

# **Using ODBC with MDaemon 6.5**

Alt-N Technologies, Ltd  
1179 Corporate Drive West, #103  
Arlington, TX 76006  
Tel: (817) 652-0204

© 2002 Alt-N Technologies. All rights reserved.  
Other product and company names mentioned in this document may be trademarks.

# Contents

<b>Abstract</b> .....	<b>3</b>
<b>Internal Files and External Databases for Email Servers</b> .....	<b>4</b>
Sharing Data with Databases .....	4
Effectiveness through Sharing .....	4
One Data Element for All Applications .....	5
<b>Basic Concepts of ODBC</b> .....	<b>6</b>
<b>MDaemon ODBC Interfaces</b> .....	<b>7</b>
User Accounts.....	7
Mailing List.....	8
<b>MDaemon ODBC Setup and Configuration</b> .....	<b>9</b>
MS Access Sample Instructions.....	9
User Accounts .....	9
Mailing List.....	17
MySQL Sample Instructions .....	21
User Accounts .....	21
<b>Questions, Exclamations and Answers</b> .....	<b>28</b>
User Account Questions .....	28
What is the data structure for the MDaemon ODBC interface? .....	28
How do I switch back to userlist.dat for my user accounts? .....	28
How can I change the database used to store my user accounts?.....	29
There is no script for creating the user account tables for my DBMS! .....	29
Mailing List Questions.....	29
What tables must I use for the mailing list data source?.....	29
My data does not match the mailing list wizard data elements!.....	29
Must I be an SQL expert to use the mail list interface? .....	29

## Abstract

While maintaining user account records on MDAemon is easy, it can be redundant. Redundancy possibly occurs if your enterprise also keeps computerized information about employees or email customers, for example. Redundant record keeping wastes time and adds the risk of information being inconsistent among various data sources. MDAemon 6.5 addresses this problem by allowing account records to be maintained in any database with an Open Database Connectivity (ODBC) interface. MDAemon 6.5 also has the ability to access mailing list records through an ODBC connection.

**Note:** This white paper includes step-by-step instructions for setting up MDAemon to use an ODBC interface for Microsoft Access and the open source MySQL database. Database management — including installation and configuration — is beyond the scope of this paper.

# Internal Files and External Databases for Email Servers

## Sharing Data with Databases

Email servers generally use internal data structures for storing configuration preferences, user account information and individual email messages.

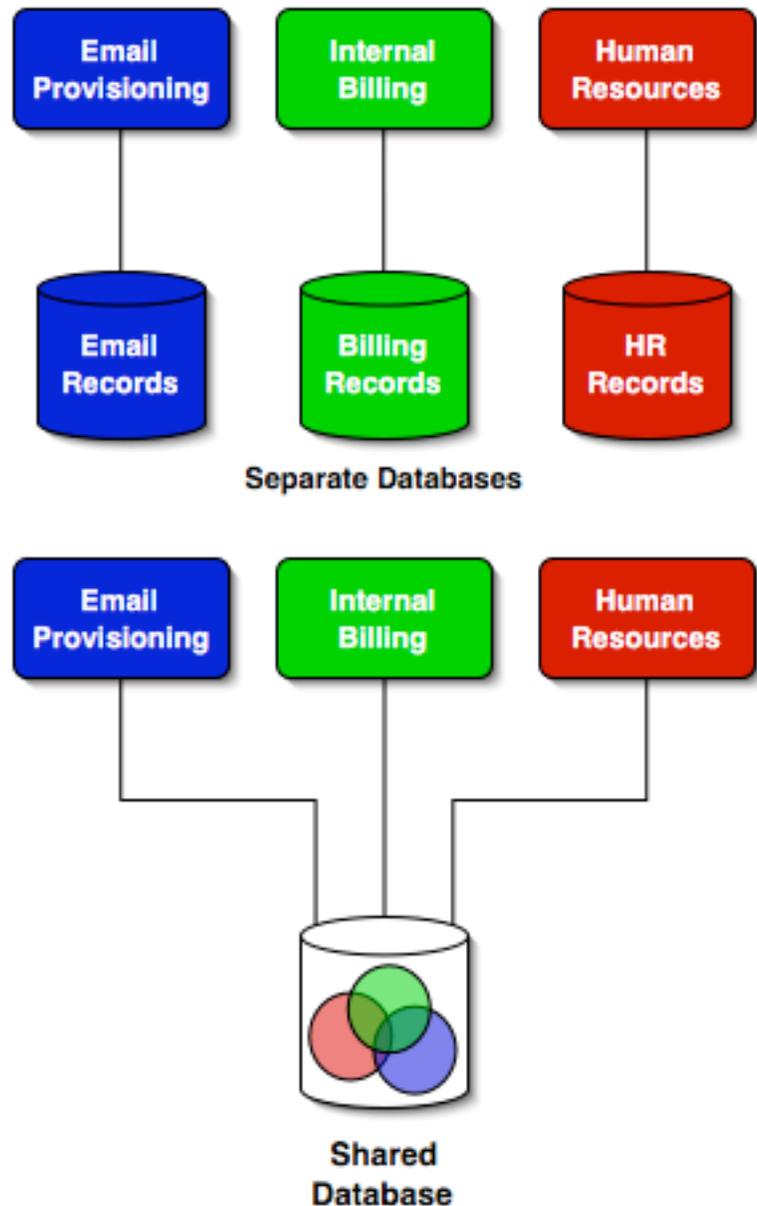
The server alone typically uses the configuration preferences and email messages. But it potentially could share some of its user account information with other applications. For example, data such as user names, ids, domain names, account options and email restrictions may be useful for marketing, sales, billing, accounting, network planning and human resource applications, as well as for the email server.

## Effectiveness through Sharing

Sharing such common information among multiple applications helps all parts of an enterprise — large or small — work together more effectively. Using a centralized source for common data promotes accuracy and consistency among multiple application programs. On the other hand, each application keeping its own common data separate can create problems with duplicate data maintenance and inconsistent records.

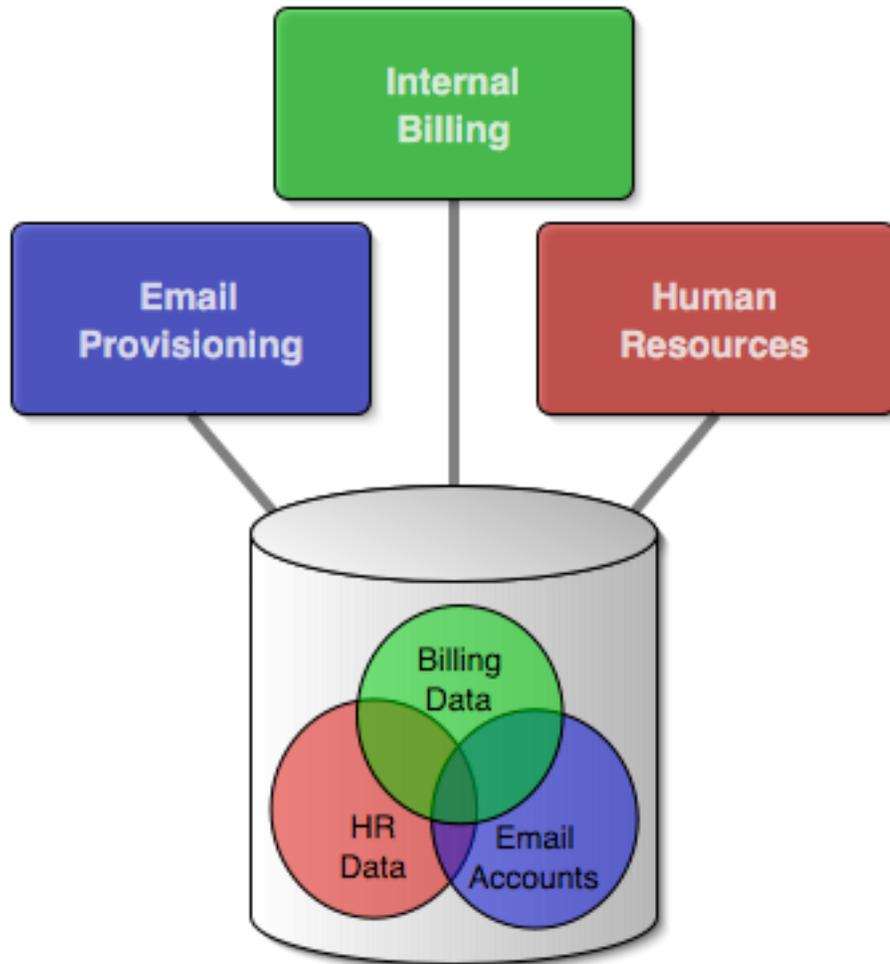
The illustration shows the difference between multiple applications keeping their own copies of common data and each application sharing common information through a database.

The separate databases might contain redundant and possibly inconsistent data such as the name of one user being stored as William F. Sharp, Will Sharp and Bill Sharp. Sharing common data through a database, assures more accurate and consistent information.



## One Data Element for All Applications

The illustration shows more details of the shared database concept. Each circle represents the data for one application. The overlapping parts of the circles represent data common to two or more applications. All three applications share some data. Each application can access its data from the central database. However, each data element occurs only one time in the data store, no matter how many applications use it. So user name, user id and company name, for example, occur only once for each person.

