

# Visual Basic Data Manager Help

[Other Sources of Information](#)

[Introduction to the Visual Basic Data Manager](#)

[Adding Fields to a Table](#)

[Closing a Database](#)

[Compacting a Database](#)

[Creating a Database](#)

[Creating an Index](#)

[Creating New Tables](#)

[Deleting an Existing Table](#)

[Modifying Table Data](#)

[Opening a Database](#)

[Repairing a Database](#)

## Introduction to the Visual Basic Data Manager

The Visual Basic Data Manager allows Visual Basic users to create new databases (.MDB format) and examine or map the structure of existing external databases in a variety of formats. Formats that you can either create or modify with the Data Manager are shown below:

<b>Database Format</b>	<b>Create</b>	<b>Modify</b>
Microsoft Access 1.0	Yes	Yes
Microsoft Access 1.1	Yes	Yes
Paradox 3.0 and 3.5	No	Yes
dBASE III and IV	No	Yes
FoxPro 2.0 and 2.5	No	Yes
Btrieve	No	Yes

---

**Note** Visual Basic shares its database engine with Microsoft Access, so databases created with Visual Basic or the Data Manager can be manipulated using Microsoft Access. In addition, databases created with Microsoft Access can be manipulated using Visual Basic and the Data Manager.

Throughout the Visual Basic documentation, databases created with this engine are referred to as Visual Basic databases.

---

For additional information on using external databases, see the file EXTERNAL.TXT or BTRIEVE.TXT supplied with Visual Basic.

---

**Note** Many external databases exist as directories on your disk. To create databases in these formats, use the File Manager to create a directory which becomes the database. Once the database directory is created, use the Data Manager to add tables and indexes, which become files in this directory.

---

## **Closing a Database**

When you are finished using a database, you should close the database so it becomes available for others to use.

### **To close a database**

1. Choose the Close Database command from the File menu.

## Creating a Database

See Also

### To create a Visual Basic database

1. Choose the New Database command from the File menu.  
From the submenu, select the type of database you want to create (Microsoft Access version 1.0, or Microsoft Access version 1.1).
2. Enter the name of the database you want to create in the displayed dialog.

---

**Note** The file extension must be .MDB. Do not enter or choose the name of an existing .MDB file.

---

3. Choose OK to accept the new database name, or Choose Network (if available) to connect to a network drive.  
Once the database file has been created, use the Data Manager to add new tables and indexes to store the data.

### To create an external database (dBASE III, dBASE IV, FoxPro, and Paradox)

- ▶ Use the File Manager to create a directory.  
Many external databases are simply disk directories. Once the database directory is created, use the Data Manager to add tables and indexes, which become files in the directory, to store the data.

**See Also**

[Creating an Index](#)

[Creating New Tables](#)

## Opening a Database

See Also

### **To open an existing database**

1. Choose the Open Database command from the File menu.  
From the submenu, select the type of database you want to open (Microsoft Access, FoxPro, dBASE III, dBASE IV, Paradox, BTRIEVE).
2. Enter the name of the database you want to open in the displayed dialog.  
If you are opening an external database (dBASE III/IV, FoxPro, Paradox), use the dialog to select the database directory.  
To open a BTRIEVE database, use the Open Database dialog to select the desired .DDF file.
3. Choose OK to open the database, or choose Network (if available) to connect to a network drive.

Once the database file has been opened, use the Data Manager to add new tables and indexes or to modify the existing tables or data.

**See Also**

[Closing a Database](#)

[Creating an Index](#)

[Creating New Tables](#)

## Compacting a Database

Use the Compact Database command to copy all data from one database to another and to organize the data in the resulting database contiguously so that disk space can be recovered.

---

**Note** You can compact only Visual Basic and Microsoft Access databases. To remove deleted records from external databases, you must use the packing tools that are shipped with the database.

---

### To compact a database

1. Make sure the database you want to compact is closed.
2. Choose the Compact Database command on the File menu.
3. In the displayed dialog, enter the name of the database you want to compact from and choose OK. Another dialog is displayed.
4. Enter the name of the file where the new compacted database is to be copied.

