

# Creating Database Objects with Microsoft Jet

---

In this chapter, I'm going to show you how to use the Visual Data Manager add-in to Visual Basic to create a Microsoft Jet database and various objects, such as tables and indexes, inside it. I'll also cover how to create users and groups, and how to grant them permission to access database objects.

## Introducing the Visual Data Manager

The Visual Data Manager (also called VisData) is a Visual Basic add-in that was written to make it easier to design a Jet database without building your own application or using Microsoft Access. To run the Visual Data Manager, choose Add-Ins ⇨ Visual Data Manager from the main menu. This will start the application.



Tip

**You got it:** The Visual Data Manager utility was written in Visual Basic 5 and shows what you can do with a little bit of work. The complete source code to the Visual Data Manager is included in the `Samples\VisData` directory of Visual Basic.

Since the Visual Data Manager is based on DAO, you can use it to open many other types of databases besides Jet databases. You can access dBase, FoxPro, and Paradox databases, as well as Excel worksheets and various types of text files. You can even open an ODBC connection to a remote database, such as



### In This Chapter

Introducing Visual Data Manager

Constructing databases, tables, and indexes

Managing security for users and groups



SQL Server or Oracle, though depending on your choice of database, some of the capabilities may not be supported.

Note

**Why doesn't it open the database I created with Access 2000?:** Visual Data Manager uses DAO 3.5 to access databases. To access a Jet 4.0 database created with Access 2000, you must use DAO 3.6. While you could recompile the Visual Data Manager application using DAO 3.6, it's probably just easier to use Access to design your database.

## Opening an existing database

To open an existing database, choose File ⇨ Open DataBase ⇨ Microsoft Access from the Visual Data Manager main menu. This will display a File Open dialog box. Simply choose the file containing your database and press the Open button. After you've opened your database, you'll see a form like the one shown in Figure 30-1. By default, the Visual Data Manager opens your database and displays two multiple document interface (MDI) windows. The first window (Database Window) displays your database's structure, while the second window (SQL Statement) allows you to enter and run an SQL statement.

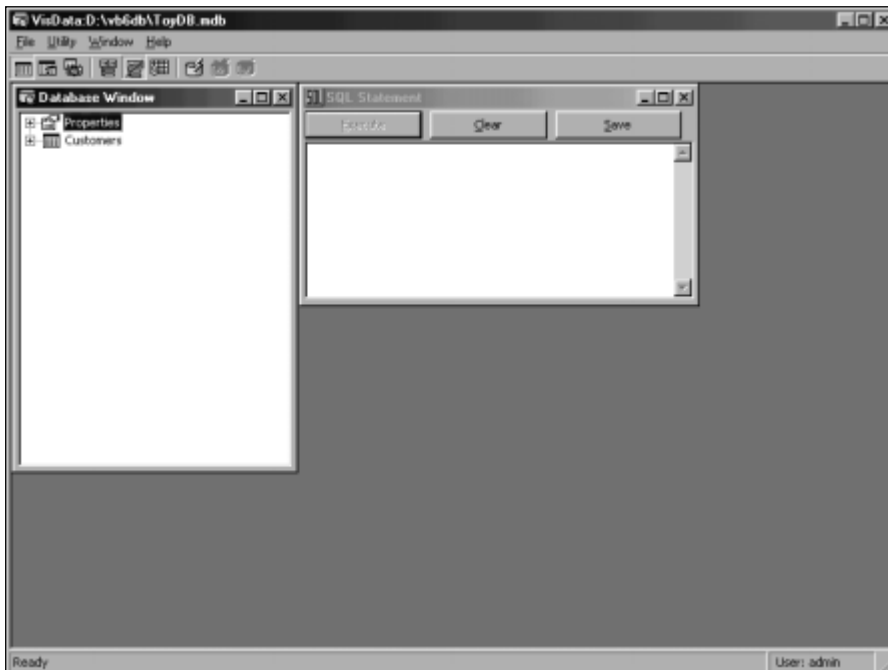


Figure 30-1: Viewing an open database.

## Viewing Database Information

The Visual Data Manager allows you to see the structure of your database (see Figure 30-2). The tree view displays each table and its attributes, including columns and indexes. As you expand each entry in the table, you will eventually reach the details that would normally show up in the Properties collection associated with the object.