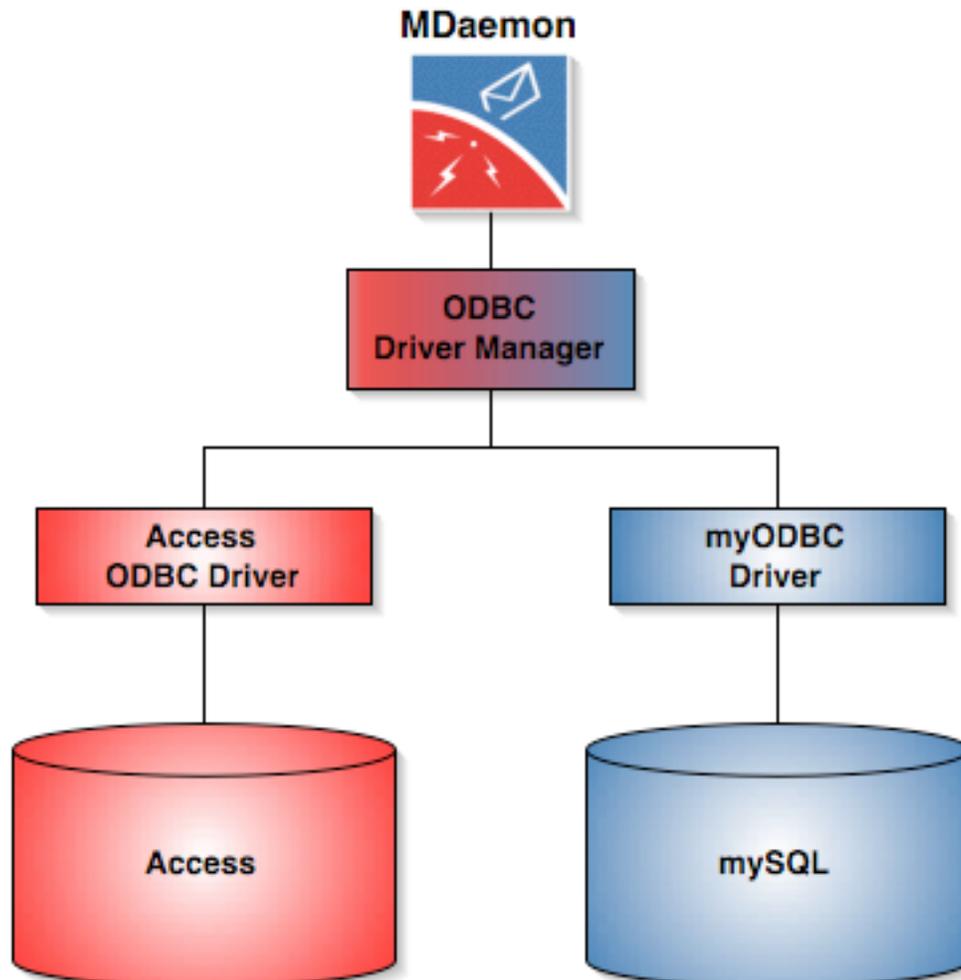


In Microsoft Windows, the most common way of implementing this type of sharing is through Open Database Connectivity, a Microsoft data management tool for application development.

## Basic Concepts of ODBC

Open Database Connectivity (ODBC) defines a database interface. It allows any compliant application to access various types of databases without regard to the specifics of the data structures. Each database management system provides its own ODBC interface. Most popular database management systems for Windows, Unix and Macintosh have ODBC drivers.

The illustration shows schematically how a single application can communicate with diverse databases through the ODBC interface.



When a database has an ODBC driver, an application can access that database by using the driver as its interface. This means the application does not know or care about the type of database containing the information. To change databases means only changing ODBC drivers. Nothing else is normally needed. The driver handles all of the details.

MDaemon uses its ODBC interface for optionally accessing external databases for user accounts and read-only mailing lists.

# MDaemon ODBC Interfaces

## User Accounts

The data available through the ODBC interface identifies the account and its enabled features. An enterprise could use this data for account billing, for example. Billing might include optional charges for a service such as the amount of disk space allocated to the account.

The data elements available for the user account are:

**UserID**

Sequential number — such as 1, 2 or 1062 — assigned to the account by the software. This is the primary key.

**Mailbox**

Account name, such as *vernon* in *vernon@company.net*. This is part of the unique key.

**DomainID**

Domain name, such as *company.net* in *vernon@company.net*. This is part of the unique key.

**FullName**

Name of the account holder, such as *Vernon Talbert*.

**MailDir**

Directory location of the mailbox, such as *c:\MDaemon\Users\company.mail\ted*.

**Password**

Password for accessing the account.

**AutoDecode**

Does MDAemon automatically extract and store MIME encoded attachments send to the account.

**IsForwarding**

Does the account automatically forward email?

**AllowAccess**

Is the account active?

**AllowChangeViaEmail**

Can the can configure the account by emailing MDAemon?

**KeepForwardedMail**

Does the account keep a copy of the email automatically forwarded?

**HideFromEveryone**

Is this a hidden account?

**EncryptMail**

Does MDAemon encrypt mail for this account before storing it?

**ApplyQuotas**

Are size or quantity storage limits applied to the account? *MaxMessageCount* and *MaxDiskSpace* show the size of the limits.

**EnableMultiPop**

Does the account collect mail from Post Office Protocol accounts?

#### CanModifyGAB

Can this account modify the public address book?

#### MaxMessageCount

What is the maximum number of messages the account can store? This applies only if account quotas are active.

#### MaxDiskSpace

What is the maximum amount of space the account can use? This applies only if account quotas are active.

## Mailing List

Every MDAemon mailing list can have an ODBC link to one unique data source. One list might use data on a local machine running MySQL. Another could link to mailing list data in a SQLServer table on a remote server.

The ODBC interface for mailing lists is read-only. The interface is a means of using a send-only mailing list maintained by an application external to MDAemon. For example, an enterprise might have a mailing list of everyone in the personnel database. List members would be added and removed automatically from the list when they join and leave the company.

However, even with when using an ODBC link, all other functions of a mailing list still work. This allows for creating a mailing list where some members can choose to subscribe and unsubscribe, while others always receive all mailings. An example would be a company mail list where all employees receive the email and people outside can subscribe.

The data elements available for the user account are variable because they come from existing databases. Generally they would include data elements such as:

#### Email

Complete email address, such as *vernon@company.net*.

#### FirstName

First Name of the account holder, such as *Vernon*.

#### LastName

Last Name of the account holder, such as *Talbert*.

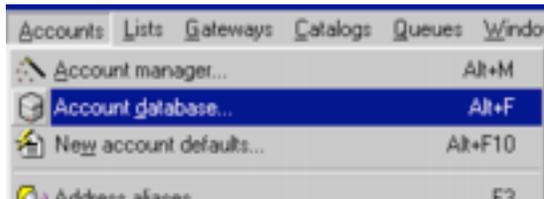
# MDaemon ODBC Setup and Configuration

**Note:** These Instructions include steps for setting up MDaemon to use an ODBC interface for Microsoft Access and the open source MySQL database. Database management — including installation and configuration — is beyond the scope of these instructions.

