dBASE IV®

Version 2.0

Getting Started

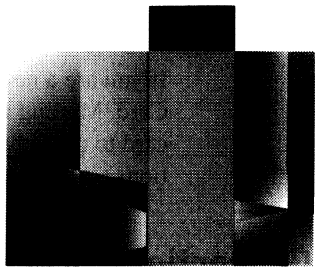
BORLAND INTERNATIONAL, INC. 1800 GREEN HILLS ROAD
P.O. BOX 660001, SCOTTS VALLEY, CA 95067-0001

PRINTED IN IRELAND

10 9 8 7 6 5 4 3 2 1

Contents

Introduction



Getting Started

Getting Started with dBASE IV will get you up and running with dBASE IV®. It consists of three major sections:

- Installation and Configuration
- What's New
- Exploring dBASE IV

You do not need to read every chapter of each section. For example, if you do not plan to run dBASE IV in a network environment, you will not need to read Chapters 3 or 4. Likewise, if this is your first time using the dBASE® product, the entire What's New section will not be relevant.

Installation and Configuration

This section tells you how to install dBASE IV onto your hard disk, how to set up the dBASE IV environment to suit your particular needs, and how to install and use dBASE IV in a multi-user, network environment.

What's New

This section describes new features and improvements in this version of dBASE IV, and summarizes all changes to the dBASE product since dBASE III PLUS®.

Exploring dBASE IV

This section teaches you the basics of dBASE IV. It demonstrates the ease and convenience of using the Control Center to access the features of the menu system, and the speed and flexibility of entering commands directly at the dot prompt.

Send Your Warranty Registration Card

If you are a new user of dBASE IV, your package includes a Warranty Registration Card. Mailing this card makes you a licensed user of the program. Be sure to mail this card to us within 30 days of the date you purchased dBASE IV, following the instructions on the card.

Conventions Used in dBASE IV Manuals

The following conventions apply throughout the dBASE IV documentation:

- Function keys (in the dedicated keypad on your keyboard) are shown either by themselves (**F1**), or with their assigned functions (**F1 Help**).
- Commands that you type appear in a different typeface, for example, DBASE.
- The RETURN or ENTER key is represented by the ↵ symbol, and generally follows any command you type: DBASE ↵.
- Menu choices, program messages, and screen references are in **boldface**.
- New terms and titles of books or manuals are shown in *italics*.
- Program listings appear in a different typeface.
- Notes, tips, and warnings contain additional information, provide helpful hints, or help you avoid problems.

About the Manuals

This section summarizes the topics covered in the rest of the dBASE IV documentation:

Using dBASE IV

Includes instructions on how to use the dBASE IV menu system to create database files, forms, reports, and labels, as well as write programs, manage files, and customize dBASE IV. This book also contains a section on using the Applications Generator, which tells you how to use this tool to create easy-to-use, menu-driven database management programs without actually programming in the dBASE language.

Programming in dBASE IV

Contains instructions on how to write dBASE IV programs and applications, how to use the Template Language to generate dBASE code, how to use Structured Query Language (SQL) in dBASE IV to enhance your database programming capabilities, and how to optimize dBASE's performance.

Language Reference

A complete, alphabetical reference to all dBASE IV and SQL commands, functions, system memory variables, and error messages.

Quick Reference

A fast guide to all dBASE IV terms, symbols, commands, functions, system memory variables, SQL commands and functions, and cursor movement keys.

Read.me files

There are two text files that are included on Disk 1. These are:

Filelist.doc	lists all the files contained in each compressed disk file on your original program disks.
Read.me	describes changes that occurred too late to be included in the printed documentation. Every user should read this file. This file is also copied to your dBASE home directory.

Installation and Configuration

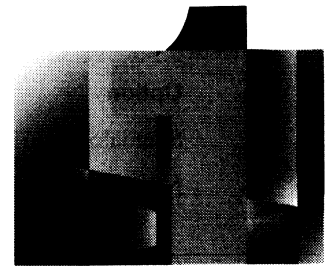
Installing dBASE IV

Configuring dBASE IV

Setting Up dBASE IV on a Network

Multi-User dBASE IV

Installing dBASE IV



This chapter contains instructions for installing and starting dBASE IV. It also contains information about running dBASE IV in different operating systems and environments.