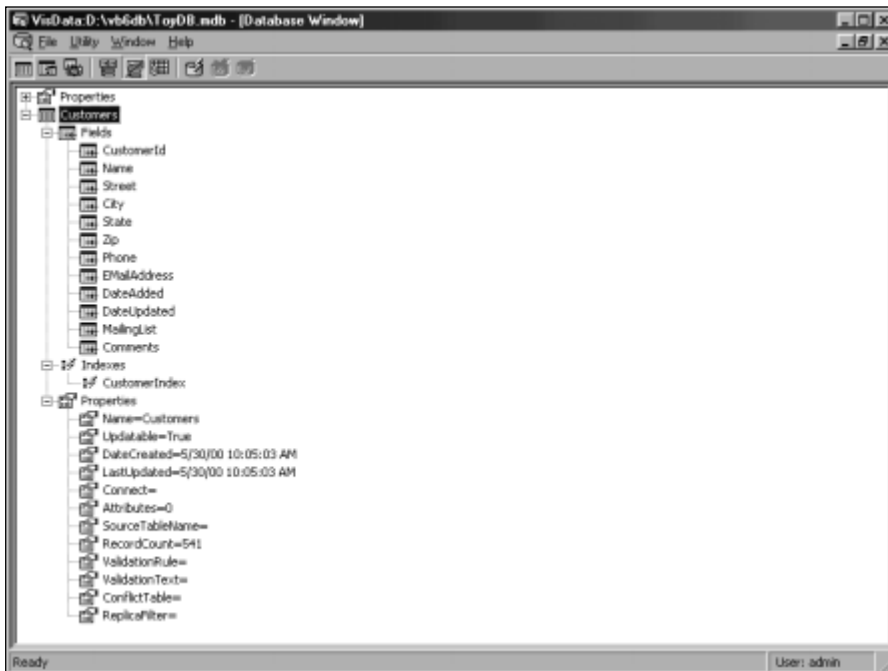


Figure 30-2: Viewing the structure of your database.

## Running an SQL query

Another useful tool included in the Visual Data Manager is the ability to execute an SQL query against the current database. You can enter the query directly in the SQL Statement window (see Figure 30-3) or use the Query Builder to create the query (see Figure 30-4). Either way, when you run the query, your results will be displayed one row at a time in the Results window shown in Figure 30-5.

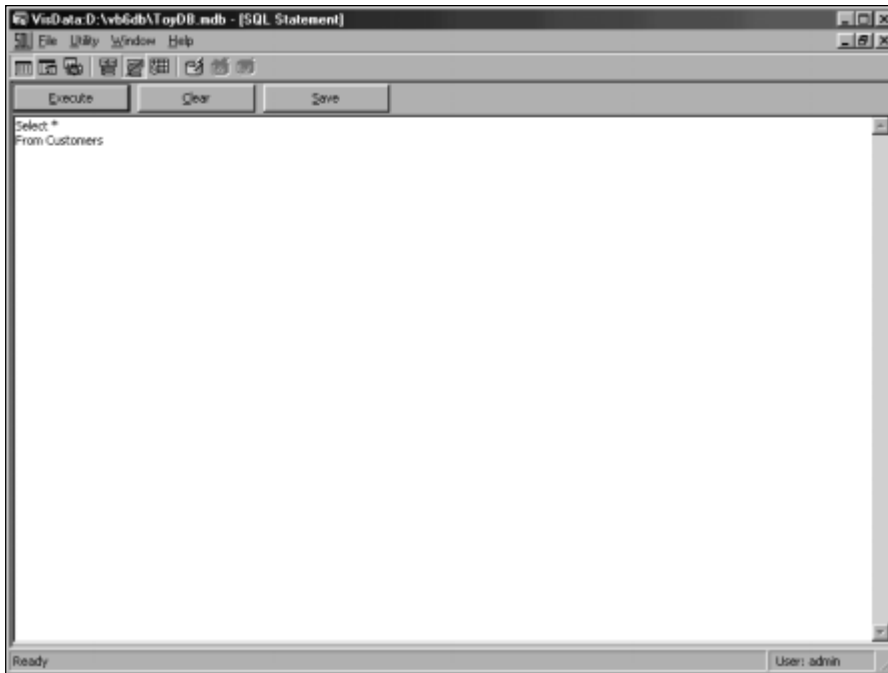


Figure 30-3: Entering an SQL statement in the SQL statement window.