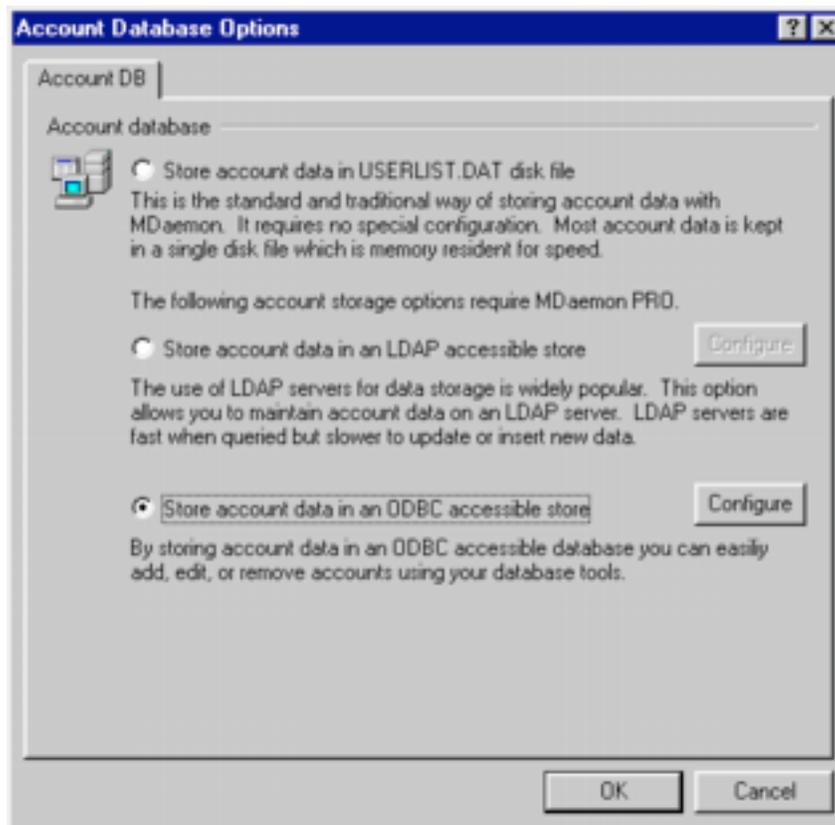
## MS Access Sample Instructions

### User Accounts

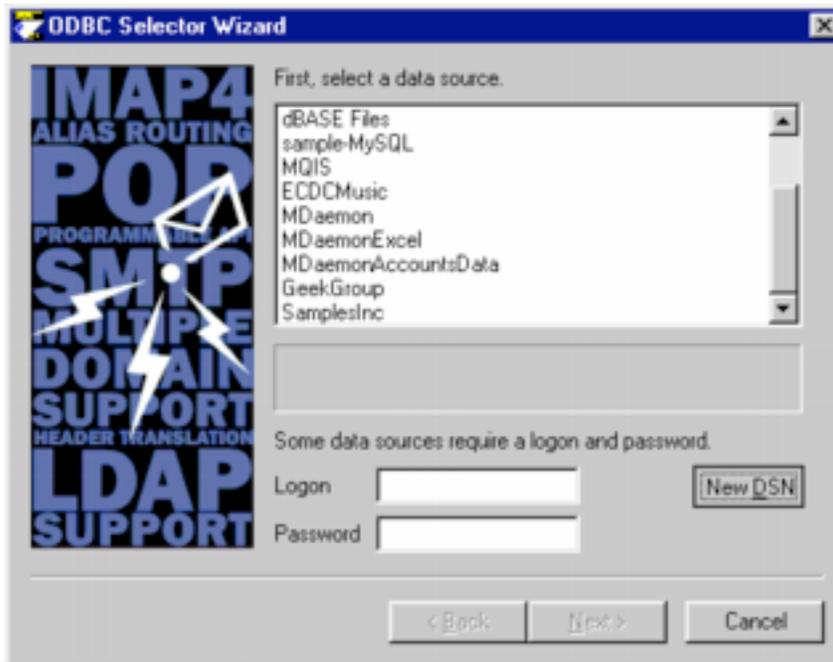
The instructions begin with the MDAEMON ADMINISTRATOR GUI running. They show how to configure a new DSN. The datasource can be password protected.



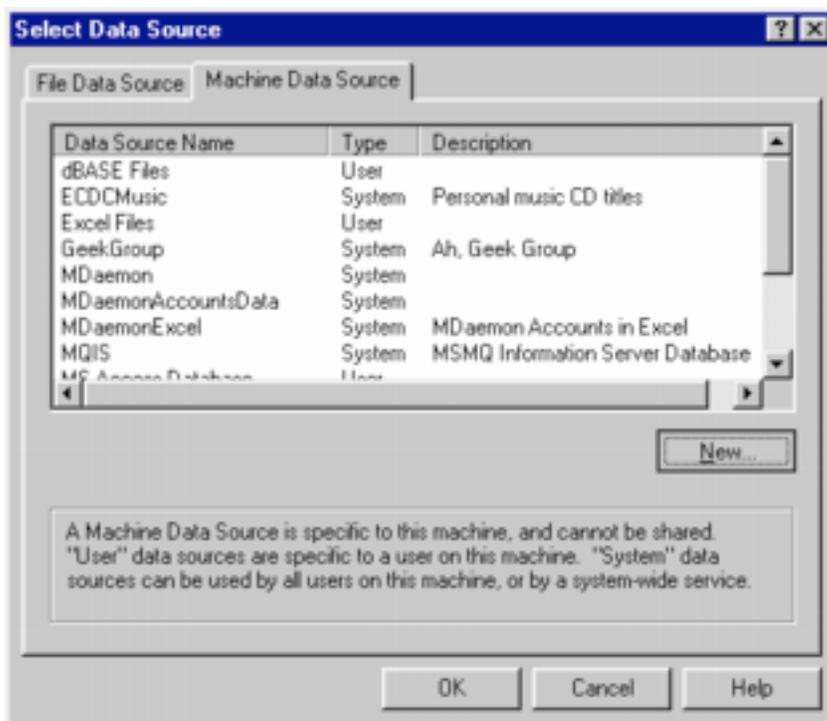
1. Use the ACCOUNT DATABASE command from the ACCOUNTS menu. This displays the ACCOUNTS DATABASE OPTIONS dialog.



2. Choose the STORE ACCOUNT DATA IN AN ODBC ACCESSIBLE STORE radio button and use the CONFIGURE command button. This displays the ODBC SELECTOR WIZARD.



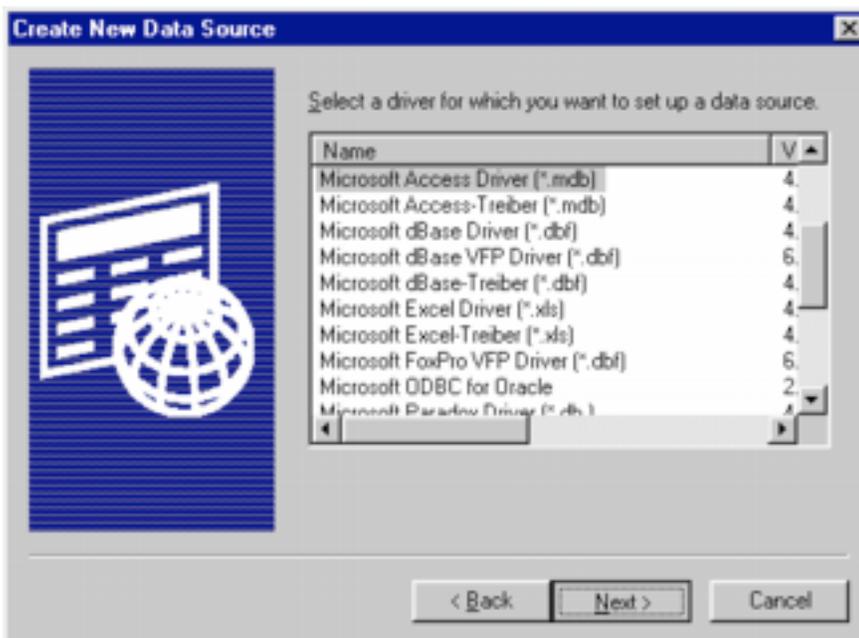
3. Use the NEW DSN command button. This displays the SELECT DATA SOURCE dialog.



4. Select the MACHINE DATA SOURCE tab and Use the NEW command button. This displays the CREATE NEW DATA SOURCE wizard.



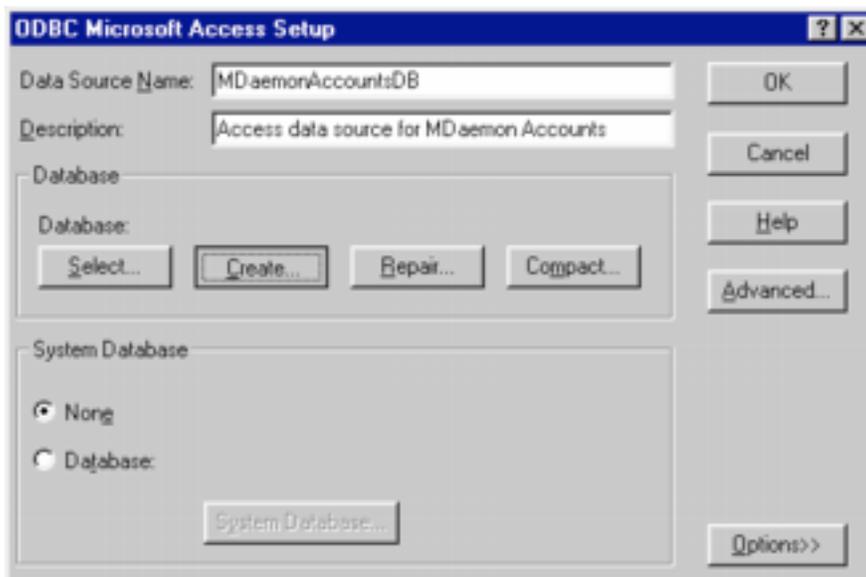
5. Choose the SYSTEM DATA SOURCE radio button and use the NEXT command button. This displays the dialog for selecting the ODBC driver.



6. Choose MICROSOFT ACCESS DRIVER from the list box and use the NEXT command button. This displays a confirmation dialog.