Hardware and Software Requirements

To use dBASE IV 2.0, you need

- An IBM or IBM-compatible, protected-mode 286, 386, or 486 computer
- A monochrome, CGA, EGA, or VGA monitor
- 2MB RAM
- 7.5MB hard disk space to install all dBASE options
- 4.5MB hard disk space to install dBASE IV system files only
- An additional 2MB hard disk space during installation
- DOS 3.3x, 4.01, 5.0, or 6.0

For a list of supported networks, see Chapter 3, "Setting Up dBASE IV on a Network."

Types of Installation

The installation utility provides three installation options: Standalone, Network, and Optional. The following table describes each installation option.

Table 1-1 Installation options

Option	Description
Standalone	Installs all files; single-user access only.
Network	Installs all files; supports multi-user access to dBASE IV if you have the dBASE IV LAN Access Pack version 2.0.
Optional	Lets you select and install the following optional files: sample files, tutorial files, template language, and utilities. Use this installation if you have already installed dBASE IV 2.0 and just want to add the optional files.

dBASE's installation procedure is the same for the standalone and network installation options. However, if you are installing dBASE for multi-user capability, you also need to set up your network environment so that the dBASE files can be shared by multiple users. For information about configuring your network environment to run multi-user dBASE, read Chapter 3, "Setting Up dBASE IV on a Network."

Running the dBASE Installation Utility

Before you install dBASE, make a complete copy of the program disks using the DOS DISKCOPY command. Use the copy as your working copy and store the original program disks in a secure place.

To install dBASE,

1. Insert Disk 1 into drive A or B of your computer.
2. At the DOS prompt, type A: or B: and press ↵.
3. Type INSTALL and press ↵.
4. Follow the onscreen instructions.

Each screen provides messages and key prompts to help you make the appropriate choices.

Things You'll Be Asked During Installation

The dBASE Installation utility prompts you for the following information during installation:

- The source drive where the dBASE program disk is. Either use the default, A (drive A), or enter B (drive B).
- The type of installation you require: Standalone, Network, or Optional. For more information about the different types of installation, read the Types of Installation section in this chapter.

- Signature information: your name, company name, product serial number, and code page value. The installation utility asks you for this information only if you choose the Standalone or Network installation.

  Your product serial number is located on the first disk. You must provide this information in order to install and run dBASE.

  The code page value determines the character set that dBASE will use so that you can type, display, and print characters for a particular language. You should select the code page value that matches your operating system's code page value. dBASE IV version 2.0 supports code pages 437 (the default) and 850.

  Note: The code page value you select during installation of dBASE does not change your operating system's code page. You must load the DOS code page through DOS. For more information on how to load code pages, read the *DOS User's Guide and Reference.*

- The files you want to install and where you want them installed. The installation utility lets you select the files you want to install: all dBASE IV system files, sample files, tutorial files, template language, or utilities. These dBASE files are installed in designated directories unless you specify different ones through the Directories option.

  The installation utility also lets you choose whether or not to protect existing dBASE IV and SQL system files in the target directory. If you change the default from Yes to No, the installation utility overwrites all existing system files in the target directory automatically. If you use the default, the installation utility warns you about duplicate dBASE files and doesn't copy over those files into the target directory until you change the protection option to No.

Migrating From Previous Versions of dBASE

The installation utility doesn't modify or replace an existing Config.db. If you have an existing Config.db in the dBASE home directory, dBASE IV 2.0 uses that configuration file when it starts. Otherwise, the installation utility creates a new Config.db file.

If you want to use the new dBASE settings such as LDCHECK, MBLOCK, or IBLOCK in your configuration file, either modify your existing Config.db or create a new one. You can do so through DBSETUP or by using a text editor to edit the file. For more information, read Chapter 2, "Configuring dBASE IV."