## Using the Query Builder

Choosing Utility ⇨ Query Builder from the Visual Data Manager main menu starts the Query Builder (see Figure 30-4). This utility assists you with building a **Select** statement. You begin building the **Select** statement by choosing the table you wish to start with in the Tables section of the form, and the columns you want to display in the Fields to Show section of the form. If you don't select any columns, all of the columns will be returned.

By default, all of the rows in the table will be returned. However, you can use the Field Name, Operator, and Value fields to select a value or range of values for a specific column. List Possible Values will populate the Value drop-down box with all of the values for that specific column in the table.

Once you've identified a field, operator, and value, you should press the And into Criteria button or the Or into Criteria button to include this expression in the Criteria section at the bottom of the form. Then you can select other criteria by selecting a new column, operator, and value combination. Pressing the And into Criteria button will insert an **And** operation between the two criteria, while pressing the Or into Criteria button will insert an **Or** operation.

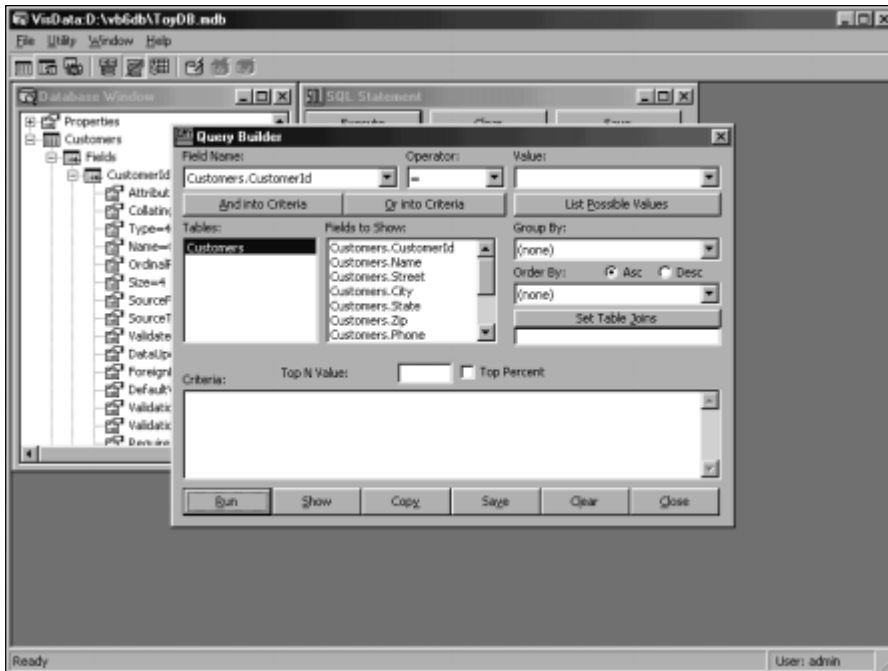


Figure 30-4: Building a query using the Query Builder.

You can choose to include **Group By** and **Order By** clauses by using the Group By and Order By drop-down boxes. If you have selected multiple tables, pressing the Set Table Joins button will display a dialog box that helps you select the join columns. The Top N Value field allows you to limit the number of rows retrieved by specifying an absolute number of rows or a relative percentage value.

When all of the information is entered into the form, you can press the Show button to see the **Select** statement you just built. Pressing Save will save the **Select** statement as a QueryDef in the database for use with Access, while pressing Copy will save a copy of the **Select** statement to the clipboard so that you may paste it into a different application. Pressing Close will close the Query Builder window and place the query you built in the SQL Statement window.

### Getting your results

If you pressed the Enter key or clicked on the Execute button in the SQL Statement window, or clicked on the Run button in the Query Builder, the **Select** statement will be executed and the results will be returned in the Results window, as shown earlier in Figure 30-5.