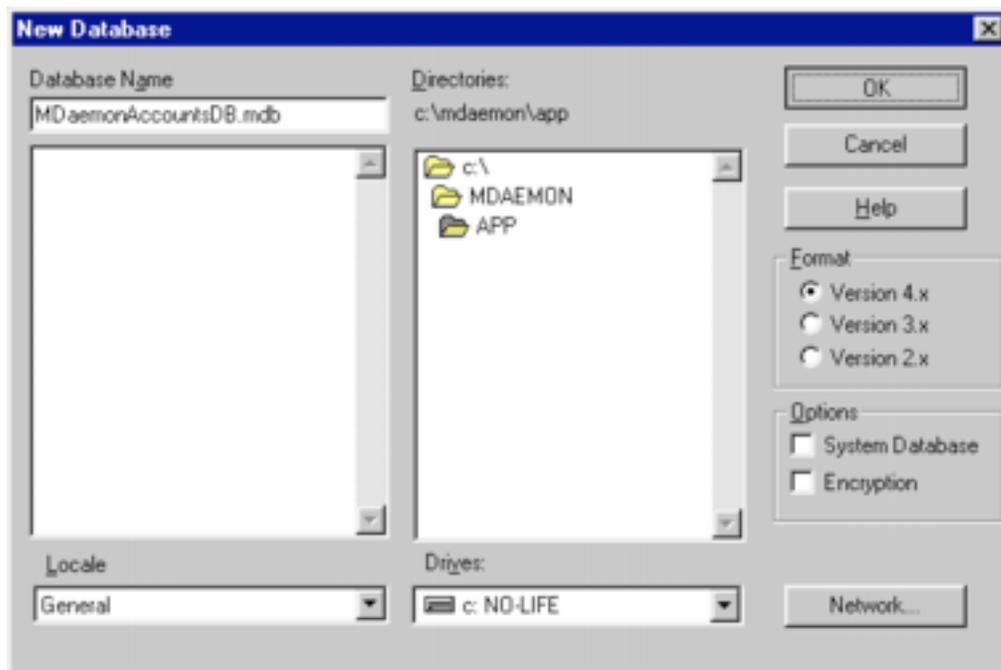


7. Use the FINISH command button to display the ODBC MICROSOFT ACCESS SETUP dialog.

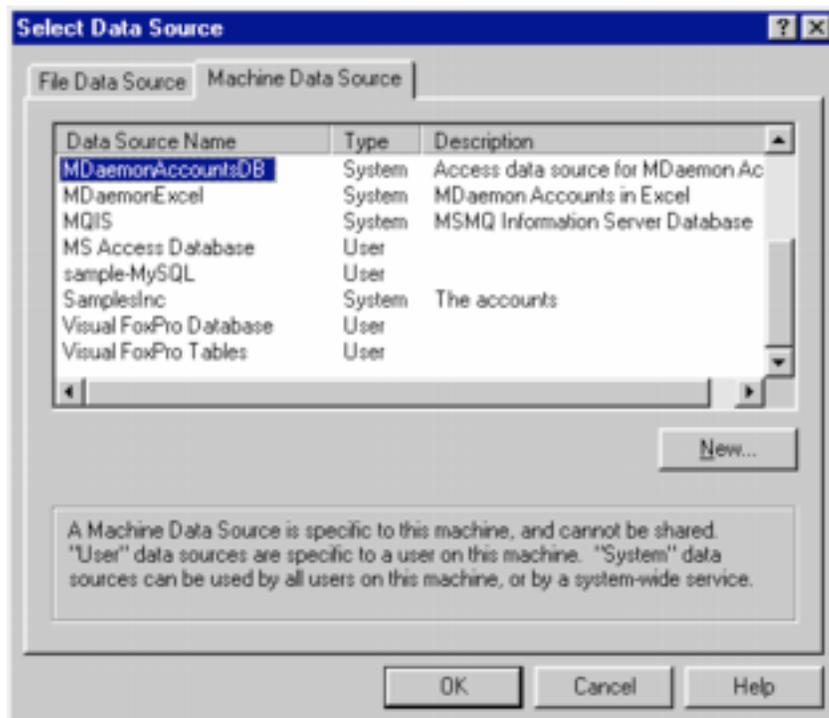


On this dialog you can SELECT or CREATE a database. Select a data source to add the tables to an existing database. Create a database to make a new database. These instructions show how to create a database. If you are selecting a database, you can skip to step 13.

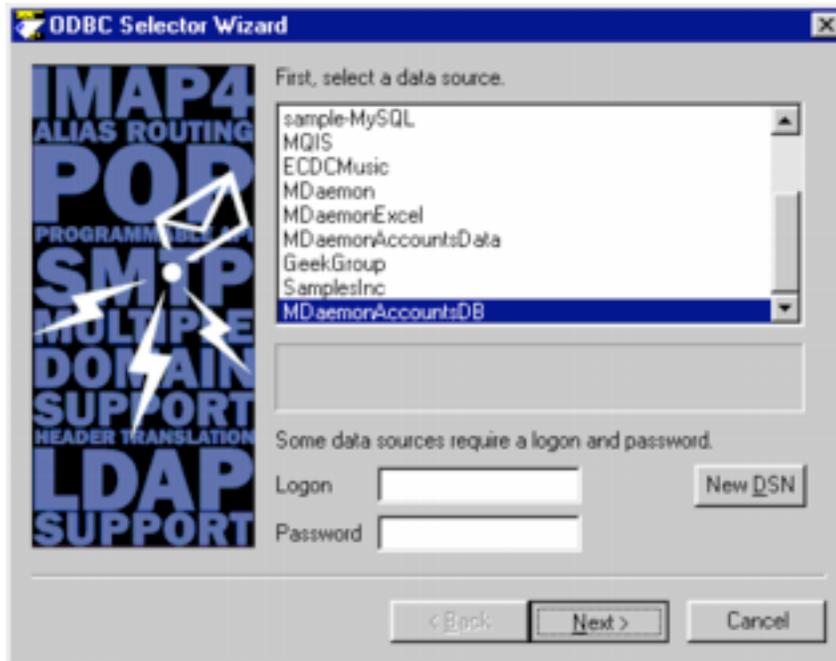
- Enter a DATA SOURCE NAME and DESCRIPTION, then use the CREATE command button. This displays a dialog for naming the database.



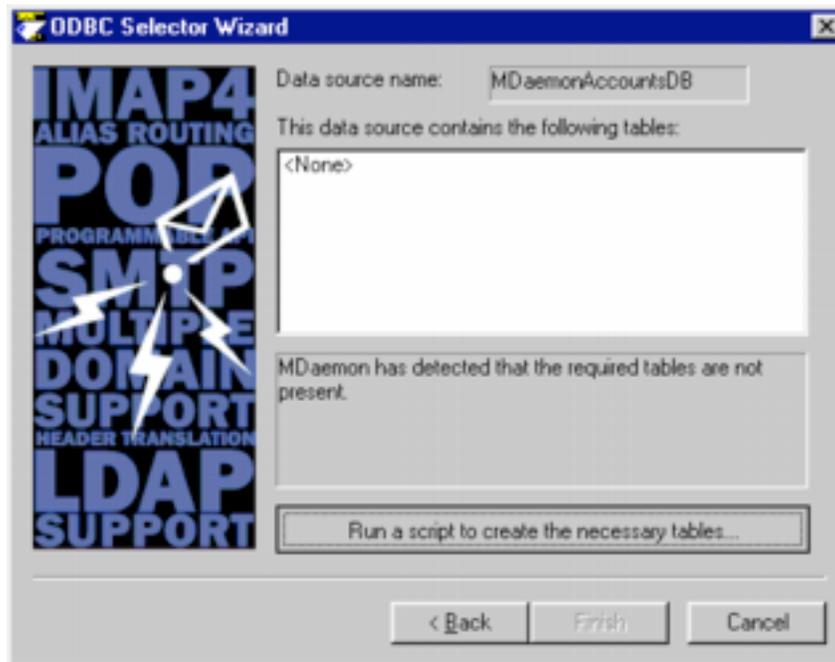
- Name the database, find a place to store the database (in this case it must be on the local machine), use the OK command button to display the confirmation message. Use the OK command button to redisplay the ODBC MICROSOFT ACCESS SETUP dialog.
- Use the OK command button to redisplay the SELECT DATA SOURCE dialog.



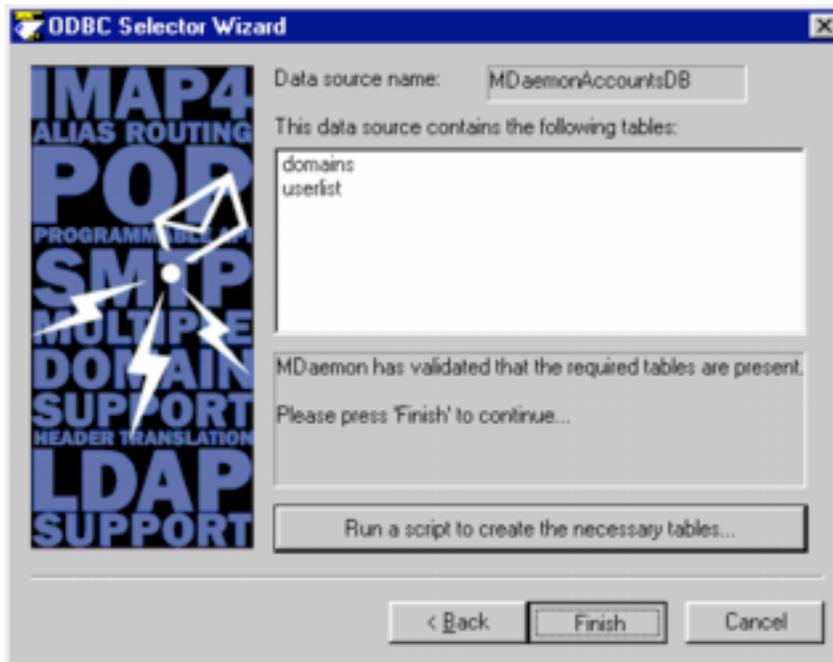
11. Use the OK command button to redisplay the ODBC Selector Wizard.