If You Have Microsoft Windows

If you want to run dBASE IV under Windows, be sure that Window's Program Manager is configured to run dBASE. You should also set the amount of memory that dBASE uses when running under Windows. The following sections address these topics.

Configuring Window's Program Manager Initialization File

The dBASE IV installation utility checks for a directory in your PATH statement that contains the word WINDOWS (including such directories as AWINDOWS, BWINDOWS, and so on). If it finds one, it modifies the Program Manager initialization file, Progman.ini, to include the necessary files to run dBASE IV from Windows.

If your copy of Windows is in another directory (WIN386, for example), or if you install Windows after installing dBASE IV, you should modify this file yourself. Add the following line to the end of Progman.ini:

Group<n>=<dBASE directory>\DBASEIV.GRP

<n> is the next available group number. <dBASE directory> is the name of your dBASE IV directory, which is C:\DBASE by default.

Progman.ini is an ASCII file. You can use any program editor or word processor to edit it. Be sure to save it as an unformatted text (ASCII) file.

Allocating Memory to dBASE IV

Under Windows 3.1 enhanced mode, dBASE uses as much available virtual memory as possible, up to 12MB. To limit this amount (so that memory is reserved for other applications), use the DOS environmental variable, DOS16M. You set this environmental variable at the DOS prompt or in your Autoexec.bat file using the following syntax:

```
SET DOS16M =:<max memory>
```

<max memory> must be 3MB or higher. The following are examples of valid settings:

```
SET DOS16M =:3M

SET DOS16M =:3072
```

If your system runs out of memory while loading multiple copies of dBASE under Windows, you can increase Window's virtual memory size by increasing the size of the swap file. For information on how to do so, read the *Windows User's Guide*.

If You Have OS/2

To run dBASE IV 2.0 under OS/2, you must set the session's DPMI_MEMORY_LIMIT to a value equal to or greater than 3. If you don't, dBASE terminates while loading because of insufficient memory (a "Resource File Error" message appears). Use the following procedure to change a session's DPMI_MEMORY_LIMIT:

1. Select the dBASE, DOS Full Screen, or DOS Window icon.
2. Press the right mouse button. This displays a pop-up menu.
3. Click the arrow on the right side of the **Open** menu bar.
4. Choose **Settings**, then **Session**, followed by **DOS Settings**.

5. Choose **DPMI_MEMORY_LIMIT** from the list.
6. Increase the default value of 2 to 3 or higher.
7. Choose **Save**.
8. Close the **Settings** window.

You might also want to increase the **DOS_FILES** setting from its default of 20 if you want dBASE to open more than 20 files. You change this setting the same way you change **DPMI_MEMORY_LIMIT**, except you choose **DOS_FILES** from the **DOS Settings** list.

Starting dBASE IV

After you install dBASE IV, add or modify the following lines in your CONFIG.SYS file:

FILES = 99

BUFFERS=15

Also, to run dBASE without typing the full path at the DOS prompt, add C:\DBASE (or the directory where dBASE IV resides) to the path in your Autoexec.bat.

You can start dBASE IV the following ways:

1. If you're using dBASE for the first time, you might find it helpful to load dBASE's Control Center with a set of sample files. To do so, type DBSAMPLE at the DOS prompt and press ↵.
2. At the DOS prompt, type DBASE and press ↵. If you didn't add C:\DBASE to the path in your Autoexec.bat file, type the drive and directory where dBASE resides (for example, C:\DBASE\DBASE).
3. If you want to use a custom configuration file other than Config.db, enter the /C parameter after DBASE on the command line, followed by the path and file name for that configuration file (for example, DBASE /C C:\CONFIG.NEW).

   You can have multiple configuration files for different users or tasks, and select among them by using the /C parameter. If you type an invalid configuration filename, dBASE is loaded without a configuration file.
4. If you have written a dBASE IV program, you can run it automatically at start-up by typing the program name after DBASE. For example, if your program name is Employee.prg, type DBASE Employee at the DOS prompt. If your program accepts parameters, type them after the program name. A program name and its related parameters must be the last parameters on the command line. For example:

   DBASE /C C:\SMITH\CONFIG.NEW MYPROG DATE() 'SMITH' 24

   The program name (MYPROG) is followed by three parameters for that program (DATE(), 'SMITH', and 24). They are delimited with a blank and are the last parameters on the command line.