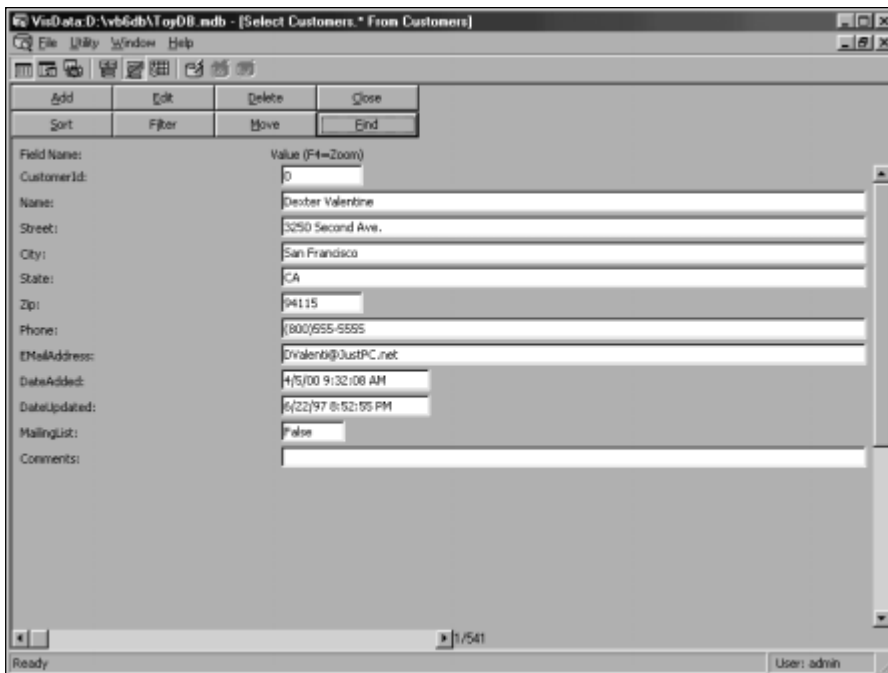


Figure 30-5: Seeing the results from running your query.

One row at a time will be displayed in the form. Clicking on the scroll bar at the bottom of the form will move you to another row that was retrieved from the database. The buttons at the top of the form perform the functions you would expect. Pressing the Add button will display an empty version of the Edit form in which you may enter the information you want to add to the database. Again, press Update to save your new row, or Cancel to discard it. Pressing the Edit button will change the configuration of the form slightly (see Figure 30-6). You can change any of the values and press the Update button to save the changes to the database, or press the Cancel button to discard the changes. The Delete button is used to delete the current row. Pressing the Close button will close the Results window and return you to the previous window where you created the query. Clicking on the Sort button will display a message box prompting you to enter the sort column, while clicking on the Filter button will display another message box prompting you to enter a filter expression. The Move button will allow you to specify the number of rows to be moved. The Find button displays a dialog box to help you find a particular row in the query. You must select a single field, a relational operator, enter a value and then choose whether you want to find the first value, next value, previous value or the last value in the table. Pressing the OK button will start the search and the record will be displayed in the main form.

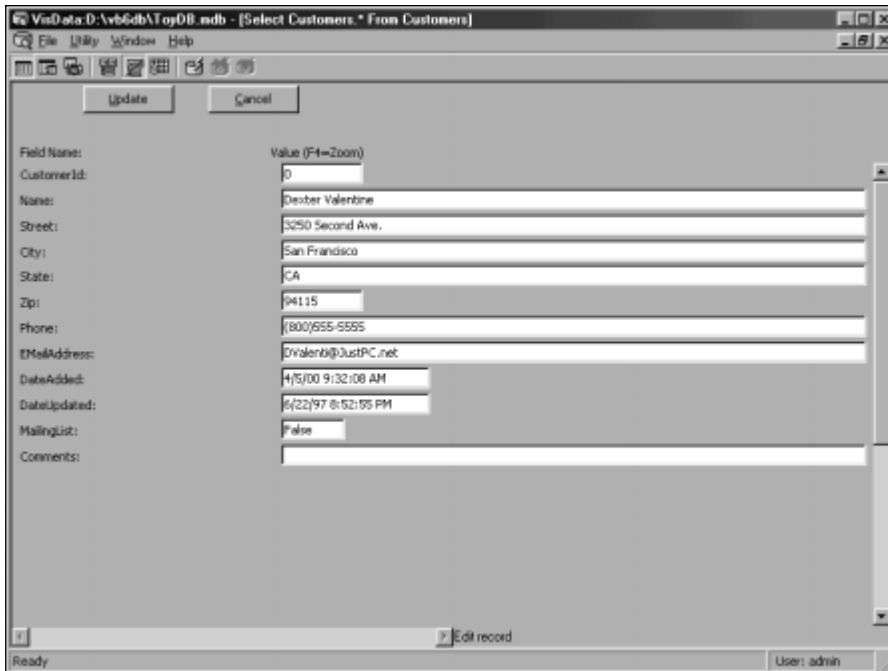


Figure 30-6: Editing a row in your database.