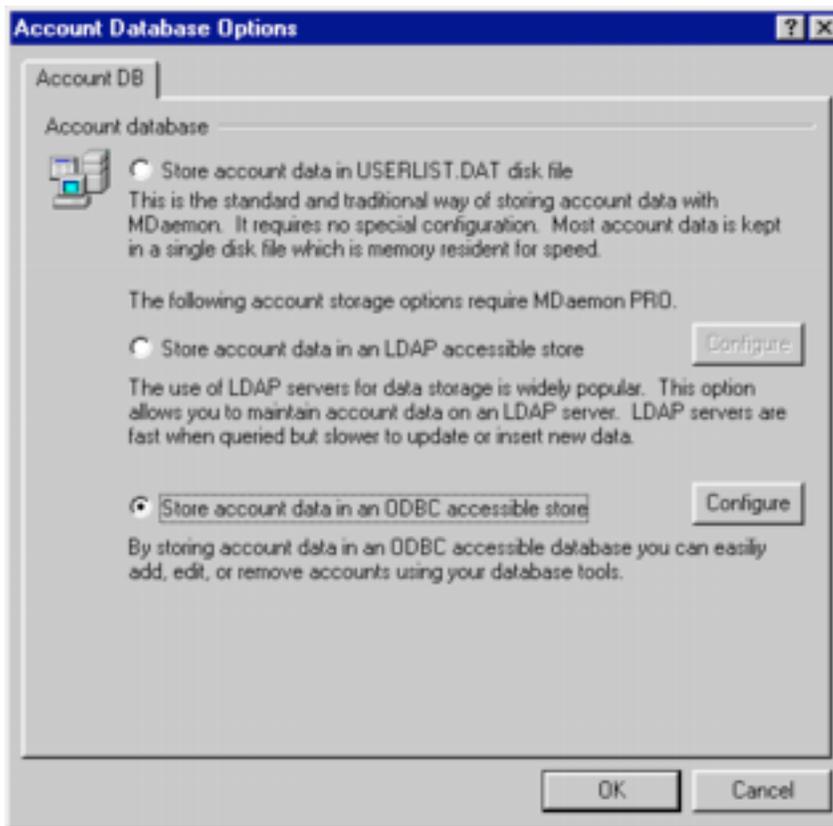
12. Select the data source and use the NEXT command button to display the dialog showing the tables in the database.



13. Use the RUN A SCRIPT TO CREATE THE NECESSARY TABLES command button to display a dialog for creating the tables. Use the BROWSE button to locate and choose the ACCOUNTSDB-ACCESS.SQL script. Use the RUN SCRIPT AND CREATE TABLES NOW command button to create the tables. Then use the Close command button to return to the ODBC SELECTOR WIZARD, which now shows the tables.

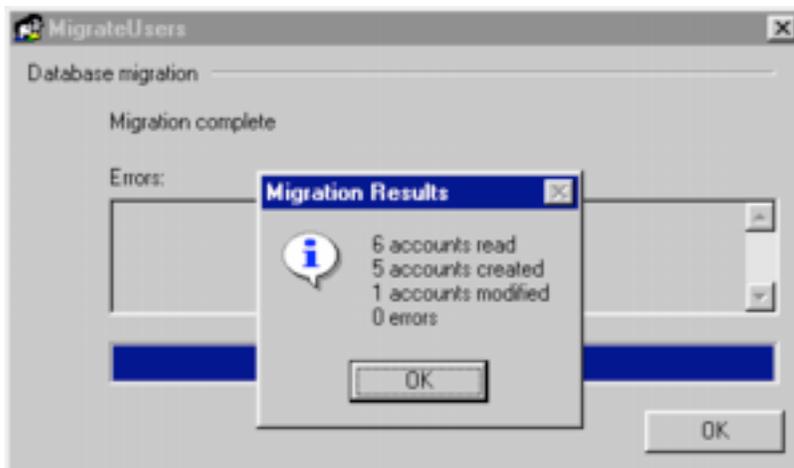


14. Use the FINISH command button to return to the Account Database Options dialog.



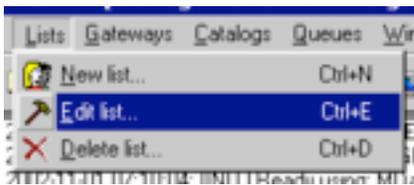
**Important:** The next step moves all of the user accounts to the newly created tables. You can move the accounts back to USERLIST.DAT by coming back into this dialog. You cannot move accounts from one ODBC data source to another directly. You must move the accounts back to USERLIST.DAT first and then move them to another database.

15. Use the OK command button to move all of the user accounts to the newly created tables.



## Mailing List

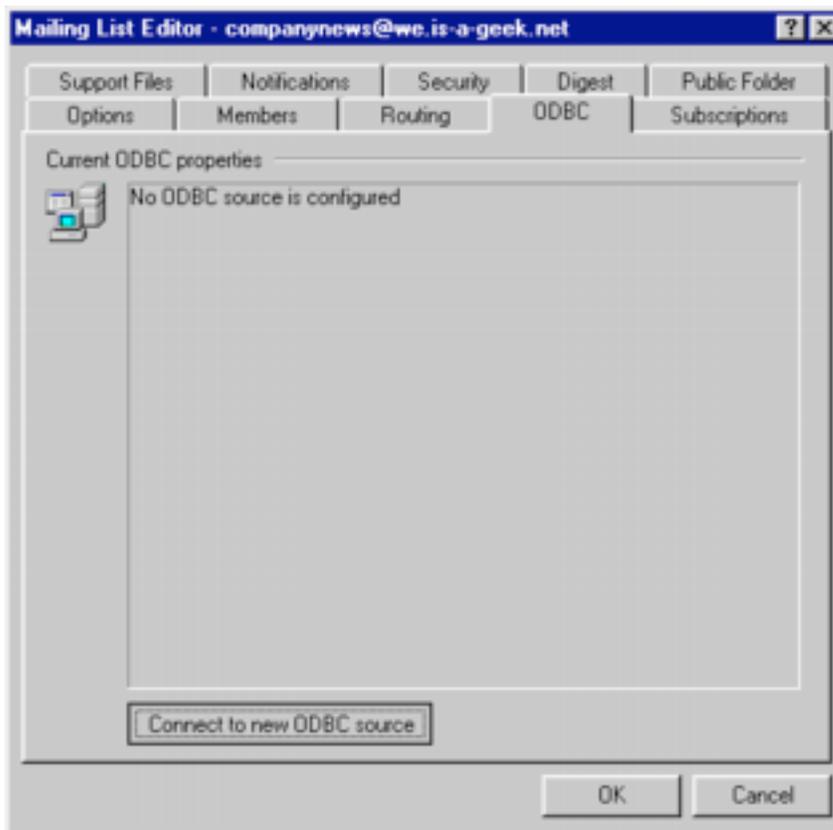
The instructions begin with the MDAEMON ADMINISTRATOR GUI running. They show how to add an ODBC data source to an existing mailing list. The database containing email addresses and names must already exist.



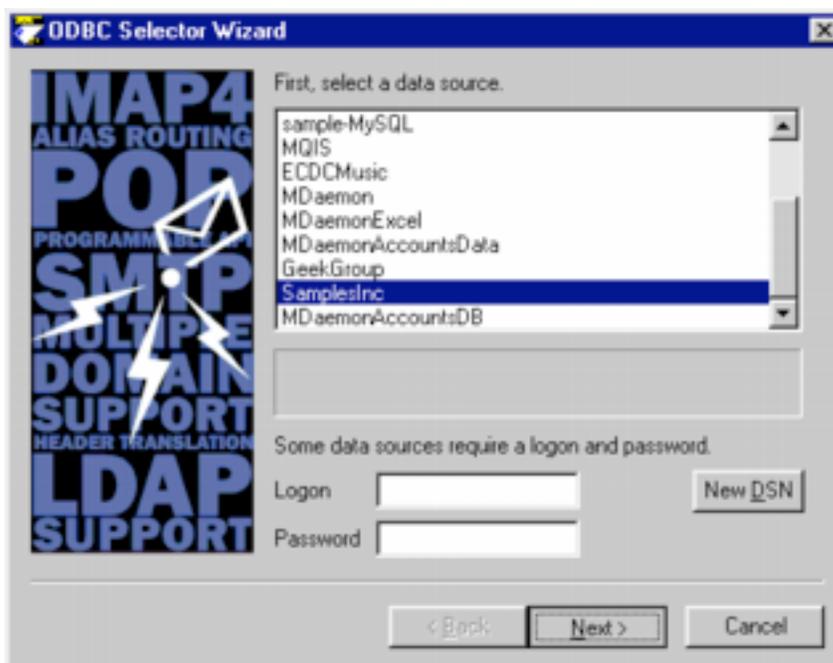
1. Use the EDIT LIST command from the LISTS menu. This displays the SELECT MAILING LIST dialog.



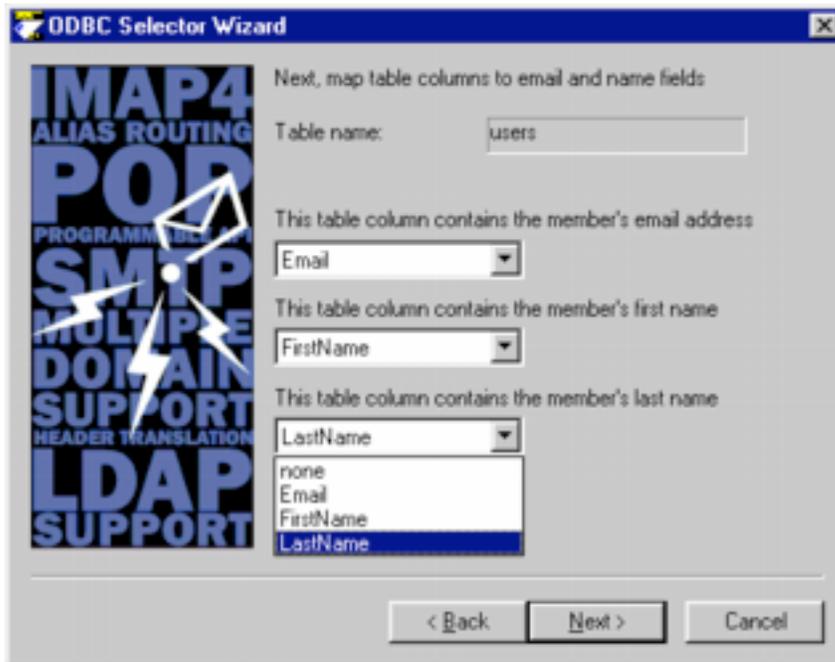
2. Choose the list you want to edit and use the OK command button to display the MAILING LIST EDITOR dialog.