Quitting dBASE IV

Always exit properly from dBASE IV. If you turn off or reboot your computer without first exiting dBASE, any files you have open might not be saved. There are two ways to exit from dBASE IV:

- From the Control Center, press Alt-E to display the Exit menu, then select **Quit to DOS**.
- From the dot prompt, type QUIT and press ↵.

Both methods close all open files and return you to the DOS prompt.

Using the dBASE IV Tutorial

The dBASE IV tutorial teaches you the basic operations of dBASE IV. It shows you how to use the Control Center and catalogs, create and use database files, print reports and labels, create data entry forms, and use the dot prompt.

NOTE
Unload any memory-resident programs before you run the dBASE IV tutorial.

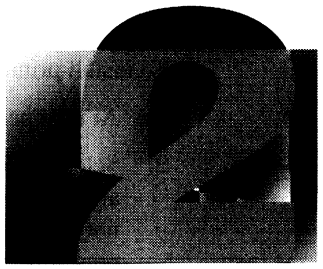
To start the tutorial,

1. From the dBASE IV directory, type DBTUTOR and press ↵.

   If you installed the tutorial in another directory, change to that directory and type INTRO ↵.

2. Select a topic from the list that appears on the screen.

Configuring dBASE IV



This chapter tells you how to set default options for dBASE IV and how to configure PostScript printer defaults.

When you start dBASE IV, it checks for the file Config.db in the dBASE IV home directory, or a file referenced in the command line following the /C parameter. This configuration file sets the dBASE IV environment (type of screen display, color, SET commands, and so on). For references to Config.db in this chapter, substitute the name of the file you intend to use as your dBASE IV configuration file.

Setting Default Options in the Configuration File Using DBSETUP

The installation utility checks for a configuration file called Config.db that contains default settings for dBASE IV. If the file is not already present in the dBASE IV directory, the installation utility creates one. If the file already exists, the installation utility leaves the file unaltered.

You can use the DBSETUP program to create or modify any dBASE IV configuration file, or edit it directly (it is a simple text file). To edit a configuration file directly, see the Configuration File Syntax section later in this chapter.

Installing Printer Drivers

Unless the configuration file tells it otherwise, dBASE IV will automatically use the default printer driver, Generic.pr2. You will most likely want to install at least one printer driver for your system. This is done most easily using the DBSETUP utility.

To install printer drivers or fonts by directly editing your configuration file, see the Configuration File Syntax section later in this chapter.

NOTE
Users with PostScript printers should refer to the PostScript Printers section later in this chapter.

Installing Printers Using DBSETUP

To install printer drivers using DBSETUP, follow these steps:

1. At the operating system prompt, type DBSETUP ↵.
2. Select either **Modify existing CONFIG.DB** or **Create new CONFIG.DB**. If you are modifying an existing configuration file (whether it is called Config.db or not), type the full path and filename (including extension) if they are not already present in the filename box.
3. Select the **Drivers** option from the **Printer** menu. The printer driver selection table appears. Press **Shift-F1** with the cursor in the **Printer name** column. A list of printer manufacturer names appears.
4. Highlight the manufacturer of the printer you want to install and press ↵. A list of printer models appears. Highlight the model you want to install and press ↵. The cursor moves to the **Device** column.
5. Press **Shift-F1** to display the list of devices. Highlight the device you want assigned to the printer and press ↵. (LPT1 and COM1 are the most common devices for parallel and serial printers, respectively.)
6. Repeat steps 3 through 5 to install additional printers. The name in the **Printer name** column can be changed to any descriptive name. Only the first 15 characters will display in the **Print/Destination** menu in dBASE IV.
7. When you have selected all the printers you want to install and assigned a device to each, press **Ctrl-End**.