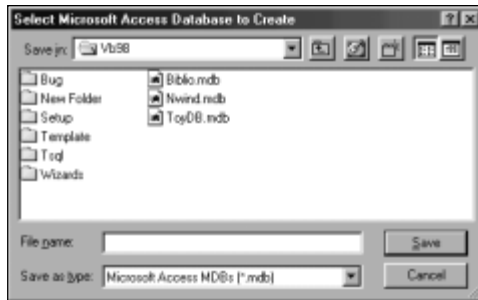
## Constructing Databases, Tables, and Indexes

Probably the most common use for the Visual Data Manager utility is to create Jet database objects without using Access and without building your own application program.

### Creating a new database

The following steps will show you how to use the Visual Data Manager to create a new database:

1. Start the Visual Data Manager and choose **File** ⇨ **New** ⇨ **Microsoft Access** ⇨ **Version 7.0 MDB** from the main menu. This will display the **Select Microsoft Access Database to Create** dialog box (see Figure 30-7). Unless you have a specific need for an obsolete database, don't create a **Version 2.0 MDB**.



**Figure 30-7:** Choosing the name of your new database.

2. Using the Select Microsoft Access Database to Create dialog box, choose a name for your database and press Save to create it.



Tip

**An empty database is a good thing:** Because your database file is really just another disk file, I find it useful to keep an empty copy of the database file around when I'm developing an application. I can just copy the .MDB file to another file and use that to test my application, without having to go through the process of re-creating the database. Also, you may want to keep a populated database around with a set of known values. You can create a new copy of this file to test your application, while leaving the original populated file untouched for the next time you want to do some testing.

## Creating a table

After creating a new database or opening an existing database, you can easily create a new table by following these steps:

1. While in the Visual Data Manager, select the Database window. Right click anywhere in the window and choose New Table. This will display the Table Structure window, as shown in Figure 30-8. Note that you can use the same window to display information about an existing table by right clicking on the table's name in the Database window and choosing Design from the pop-up menu.
2. Enter the name of the table in the Table Name field. Note that your table name should not duplicate any tables already defined in your database.



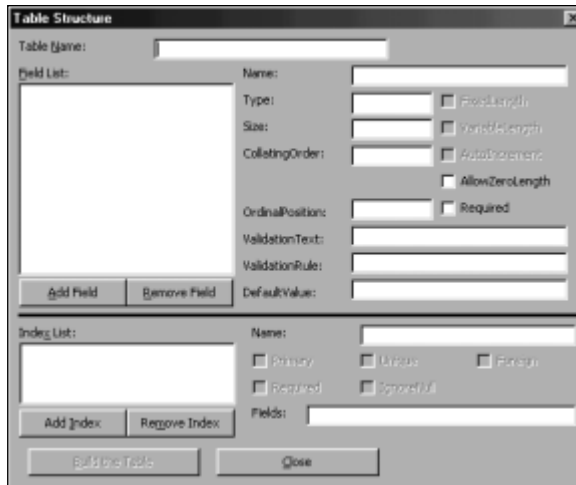


Figure 30-8: Creating a new table with the Table Structure window.

- To add fields to the table, press the Add Field button. This will display the Add Field dialog box, as shown in Figure 30-9. Enter the name of the column in the Name field and select its data type in the Type drop-down box. You can ignore the OrdinalPosition box, since this value will be computed when the field is added. You may also specify a validation rule in the ValidationRule box. If you do so, you may also indicate the text that will be displayed when the validation rule fails in the ValidationText box. The value you enter in the DefaultValue text box will be used, instead of a **Null** value, whenever a new record is created. When you have entered the information for a field, press the OK button. The form will be cleared and you can enter the next field. When you have finished entering all of the fields, press the OK button to save the last field and then press the Close button to return to the Table Structure window.