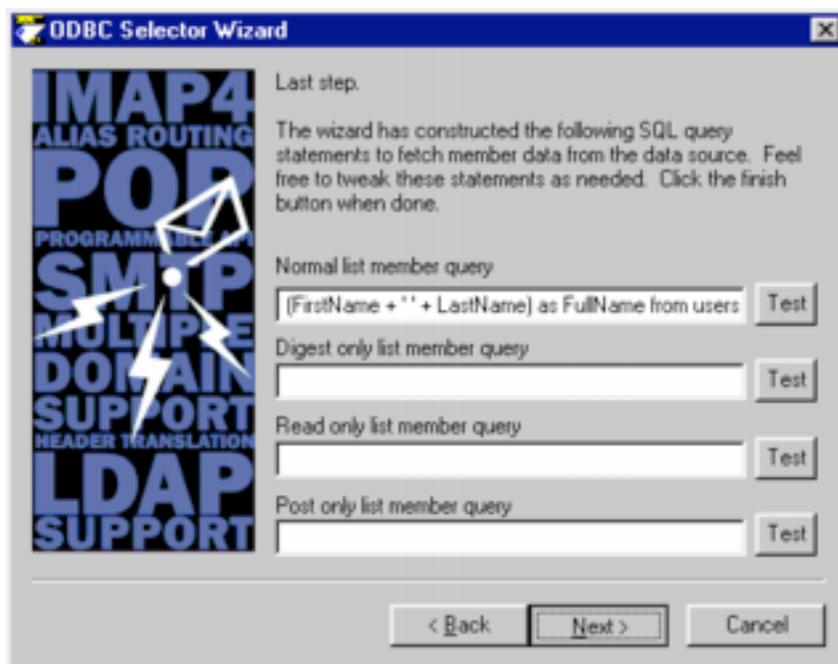
3. Select the ODBC tab and use the CONNECT TO NEW ODBC SOURCE command button to display the ODBC SELECT WIZARD.



4. Choose from the list the database you want (logon and password are optional) and use the NEXT command button. This displays the list of tables in the database.

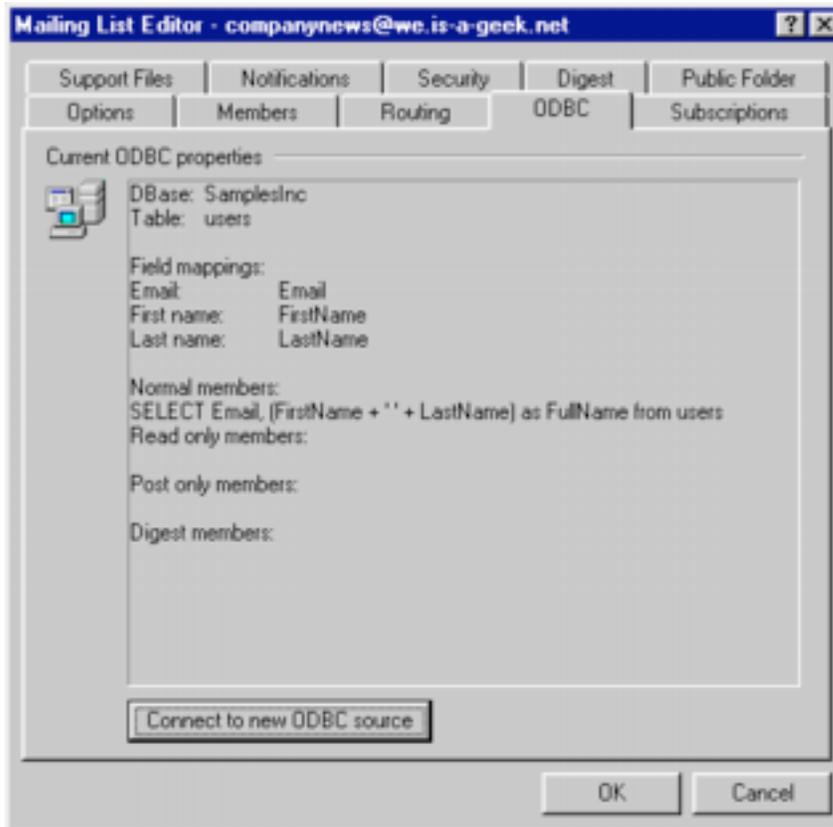


Choose the table you want and use the Next command button to display a dialog for selecting the table columns for the email address, first name and last name. This displays a dialog containing an SQL statement for retrieving the data.



On this dialog you can modify and test the SQL statement. Then you can use the Next command button to display a dialog for completing the connection.

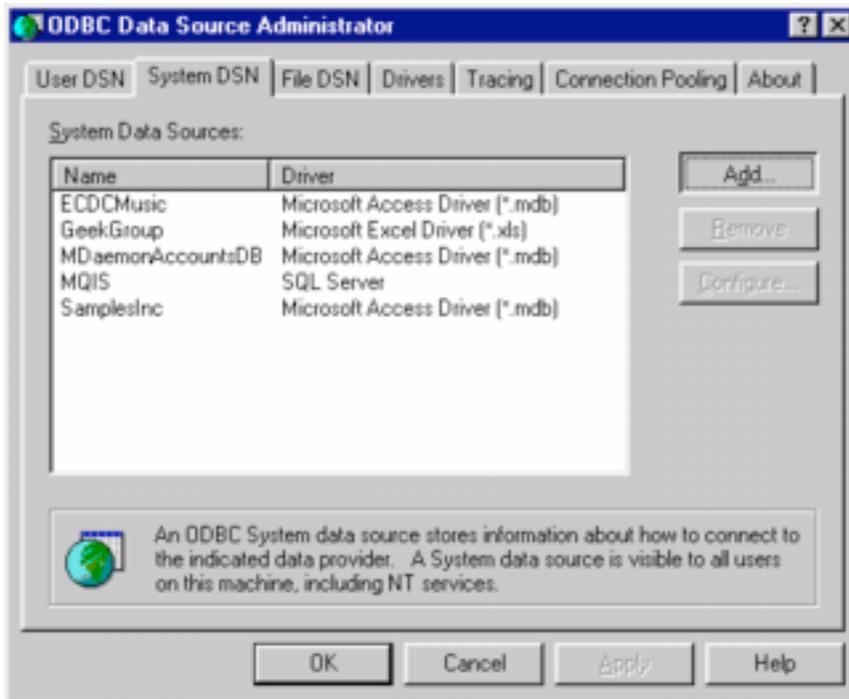
5. Use the NEXT command button to display a dialog for completing the connection.
6. Use the FINISH command button to see the datasource information for the list.