Replacing the Printer Drivers File

Printer drivers are contained in the Drivers.exe file in your dBASE IV directory, and are automatically extracted by DBSETUP as needed. If you need to extract a driver manually, type DRIVERS -H ↵ at the operating system prompt for help.

If you deleted Drivers.exe from your dBASE IV directory, you'll need to restore it from the original disk. To restore a file from the original product disks, you need to know which disk contains that file. Filelist.doc is a text file in Disk 1 that shows the files found on each disk. Look at Filelist.doc using a text editor and locate the disk number containing Drivers.exe.

Copy Drivers.exe to your dBASE home directory, then run DBSETUP:

1. If prompted for the destination of the printer driver files, enter your dBASE IV directory name (usually C:\DBASE, the default) and press ↵.
2. Move the highlight to the **Default Printer** option of the **Printer** menu and press ↵. A list of installed printer drivers appears. Highlight the driver that you want as the default and press ↵.

3. Change any other options for this configuration file, then select **Save and exit** from the **Exit** menu. Press ↵ to accept the the configuration file drive, directory, and filename, or type a new path or filename, then press ↵ to save the file. If the configuration file already exists, you can select **OK** to overwrite it.
4. To exit DBSETUP, choose **Exit to DOS** from the **Exit** menu.

Installing Printer Fonts Using DBSETUP

You can configure up to five special fonts for each printer. Fonts will be available from the **Words** menu when printing.

Follow steps 1 and 2 from Installing Printers Using DBSETUP. Then select **Fonts** from the **Printer** menu, type a font name (for example, Superscript), and press ↵. Type the starting and ending codes for that font, which you will find in your printer manual, followed in each case by ↵. For example, the "superscript" starting code for an Epson FX-85 printer is {esc}S{null}, and the ending code is {esc}T.

For the dBASE IV equivalent of special keys like **Esc**, see the ??? command in Chapter 2 of *Language Reference*. Press **Ctrl-End** when finished setting fonts. To set up fonts for an Apple LaserWriter or 100% compatibles, see the PostScript Printers section later in this chapter.

Speeding Up Multi-User dBASE IV

If you are using dBASE IV in a network environment, you can speed it up by copying certain important files from the server to your own workstation. The menu option **Copy dBASE to workstation** on the **Reconfigure** menu does this automatically.

Select this option, then answer the questions as they appear. This will also create a Config.db file on your local drive. If you edit this file, do not delete the CONTROLPATH= setting, which links you to the server dBASE IV.

Gateway to the Operating System

The **DOS** menu option (at the same menu level as the **Config.db** menu) allows you to perform an operating system command, open an operating system shell, or set the default (start-up) drive and directory for dBASE IV.

Other Config.db Menu Options

When you open the **Config.db** menu and select either **Modify existing CONFIG.DB** or **Create new CONFIG.DB**, you see several menus that can set various conditions as default mode.

The **Printer** menu (described earlier in this chapter) is used to install printer drivers, select the default driver, and set up special printer fonts. Most of the items on the **Database**, **General**, and **Files** menus are the equivalent of SET commands, discussed in Chapter 3 of *Language Reference*. The **Display**, **Keys**, and **Memory** menus set default values for your monitor, function key actions at the dot prompt, and memory-related variables. Finally, the **Exit** menu sends you back to the top level of DBSETUP.

Setting Colors and Line Modes

The **Display** menu sets screen colors and line mode for dBASE IV. To set screen colors by editing the configuration file, see the Configuration File Syntax section later in this chapter. The default colors for a color display are listed in Table 2-1.

Table 2-1 Default screen colors for a color monitor

Menu item	Color	Color code
NORMAL TEXT:	Grey on Cyan	N+/BG
MESSAGES:	White on Blue	W/B
TITLES:	Bright White on Cyan	W+/BG
HIGHLIGHT:	Yellow on White	RG+/W
BOXES:	Bright White on Blue	W+/B
INFORMATION:	Light Cyan on Blue	BG+/B
FIELDS:	Bright White on Black	W+/N

The menu choice **Perimeter of screen** will only be accessible if your monitor allows the screen perimeter (the area that borders the text area of the screen) to be set.