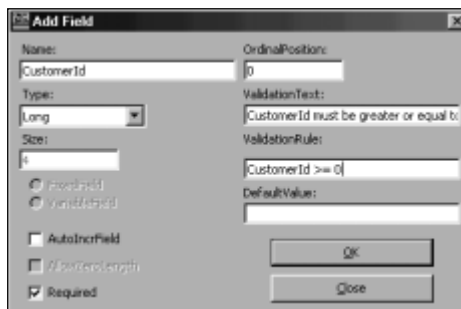


Figure 30-9: Adding a field to your table.

4. You can view the information associated with a field by clicking on the field's name in the Field List box. The fields to the right of the Field List will be populated with the information. You can change any of the information associated with a field, except for its data type and related information. If you wish to change this information, you must delete the field and add it back again.
5. When you have entered all of the fields, you may press the Close button, or continue this process by adding an index.

## Creating an index

In order to create an index, you must have an existing table. You can create an index at the same time you create the table, or you can add an index to an existing table by right clicking on the table name in the Database window and selecting design from the pop-up menu to display the Table Structure window shown in Figure 30-9. Then follow these steps:

1. Press the Add Index button. This will display the Add Index dialog box, as shown in Figure 30-10.



Figure 30-10: Adding an index to a table.

2. Enter the name of the index in the Name field. Choose Primary, Unique, and IgnoreNulls as appropriate. Just remember that you may have only one Primary index, and it should always be unique.
3. Add fields to the index by double-clicking on the field name in Available Fields. The field name will be added to the list of field names in the Indexed Fields field. If you make a mistake, you can delete the field by simply selecting the text in the Indexed Fields box and pressing the Delete key on the keyboard.
4. When you have finished adding your index, press the OK button. The form will be cleared so that you may add another index. If you are finished adding indexes to this table, press the Close button to return to the Table Structure window. Press the Close button on the Table Structure window to return to the Database Window.

## Using the DAO and ADOX alternatives

You can also create database objects in a Jet database by using either DAO or ADOX (the ADO Extensions for Data Definition Language and Security). While the code to do this is very straightforward, it can be long and repetitive and not easily updated if your database design is still in a state of flux. However, having the ability to create a new, empty database from within your application can be useful in a single user application. This allows a user to easily create multiple, independent data files containing different sets of data.

A small icon of an open book with the text "Cross-Reference" written on it.

See Chapters 6 and 11 for more information about the DAO and ADOX object models, respectively.

## Managing Security

Every database system needs security, even if you plan to allow only one user to access it. If you have Access installed, managing security isn't very difficult, and this section discusses how to use Access 2000 to manage security for your Jet 4.0 database. Similar features exist in Access 97 in order to manage your Jet 3.5 database.

### The security definition file

The first step in implementing security on a Jet database is to ensure that you have a workgroup database (SYSTEM.MDW file). This file contains information on the users and groups that secure a Jet database. By default, the SYSTEM.MDW file is created when you install Access 97 or Access 2000. Typically, this file is located in the `\Windows\System` directory; however, it can be located anywhere on your system, since a value in the Windows registry is used to locate the file.

If your application is running in a networked environment, where the database is shared among multiple users, you should make sure that the client systems point to a shared copy of the SYSTEM.MDW file. You can do this in DAO by specifying the `SystemDB` property in the `DBEngine` object, or by adjusting the windows registry key to point to the file.

### Managing users and groups

Since the workgroup file is independent of any database file, you can manage the users and groups in Access without opening a database. You simply use the User and Group Accounts dialog box, which has the ability to view and change information about existing users and groups, as well as create new users and groups.

## Managing users

To see the information about an existing user, start Access 2000 and choose Tools ⇨ Security ⇨ User and Group Accounts to display the User and Group Accounts dialog box (see Figure 30-11). Choose the user that you wish to view by selecting the user from the Name drop-down box.



Figure 30-11: Displaying the User and Group Accounts dialog box.

### Note

**Hidden secrets:** While you can't see the password for a user, you can press the Clear Password button to remove the password from the user if you are a member of the Admins group, or you can switch to the Change Logon Password tab to change the password to a new value.