---

**Warning** Do not specify the same name for both the original (step 3) and compacted (step 4) files. If you do, and the Compact Database operation is not completed, your database will be deleted. In addition, if both files are on the same disk, be sure to provide enough room for both the original and the compacted database.

---

---

**Note** To change the Locale or Encryption scheme of a database, use the CompactDatabase statement in Visual Basic.

---



## Creating an Index

See Also

Indexes improve performance and prevent duplicate records in your tables.

### To create an index:

1. Use the Data Manager to create a new database, or open an existing database.
2. From the database window, choose the New button to create a new table, or choose the Design button to modify an existing table. The Table Design window is displayed.
3. Choose the Add button in the Indexes portion of the Table Design window. The Add Index dialog is displayed.
4. Enter a name for the index in the Index Name text box.
5. Select a field name from the Fields in Table list. Choose Add (Asc) to create an ascending index, or Add (Dec) to create a descending index.

---

**Note** There is no limit to the number of fields in an index. However, the combined length of the field names in an index must be less than 254 characters.

---

6. Select the appropriate check boxes in the bottom of the dialog.
  - Select the Require Unique Index Values check box to prevent duplicate records in the table.

---

**Note** If a database contains duplicate records or Null values in an index, the Data Manager cannot create a new index with either the Require Unique Index Values or Primary Index check boxes selected.

---

- Select the Primary Index check box to indicate the index is to serve as the primary basis for sorting the table. Setting this check box also indicates that the index is to be unique.
7. Choose Done to create the index.

---

**Note** Not all fields may be used in an index. For example, MEMO or binary object fields cannot be specified in an index.

---

**See Also**

[Adding Fields to a Table](#)

[Creating a Database](#)

[Creating New Tables](#)

[Opening a Database](#)

## Creating New Tables

See Also

All the data in your database are stored in tables. A table is defined as a set of one or more *fields*. The type of data a field can contain is determined by attributes such as data type and size.

Tables are used to group related data. For example, in the sample BIBLIO.MDB database, most of the information about each book publisher is stored in the *Publishers* table. The *Titles* table contains information about books and the *Authors* table contains information about authors.

### To Create a New Table

1. Open an existing database, or create a new database.

Once a database has been opened, the Tables window is displayed.

2. Choose the New button and enter the name of the table you want to create. Choose OK to accept the new table name and display the Table Design window.

For Visual Basic databases, the table name must follow the same rules as any Visual Basic object name. For external databases, this name must comply with the rules established by the database vendor. In most cases the table name must be less than eight characters, and the first character must be a letter (not a number).

3. Create the tables fields.

4. Add any desired indexes.

5. Repeat steps 2 - 4 to create the remaining tables in your database.

6. Close the Table window using the window's Control menu.

Once you have created a database, you can write a Visual Basic application to fill the database with data or use the Data Manager to enter data into its tables.

**See Also**

[Creating an Index](#)

[Modifying Table Data](#)

## Adding Fields to a Table

See Also

### To add a field to a table

1. Open an existing database. The Tables window is displayed.
2. Select the desired table and choose the Design button.
3. Choose the Add button in the Fields grid to add fields to the table. The Add Field dialog is displayed.
4. Enter the field name.

For Visual Basic databases, the field name must follow the same rules as any Visual Basic object name. For external databases, field names must comply with the rules established by each database vendor.

5. Enter the data type for the field.

The field's data type determines what kind of information is to be stored in the field—a number or a string, for example. If the field data type is string, you must also enter a value in the Field Size text box. The field size represents the maximum number of characters that can be entered in the field. The combined size of all the fields in a table cannot exceed 2000 bytes (each character in a string field requires one byte).

In Visual Basic, the Size property returns the maximum size, in bytes, of a field.

---

### Note

You can add additional fields to a database table, but you cannot rename, modify, or delete fields without deleting the entire table and rebuilding each field. For this reason, double check your design before choosing OK in the Add Field window.