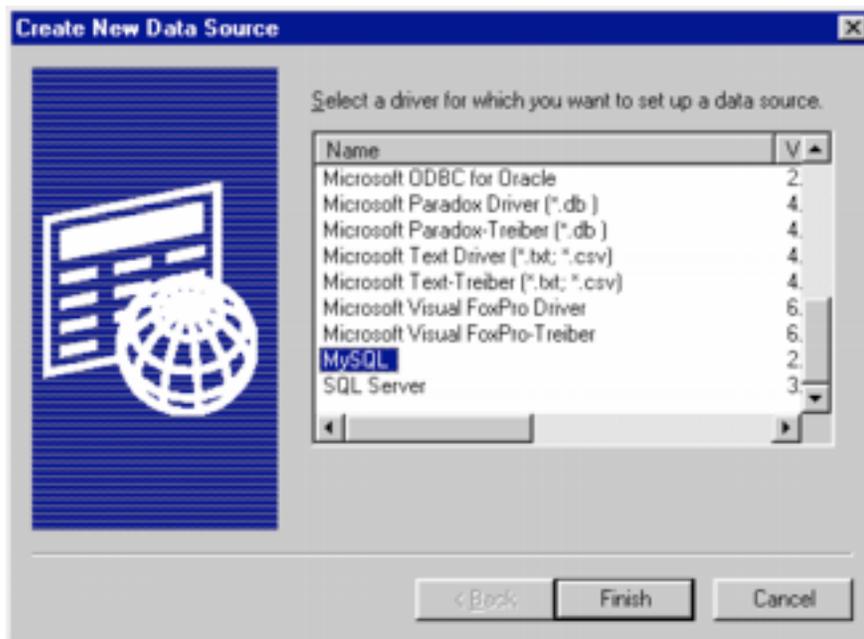
## MySQL Sample Instructions

### User Accounts

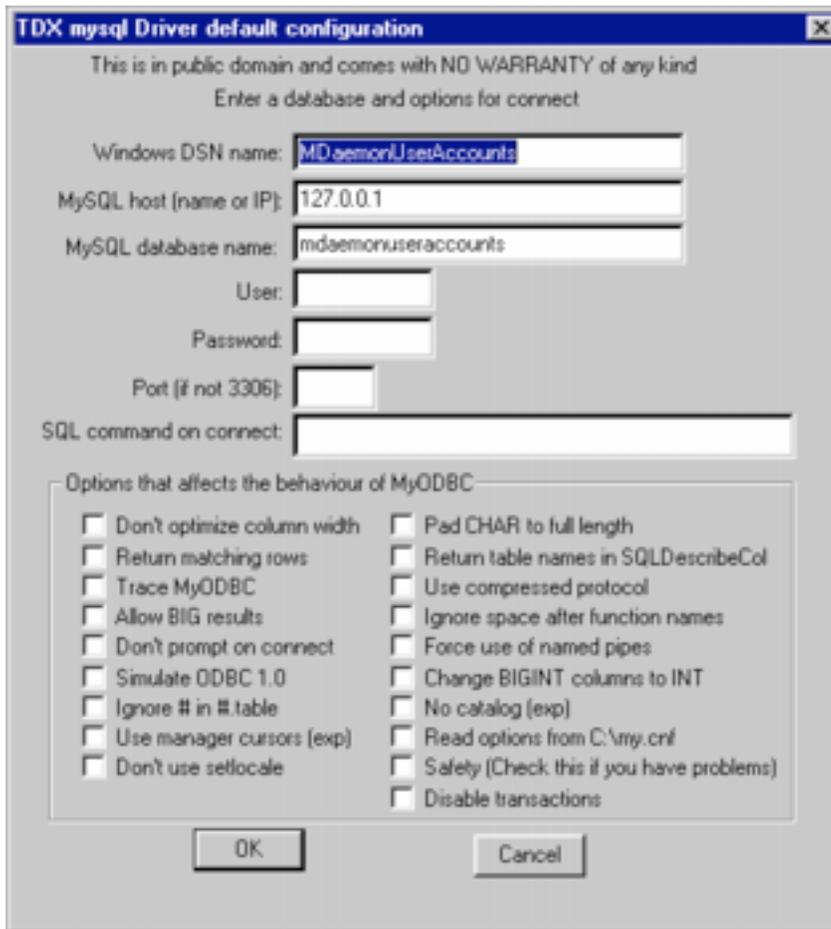
These instructions begin with setting up the MySQL data source using the ODBC Data Source Administrator control panel. The instructions assume you have previously (1) installed MySQL on a local or network machine, (2) created a MySQL database and (3) installed MyODBC on the local machine. The detailed procedures for these three prerequisites differ somewhat depending on the operating system you are using and are beyond the scope of this document.



1. Click the SYSTEM DSN tab and use the ADD command button. This displays the CREATE NEW DATA SOURCE wizard.

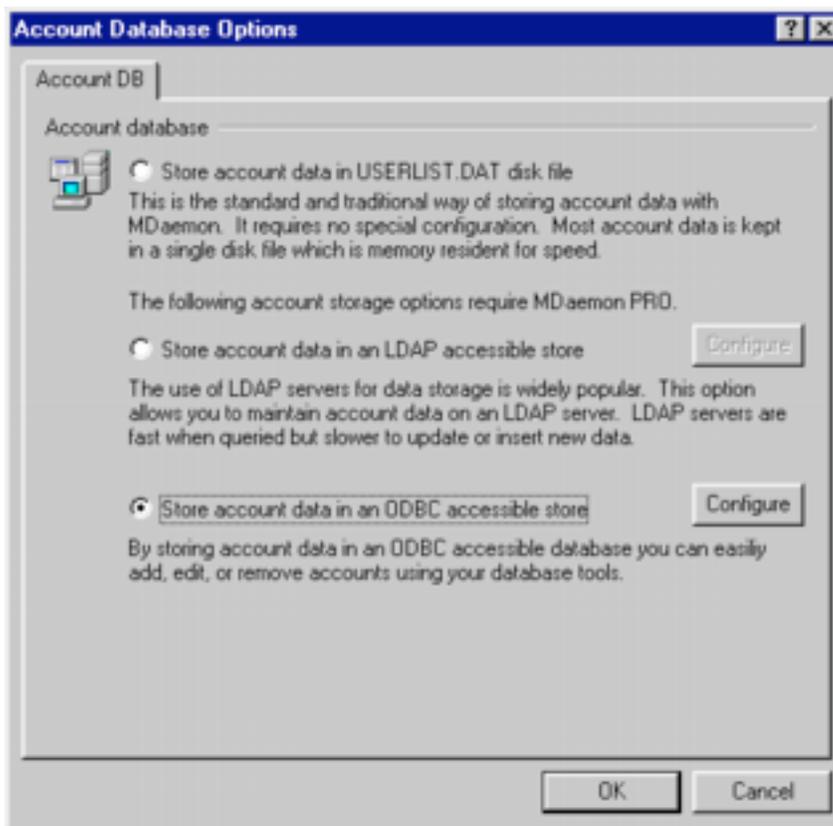


2. Select MySQL as the ODBC driver and use the FINISH command button. This displays a dialog for configuring the MySQL ODBC driver.

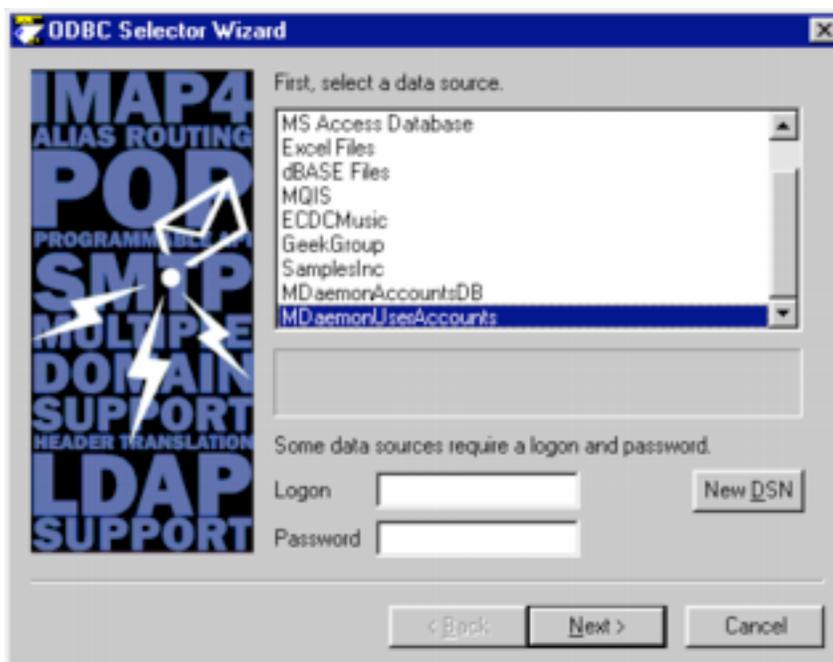


3. Enter content for WINDOWS DSN NAME, MYSQL HOST and MYSQL DATABASE NAME. MySQL host must be the IP address or DNS name of the server running MySQL. — LOCALHOST or 127.0.0.1 works for the local machine. Then use the OK command button to complete setting up the ODBC data source.
4. Use the OK command button to close the ODBC DATA SOURCE ADMINISTRATOR.

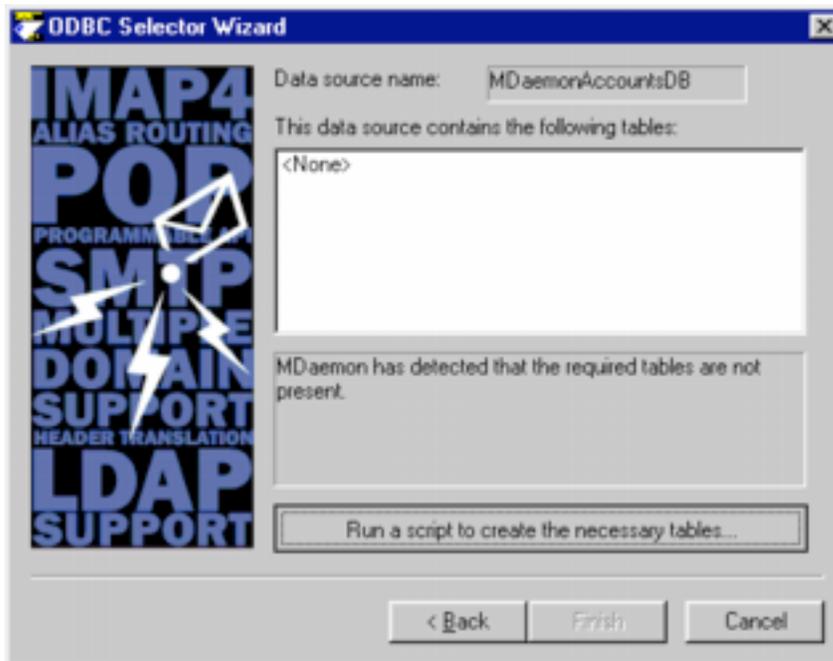
- In the MDAEMON GUI use the ACCOUNT DATABASE command from the ACCOUNTS menu. This displays the ACCOUNTS DATABASE OPTIONS dialog.