You can add a new user by pressing the New button in the User frame. This will display the New User/Group dialog box (see Figure 30-12). Enter the name of the user and a code phrase that will make the user name unique.



Figure 30-12: Creating a new user or group.

## Ousting Admin

Removing Admin from power is one of the most important things you can do when you implement user-level security for your database. Unless you take the proper steps, it would be easy for a disgruntled employee to subvert your entire security system. Even if you trust everyone that uses your database, you should still take this precaution. Someone may take advantage of your open security and try to help you improve your database. Even with the best of intentions, and even if the individual has more knowledge about databases than you do, do you really want someone changing your application without your knowledge? A healthy paranoia about security is very important and helps to prevent problems in the long run.

### Note

**Make personal ids personal:** Given that you can easily substitute one workgroup file for another, it is important that user ids and groups be made universally unique. The key is supplying a value for the personal id that would not be easily guessed.

## Managing groups

By switching to the Groups tab in the User and Group Accounts dialog box, you can see the list of security groups defined in the workgroup file (see Figure 30-13). You can create a new group by pressing the New button and entering the group and personal id as you did when you created a new user.



**Figure 30-13:** Viewing the groups in your workgroup file.

## Managing permissions and ownership

Information about permissions to access a database object and the ownership of this object is stored in the system tables of your database. This information is really based on SIDs and not the actual user id or group name. Thus, your user id must have already been authenticated by using the workgroup file.

### Managing permissions

Choose Tools ⇨ Security ⇨ User and Group Permissions from Access's main menu will display the User and Group Permissions dialog box shown in Figure 30-14.

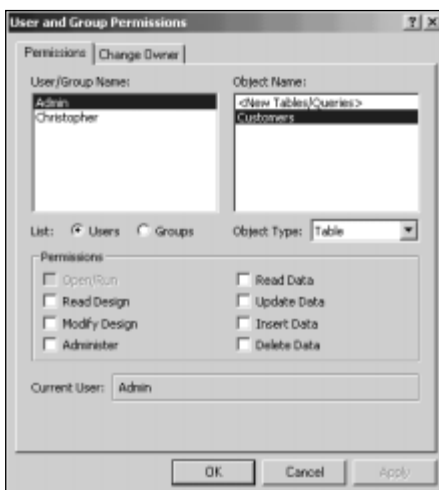


Figure 30-14: Editing a user or group's permissions.

### Changing ownership

You can switch to the Change Owner tab of the User and Group Permissions dialog box to change the owner of a database object to a different user id or group (see Figure 30-15). The Object and Current Owner lists show all of the objects in your database that match the object selected in the Object Type drop-down box.

To change the ownership of the object, you must begin by selecting the database object you wish to change. Then you must decide if the new owner will be a group or a user and click on the appropriate radio button. The list of possible owners will be displayed in the New Owner drop-down box. Choose the user id or group and press the Change Owner button to make the change.

**Caution**

**Don't forget the database:** The database itself is an object that is owned by a user id or a group. By default, Admin owns the database. You should change this to the Admins group. Otherwise, the Admin user id is still all-powerful.



**Figure 30-15:** Changing the ownership of a database object.

## Thoughts on Creating Jet Databases

Jet based Visual Basic applications are usually targeted at either individuals or small organizations with only a handful of users. This means that many of the approaches that I have discussed throughout the book may not apply. For instance, security in a small application is not nearly the problem it is in a larger application. For a single user application, the primary aim of security is to prevent the user from accidentally deleting information needed to run the application. Even with a handful of users, you are more concerned with identifying the actions of individual users rather than protecting the database from malicious actions.

Because of these differences, you might want to add some features that you wouldn't consider in larger applications. For instance, you may want to add a feature in your application to create a new copy of the database. This would let the user create multiple databases for different projects.

Another feature you might consider adding is a way to create a backup copy of the database. This means that the user has a way to make sure that their data is save in case of a system problem. Of course, creating a way to backup the database implies that a way to restore it will be necessary at some point.

## Summary

In this chapter you learned:

- ♦ how to create databases, tables using Visual Basic tools for Jet 3.5 databases.
- ♦ how to use Access 2000 to create databases and tables for Jet 4.0.
- ♦ how to use Access 2000 to define security in a Jet 4.0 database.