---

Repeat steps 3 – 5 to add additional fields to your table.

**See Also**

[Creating an Index](#)

[Modifying Table Data](#)

## Deleting an Existing Table

### To delete a table in a database

1. Choose Open Database from the File menu to open an existing database. The Database window is displayed.
2. Select the table you want to delete and choose the Delete button.

---

---

**Warning** When you use the Data Manager to delete a table, all the data stored in that table is also deleted. If you want to preserve the data in the table you are going to delete, write a Visual Basic application to copy the data to a new table before deleting the existing table.

---

---

## Modifying Table Data

Once a database has been created, you use the Data Manager to add records to a table, delete records from a table, or modify the data in a table. To perform any of these tasks, you must first open a table.

### To open a table:

1. Open an existing database. The Tables window is displayed.
2. Select the table you want to modify and choose the Open button. The Table Access window is displayed. This window displays each of the table's fields and any data it contains.

You use the Table Access window to perform the following tasks:

- Add a record to a table.
- Delete a record from a table.
- Modify data in a record.
- View the current records in a table.

### To add a record to a table

1. Choose the Add button.
2. Enter data as desired in each field.
3. To save the new record choose the Update button, or move to another record. To return the table to its previous state (without saving changes), choose the Refresh button.

### To delete a record from a table

1. Select the record you want to delete.

There are two ways to locate a record in the Table Access window:

- Use the data control at the bottom of the Table Access window to view the first, previous, next, or last record in the table.
  - Use the Find button to locate a record in a large database.  
The syntax for the search expression is identical to the **Find** method as described in Visual Basic help. Generally, these follow the syntax for Structured Query Language (SQL) WHERE clauses. For example, in the Authors table in BIBLIO.MDB, you locate the tenth record by entering "AU\_ID > 9" as the search expression.
2. Choose the Delete button and respond to the prompt. If you answer 'Yes,' the record is immediately deleted and cannot be recovered.

### To modify the data in a record

1. Select the record you want to modify (see step 1 above).
2. Enter data as desired.
3. To save the new record choose the Update button, or move to another record. To return the table to its previous state (without saving changes), choose the Refresh button.

### To view the current records in the table

1. Choose the Refresh button.

Use the Refresh button to retrieve the current records from the table.

If the database is being used by more than one user, use the Refresh button to ensure you are working on the most recent data.

---

**Note** If you add or change a record, and the values in the record's fields do not meet the requirements set by the data type, or violate one of the restrictions established by the index, an error occurs. The maximum length for a record is 2000 characters.

---



### **Rules for Naming Database Objects**

In Visual Basic, database object names can be up to 64 characters long and can include any combination of letters, numbers, spaces (the first character cannot be a space), and special characters except a period (.), an exclamation mark (!), brackets ([ ]), or control characters (ASCII values 0 – 32).

## Repairing a Database

You may need to repair the database if Visual Basic quits unexpectedly: for example, if there is a power outage or your computer experiences a hardware problem. If your Visual Basic database is damaged, you can use the Data Manager to attempt to repair it.

### To repair a database

1. From the File menu, choose Repair Database.

The Repair Database dialog is displayed. Use the dialog to specify the name of the database you want to repair.

2. Choose OK.

---

---

**Warning** Not all database damage can be repaired. Remember to back up your database files regularly to avoid data loss that cannot be recovered with the Repair Database operation.

---

---

The Repair Database operation checks all pages in the database for correct linkage, validates all system tables, and validates all indexes.

All pages that cannot be salvaged – such as invalid references to other pages

– are discarded.

When a database is repaired, it may increase in size. This is because the process of creating indexes may leave some deleted pages in the database.

It is always a good idea to compact the database after any repair to eliminate unneeded pages.

**Data Page**

A portion of the database file in which record data is stored. Depending on the size of the records, a data page may contain more than one record. A data page is 2K.

## Other Information Sources

All help and text files included with the Standard Edition are also included with the Professional Edition.