Display mode allows you to select between available display modes for your monitor.

Configuring Hot Keys

The **Keys** menu assigns hot keys for use at the dot prompt. Press **Spacebar** to edit a key assignment, and enter the text string you want to type at the dot prompt. To simulate the Enter keystroke, terminate the string with a semicolon, for example, assist;.

Setting Default SET Commands

Defaults can be specified for the SET commands listed in Table 2-2 using DBSETUP, or by editing the Config.db file directly. The SET commands themselves are fully described in Chapter 3 of *Language Reference*.

All of the DBSETUP menu choices include a one-line description or explanation, which appears at the bottom of the screen when the choice is highlighted.

Table 2-2 SET commands in the DBSETUP menus

Database Menu:		**Files Menu:**		**Memory Menu:**
Autosave	Exact	Alternate	Procedure	Refresh
Blocksize	Exclusive	Catalog	Safety	Reprocess
Iblock	Fullpath	Default	View	
Mblock	Lock	Directory		
Carry	Near	Library		
Deleted	Unique	Path		
Encryption				
General Menu:				
Bell	Design	Ldcheck	Separator	
Century	Development	Margin	Space	
Clock	Device	Mark	SQL	
Confirm	Echo	Memowidth	Status	
Console	Escape	Mouse	Step	
Currency	Headings	Odometer	Talk	
Cursor	Help	Pause	Trap	
Date	History	Point	Typeahead	
Debug	Hours	Precision		
Decimals	Instruct	Printer		
Delimiters	Intensity	Scoreboard		

Other DBSETUP and Config.db Defaults

There are some defaults you can set through DBSETUP, or by editing Config.db, that do not have equivalent SET commands.

Database Menu

- **Expsize** and **Indexbytes**. Specify the size of the memory buffer (in bytes) for parsed expressions or for index nodes. Setting **Indexbytes** to a large value makes indexing faster but uses memory; setting **Expsize** high allows more complex expressions but uses memory. The default values for **Indexbytes** and **Expsize** are 256 and 100, respectively.

General Menu

- **Asciisort**. This switch is only active when **LangTables** is set on. **Asciisort**=ON nullifies one element of the **LangTables** setting, the collation table, causing sorting to follow ASCII order, or ASCII-based dictionary order (if you select a dictionary order sort in a query). If **Asciisort**=OFF, the default, sorting order is determined by the **LangTables** setting. See **LangTables**, in this section.
- **Command**. Specifies a command to run at dBASE IV start-up. Default is ASSIST.
- **Fastcrt**. Allows fast screen updating when using a CGA card. Default is ON.
- **LangTables**. Changes the setting for the language table. **LangTables**=ON affects three areas, one of which can be modified by **Asciisort**. First, it causes dBASE IV to use the language table to determine whether a character (such as ç or ü) is an alphabetic character. Second, it causes the language table to be used when mapping a lowercase character to uppercase (for the UPPER( ) function), and vice versa. Third, it causes the collation table to be used in sorting and indexing, so that, for example, ü comes before v even though it has a higher ASCII number. This third function can be turned off by setting **Asciisort**=ON while still leaving the first two functions operational. **LangTables** defaults to OFF if your DOS country code is set to United States; otherwise it defaults to ON.
- **Noclock**. If ON, suppresses the clock display in full menu commands. Default is OFF.
- **Prompt**. Specifies characters to display in place of the dot prompt. Default is a dot (.).
- **Resetcrt**. Restores screen to current default colors and line mode when you exit from **DOS Access**. Default is ON.
- **Tabs**. Specifies default tab stops (separated by commas). Default is no stops specified.
- **Tedit**. Specifies an external text editor for MODIFY COMMAND. Default is no external editor specified.
- **Wp**. Specifies an external text editor for memo fields. Default is no external editor specified.

Files Menu

- **Files**. Specifies the number of files allowed open at once (cannot exceed 99).

