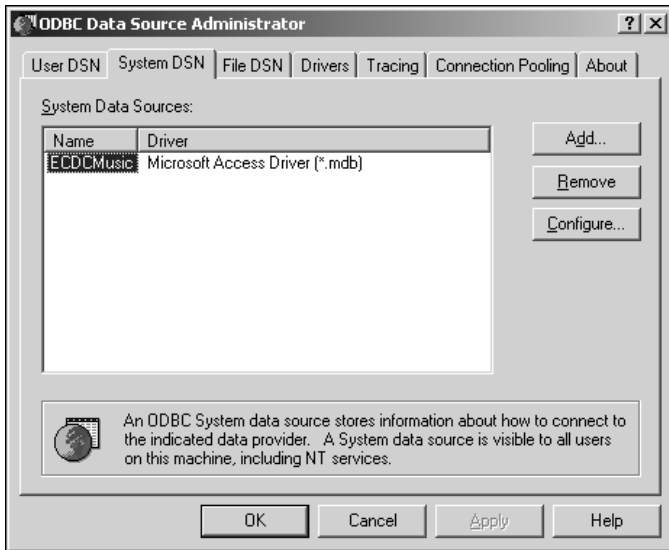
Select Data Sources (ODBC)



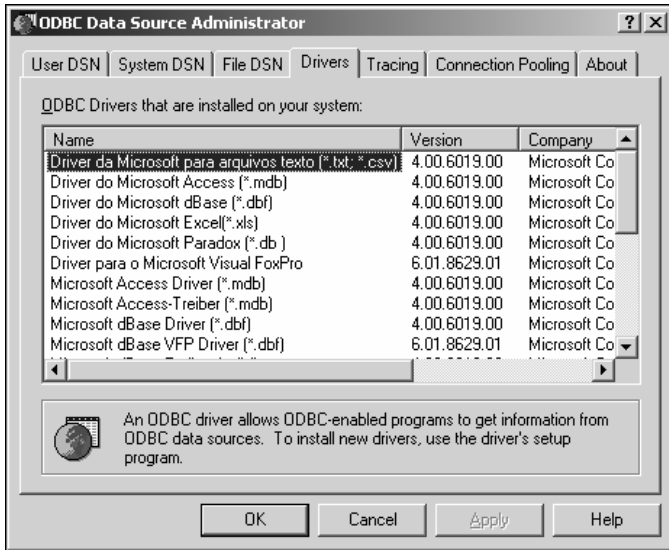
The following screen will appear.



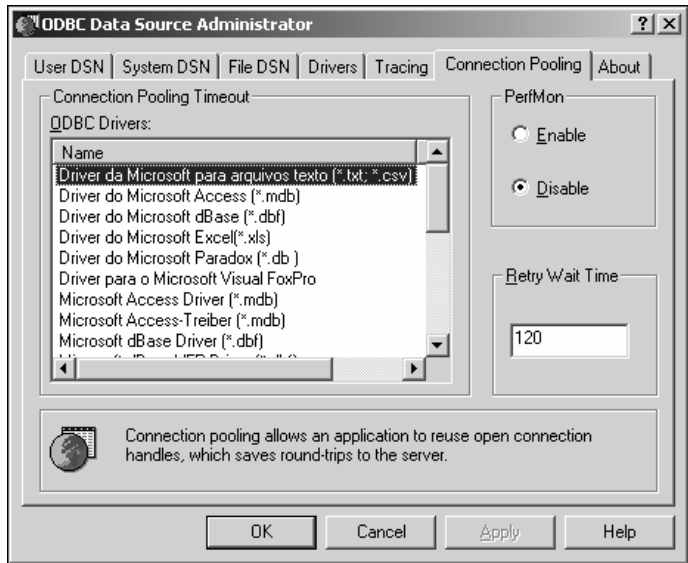
You have a choice as to define a data source at the User level or the System level. The difference is that if it is at the User level, then only this user, currently logged on, can access this data. If it is at the System level, then any user of this machine can access this data source.



Clicking on Drivers will show you which software application drivers are already installed on this machine.