- Choose the STORE ACCOUNT DATA IN AN ODBC ACCESSIBLE STORE radio button and use the CONFIGURE command button. This displays the ODBC SELECTOR WIZARD.



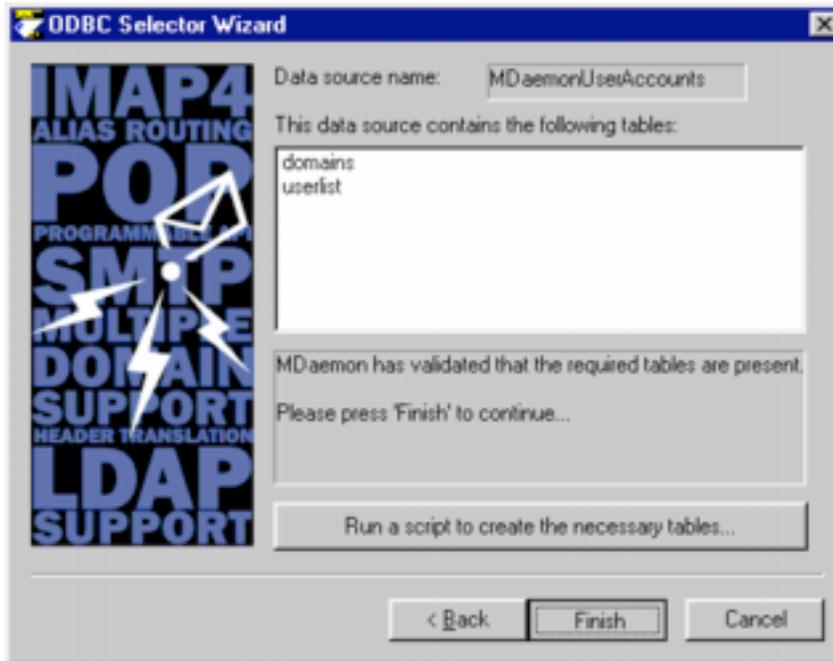
7. Select the MySQL DATASOURCE you created through the ODBC Data Source Administrator control panel and use the NEXT command button. This displays a dialog showing there are no tables in the database. The tables must be DOMAINS and USERLIST.



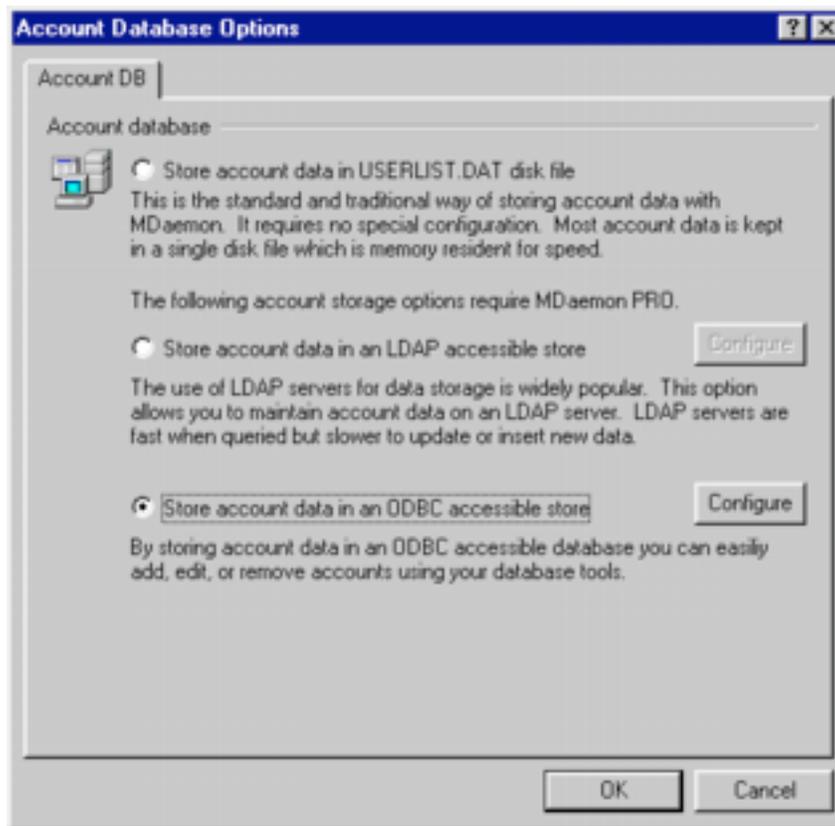
8. Use the RUN A SCRIPT TO CREATE THE NECESSARY TABLES command button to display a dialog for creating the tables by running an SQL script.



9. Use the BROWSE button to locate and choose the ACCOUNTSDB-MYSQL.SQL script. Use the RUN SCRIPT AND CREATE TABLES NOW command button to create the tables. Then use the CLOSE command button to return to the ODBC SELECTOR WIZARD, which now shows the tables.



10. Use the FINISH command button to return to the Account Database Options dialog.



**Important:** The next step moves all of the user accounts to the newly created tables. You can move the accounts back to USERLIST.DAT by coming back into this dialog. You cannot move accounts from one ODBC data source to another directly. You must move the accounts back to USERLIST.DAT first and then move them to another database.

11. Use the OK command button to move all of the user accounts to the newly created tables.

# Questions, Exclamations and Answers

## User Account Questions

### What is the data structure for the MDAemon ODBC interface?

The data structures are defined in several SQL scripts distributed with MDAemon. The scripts are specific to various popular database management systems. They are in the ~\MDaemon\App directory and have an *sql* extension, such as ACCOUNTSDB-ACCESS.SQL. Other scripts may be available for data management systems (DBMS) such as MySQL, PostgreSQL and SQLServer. But there is not a script for every DBMS.

This is an example of a script for Microsoft Access for illustration purposes only. It may not match the distributed script:

```
CREATE TABLE Domains (  
    [DomainName] varchar(66) NOT NULL,  
    [DomainID] counter NOT NULL,  
    PRIMARY KEY ([DomainID]),  
    CONSTRAINT IdxDomainConstraint UNIQUE ([DomainName])  
)  
GO
```

```
CREATE TABLE UserList (  
    [UserID] counter NOT NULL,  
    [Mailbox] varchar(60) NOT NULL,  
    [FullName] varchar(100) NOT NULL,  
    [MailDir] varchar(255) NOT NULL,  
    [Password] varchar(32) NOT NULL,  
    [AutoDecode] bit NOT NULL,  
    [IsForwarding] bit NOT NULL,  
    [AllowAccess] bit NOT NULL,  
    [AllowChangeViaEmail] bit NOT NULL,  
    [KeepForwardedMail] bit NOT NULL,  
    [HideFromEveryone] bit NOT NULL,  
    [EncryptMail] bit NOT NULL,  
    [ApplyQuotas] bit NOT NULL,  
    [EnableMultiPop] bit NOT NULL,  
    [CanModifyGAB] bit NOT NULL,  
    [MaxMessageCount] int NOT NULL,  
    [MaxDiskSpace] int NOT NULL,  
    [DomainID] int NOT NULL,  
    PRIMARY KEY([UserID]),  
    CONSTRAINT IdxUserConstraint UNIQUE ([Mailbox], [DomainID]),  
    FOREIGN KEY ([DomainID]) REFERENCES Domains  
)  
GO
```

### How do I switch back to userlist.dat for my user accounts?

1. Choose the ACCOUNT DATABASE command from the Accounts menu. This displays the ACCOUNT DATABASE OPTIONS dialog.

2. Select the STORE ACCOUNT DATA IN USERLIST.DAT DISK FILE radio button and use the OK command button. This copies all of the accounts from the current data store and activates USERLIST.DAT.

### **How can I change the database used to store my user accounts?**

MDaemon attaches to one user account database at a time. You can copy your user accounts from your current database to USERLIST.DAT and then copy the user accounts from USERLIST.DAT to the new database. To do this, use the ACCOUNT DATABASE command in the ACCOUNTS menu to access these functions.

### **There is no script for creating the user account tables for my DBMS!**

Build your tables by using one of the provided scripts as an example. The table names must be DOMAINS and USERLIST. The tables must contain the data elements listed in the distributed scripts. The scripts are located in the ~\MDAEMON\APP directory and have *sql* extensions.

## **Mailing List Questions**

### **What tables must I use for the mailing list data source?**

You can use any table. The mailing list The table must contain the email address as a minimum. For list management purposes, having the first and last name of the email recipient is helpful.

### **My data does not match the mailing list wizard data elements!**

Select as much as you can from the wizard interface, then use the Next command button. This displays a panel for testing and modifying the SQL query. Some knowledge of SQL is useful.

### **Must I be an SQL expert to use the mail list interface?**

No. If your data source has individual data elements for email address, first name and last name, you only need to select these in the wizard. A limited amount of SQL knowledge is useful if, for example, your email address consists of two data elements such as *username* and *domain*.

To illustrate the difference, if your email address is stored in one data element named *emailaddress*, for example, the wizard puts this into the SQL query:

```
SELECT emailaddress as Email
```

If your email address is stored as *username* and *domain* you would select only one data element, let the wizard generate the SQL such as this:

```
SELECT username as Email
```

then modify it to be this:

```
SELECT (username+'@'+domain) as Email
```

You have added the second data element, put the elements in parenthesis and added the email @ sign between + signs.