### Standard Edition

[Help Files](#)

[Text Files](#)

### Professional Edition

[Help Files](#)

[Text Files](#)

---

**Note** When Visual Basic is installed, as each Help file is installed, it is listed in WINHELP.INI, located in your Windows directory. If you move a Help file to a different directory, be sure to change the path in WINHELP.INI. If a Help file does not exist, is not in your path, or is not in the directory specified in WINHELP.INI, WinHelp displays an appropriate message.

---

## Standard Edition Help Files

You can click on any help file to go to the main table of contents of that file. If the file is not available, an error occurs.

Visual Basic Help—Documents Visual Basic for Windows.

SetupWizard—Documents the SetupWizard application. For information about the Setup Toolkit, search for Setup in the Visual Basic help file.

Data Manager—Documents the Data Manager application.

---

**Note** When Visual Basic is installed, as each Help file is installed, it is listed in WINHELP.INI, located in your Windows directory. If you move a Help file to a different directory, be sure to change the path in WINHELP.INI. If a Help file does not exist, is not in your path, or is not in the directory specified in WINHELP.INI, WinHelp displays an appropriate message.

---

## Professional Edition Help Files

In addition to the help files provided with the Standard Edition, these help files are included with the Professional Edition. You can click on any help file to go to the main table of contents of that file. If the file is not available, an error occurs.

Crystal Reports—Documents the Crystal Reports application.

Custom Control Reference—Documents each of the custom controls provided with the Professional Edition.

Help Compiler Reference—Documents the Windows Help application for Help writers and programmers.

Hotspot Editor—Documents the segmented hypergraphic editor for creating hotspots within graphics for use in authoring Help files.

KnowledgeBase—A collection of articles from Microsoft Technical Support with tips, ideas and solutions.

ODBC Installation Help—Documents the installation tools for ODBC.

Oracle ODBC Driver—Documents the ODBC driver for Oracle databases.

SQL Server ODBC Driver—Documents the ODBC driver for SQL Server databases.

Visual Basic API Reference—Documents the Custom Control Development Kit.

Windows 3.1 API for Visual Basic—Declarations, structures, and constants for the Windows API as used in Visual Basic.

Windows 3.1 SDK Help—Documents Windows functions as used in the C programming language.

---

**Note** When Visual Basic is installed, as each Help file is installed, it is listed in WINHELP.INI, located in your Windows directory. If you move a Help file to a different directory, be sure to change the path in WINHELP.INI. If a Help file does not exist, is not in your path, or is not in the directory specified in WINHELP.INI, WinHelp displays an appropriate message.

---

## **Standard Edition Text Files**

You can click on any of these files to launch Notepad and load the file.

README.TXT—Information on last minute changes to Visual Basic, as well as additional information.

CONSTANT.TXT—Global symbolic constants for Visual Basic properties, events, functions and statements.

DATACONS.TXT—Global symbolic constants for the data access features of Visual Basic.

EXTERNAL.TXT—Additional README information about connecting to external databases.

PACKING.LST—List of all files on the distribution disks provided with Visual Basic.

## Professional Edition Text Files

In addition to the text files provided with the Standard Edition, these text files are included with the Professional Edition. You can click on any file to launch Notepad and load that file. If the file is too large for Notepad, you may have to use a different word processor.

BTRIEVE.TXT – Supplementary information on importing, exporting, or attaching Btrieve tables with Visual Basic.

ORACLE.TXT – Setup information for the ODBC Oracle driver to run with your ORACLE RDBMS software. If you installed ODBC, this file is in your Windows\SYSTEM directory.

PERFORM.TXT – Performance tuning tips for Visual Basic version 3.0 and Microsoft Access (TM) Relational Database System for Windows version 1.1.

SAMPLES.TXT – List of applications written in Visual Basic that demonstrate techniques discussed in the printed documentation.

WIN30API.TXT – Global symbolic constants for Windows 3.0 API functions.

WIN31API.TXT – Global symbolic constants for Windows 3.1 API functions.

WINMMSYS.TXT – Type declarations and global symbolic constants for Windows 3.1 multimedia API functions.