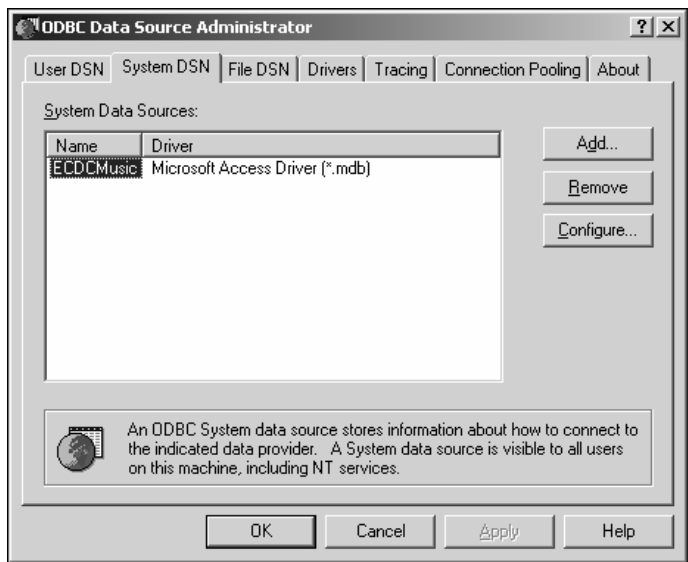


Connection Pooling allows reusing existing connections for the drivers listed here. This is more efficient.

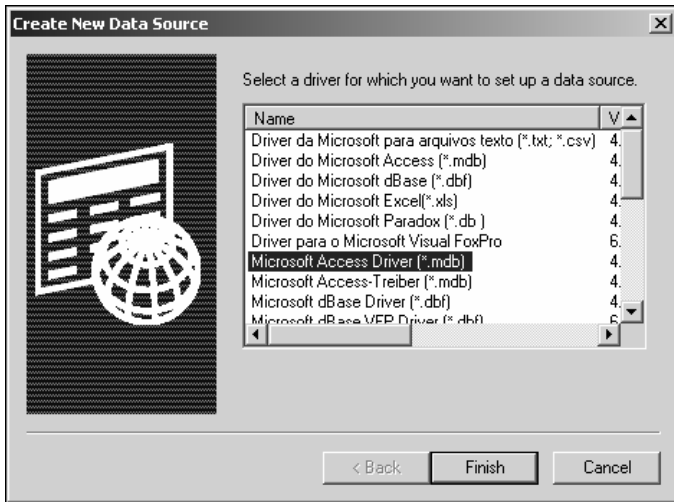


Microsoft Access will be used for this example. We can see that it is listed in all of these selections.

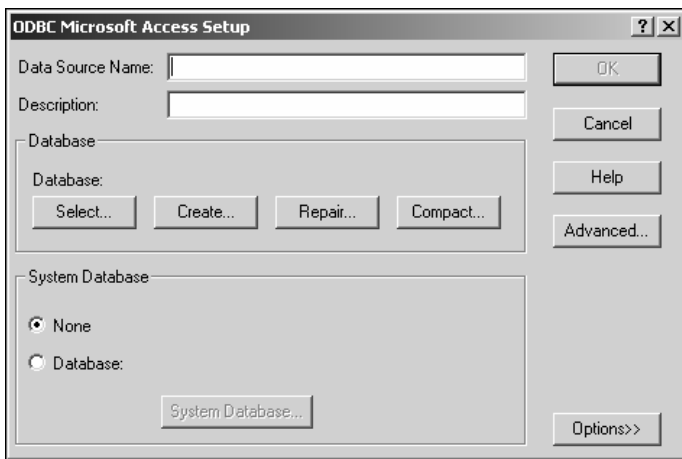
Let's go back to System DSN to add a new data source.



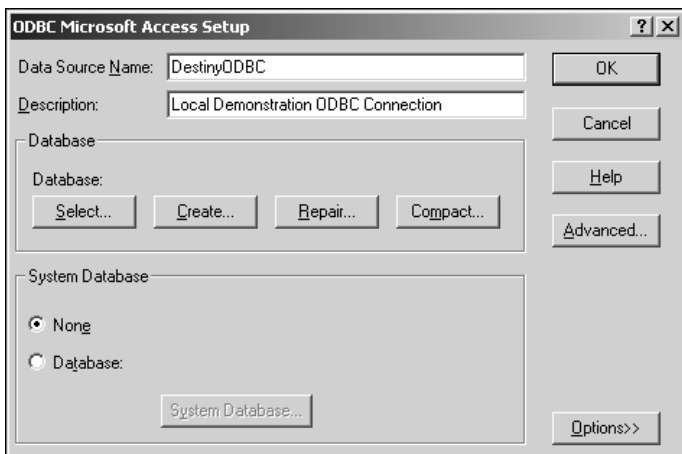
Click  and select the type of data source.



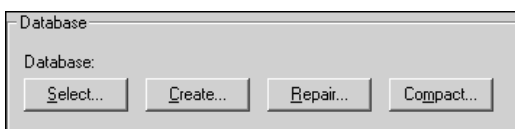
The following screen will appear.



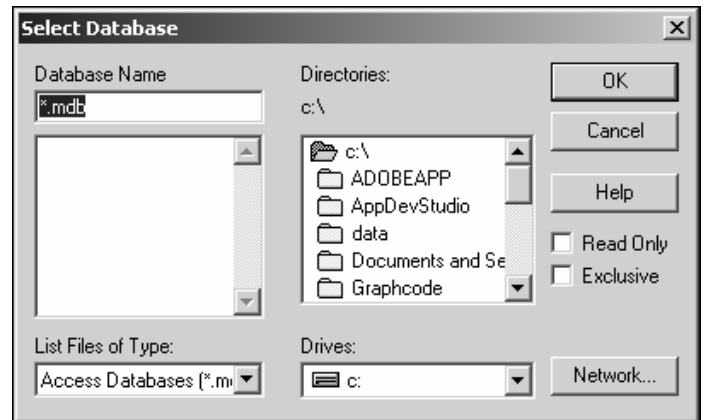
Give a name and description similar to the following.



Click the Select button to specify the Microsoft Access file.



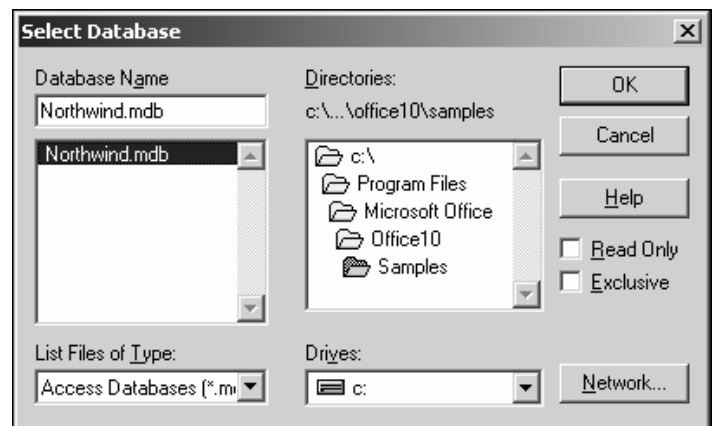
The following screen will appear.



In this example, we will use the sample Microsoft Access file called Northwind.mdb. Under this Microsoft Office XP installation, it is located in the following location.

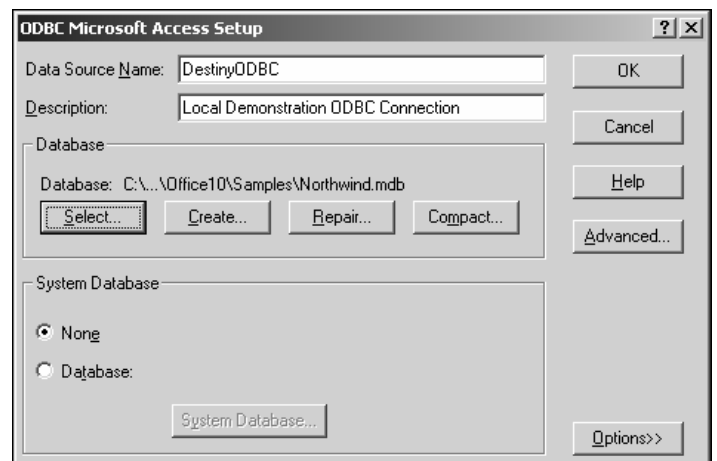
c:\Program Files\Microsoft Office\Office10\Samples\Northwind.mdb

Select it.

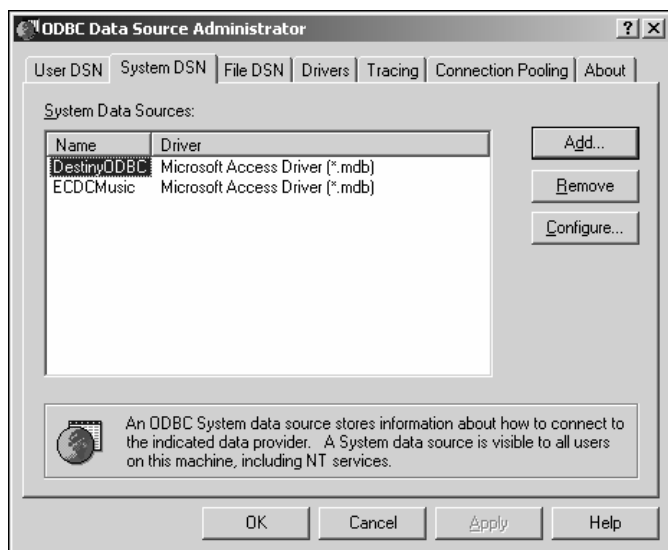


Click **OK** when finished.

You will see the following screen.



Click again and the data source should be listed as follows.

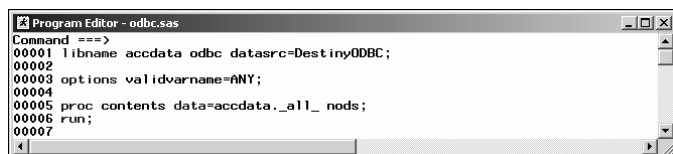


Click to close the ODBC Data Source Administrator.

Accessing the Data Source with SAS

The data source is made available to SAS with a Libname statement. The engine on the Libname statement is ODBC and the DATASRC= option references the name of the ODBC data source we defined earlier.

Microsoft Access is structured as a file that is like a database. The file contains multiple tables. These tables are seen to SAS as if they were SAS data sets.



The Proc Contents code allows us to get a list of those internal tables.

The validvarname option allows us to convert those tables with non-standard SAS names to valid SAS data sets. e.g. names with spaces.

The CONTENTS Procedure		
-----Directory-----		
Libref:	ACCDATA	
Engine:	ODBC	
Physical Name:	DestinyODBC	
Schema/Owner:	admin	
#	Name	Memtype
1	Alphabetical List of Products	DATA
2	Categories	DATA
3	Category Sales for 1997	DATA
4	Current Product List	DATA
5	Customers	DATA
6	Customers and Suppliers by City	DATA
7	Employee Sales by Country	DATA
8	Employees	DATA
9	Invoices	DATA
10	Invoices Filter	DATA
11	MSysACEs	DATA
12	MSysAccessObjects	DATA
13	MSysCmdbars	DATA
14	MSysIMEXColumns	DATA
15	MSysIMEXSpecs	DATA
16	MSysObjects	DATA
17	MSysQueries	DATA
18	MSysRelationships	DATA
19	Order Details	DATA
20	Order Details Extended	DATA
21	Order Subtotals	DATA
22	Orders	DATA
23	Orders Qry	DATA
24	Product Sales for 1997	DATA
25	Products	DATA
26	Products Above Average Price	DATA
27	Products by Category	DATA
28	Quarterly Orders	DATA
29	Quarterly Orders by Product	DATA
30	Sales by Category	DATA
31	Sales by Year	DATA
32	Shippers	DATA
33	Suppliers	DATA
34	Ten Most Expensive Products	DATA

Additional Libname Options

The Libname statement offers additional options that may be relevant to your use of SAS/Access to ODBC Software. See the specifics in the Online SAS Help facility to understand how SAS accesses your database software.

LIBNAME Options for ODBC

The following table describes the ODBC interface support for SAS/ACCESS LIBNAME options and presents default values where applicable. For detailed information about an option, click its name.

SAS/ACCESS LIBNAME Options

Option	Default Value
ACCESS=	none
AUTOCOMMIT=	data source specific
BL_LOG=	none
BL_OPTIONS=	none
BULKLOAD=	NO
CONNECTION=	data source specific
CONNECTION_GROUP=	none
CURSOR_TYPE=	DYNAMIC
DBCOMMIT=	1000 (inserting) or 0 (updating)
DBCONINIT=	none
DBCONTERM=	none
DBCREATE_TABLE_OPTS=	none
DBGEN_NAME=	DBMS
DBINDEX=	YES
DBLIBINIT=	none
DBLIBTERM=	none
DBMAX_TEXT=	1024
DBNULLKEYS=	YES
DBPROMPT=	NO
DEFER=	NO
DELETE_MULT_ROWS=	NO
DIRECT_SQL=	YES
INSERT_SQL=	data source specific
INSERTBUFF=	1
KEYSET_SIZE=	0
PRESERVE_COL_NAMES=	see Naming Conventions for ODBC
PRESERVE_TAB_NAMES=	see Naming Conventions for ODBC
QUALIFIER=	none
QUERY_TIMEOUT=	0
QUOTE_CHAR=	none
READBUFF=	0
READ_ISOLATION_LEVEL=	RC (see Locks in the ODBC Interface)
READ_LOCK_TYPE=	ROW
REREAD_EXPOSURE=	NO
SCHEMA=	none
SPOOL=	YES
STRINGDATES=	NO
TRACE=	NO
TRACEFILE=	none
UPDATE_ISOLATION_LEVEL=	RC (see Locks in the ODBC Interface)
UPDATE_LOCK_TYPE=	ROW
UPDATE_MULT_ROWS=	NO
UPDATE_SQL=	driver specific
USE_ODBC_CL=	NO