- **SQLdatabase, SQLhome**. Specifies default SQL database and SQL system file directory. Type the full drive and directory name.

A number of submenus in the **Files** menu allow you to specify dBASE IV programs or procedures to run just before you enter a called surface from the Control Center, or save and exit from a called surface. When you define these custom programs, certain dimmed menu options of the Control Center become active. These dimmed options are:

Catalog menu	Open custom utility
Layout menu	Invoke layout program
Field menu	Load field program

See Chapter 17 of *Programming in dBASE IV* for custom surface programs.

Memory Menu

It is possible that a large dBASE IV program may run out of memory. If this happens, there are some elements of memory usage you can configure through the **Memory** menu.

- **Bucket** and **Gets**. For use with the @...GET command; **Bucket** specifies kilobytes of memory for @...GET PICTURE and @...GET RANGE, while **Gets** specifies the number of @...GET statements that may be active at once. Default is 2 for **Bucket**, 128 for **Gets**.
- **Mvblksize, Mvmaxblks, Rtblksize**, and **Rtmaxblks**. For use with memory variables (memvars) and run-time symbols. **Mvblksize** sets the number of memvars per block (each memvar uses 56 bytes of dynamic memory); **Mvmaxblks** sets the maximum number of blocks. **Rtblksize** and **Rtmaxblks** work similarly for run-time symbols. Defaults are 50, 10, 50, and 10, respectively. See LIST/DISPLAY MEMORY in Chapter 2 of *Language Reference* for more information.
- **Ctmaxsyms**. Sets maximum number of compile-time symbols to compile a program or procedure file. Default is 500.

Configuration File Syntax

Most dBASE IV users find it much easier to use the powerful DBSETUP program to alter Config.db files. However, it is possible to edit Config.db directly. If you do, be sure to save it as an ASCII text (unformatted) file.

All Config.db commands have this basic syntax:

<item> = <values>

SET command equivalents are entered as follows:

<SET command> = <value>

Do not type the keyword SET. For example, REFRESH=10.

Color Settings

Color settings are entered as follows:

COLOR OF <screen part> = <foreground>/<background>

See Setting Colors and Line Modes earlier in this chapter for the default screen color codes. The values that you can enter for <screen part> correspond to the **Display** menu items in DBSETUP with the exceptions of **Normal text** and **Boxes**. To set the color of normal text and boxes in Config.db, you use COLOR OF NORMAL = and COLOR OF BOX =, respectively.

Assigning Function Keys

There are two acceptable commands to assign function key values:

<key label> = "<text>"

and

FUNCTION = <function key number>, "<text>".

In the first, <key label> can be **F2** through **F10**, **Ctrl-F1** through **Ctrl-F10**, or **Shift-F1** through **Shift-F9**; "<text>" is any text string you can type at the dot prompt, in double quotation marks. To force a carriage return (after a command, for instance), terminate "<text>" with a semicolon (inside the quotes): F2="ASSIST;".

Configuring a Printer

To configure a printer, use the following syntax. This entire command must be entered on one line:

PRINTER <printer number> = <driver name> NAME "<menu name>"
DEVICE <device name>

The <printer number> can be 1 through 4. You must know the exact <driver name>, for example, Hplas260.pr2. The "<menu name>" can be anything descriptive, in quotes; only the first 15 characters will display on the **Print** menu. The <device name> can be PRN, LPT1–LPT3, COM1–COM4, or NUL. (DEVICE NUL simulates write operations for test purposes, but does not actually print.)

If you set up a printer by editing Config.db directly, you must also extract the driver from the Drivers.exe file. Type DRIVERS -H and press ↵ in the DBASE directory for instructions. If you use DBSETUP to configure a printer, DBSETUP will extract the printer driver for you, so long as it can find Drivers.exe.

Setting Default Printer

To designate one of the printers as the default, use the PDRIVER command:

PDRIVER = <driver name>

You must use one of the drivers you set up in a PRINTER statement.

Setting Printer Fonts

You can set up to five special fonts for each printer. To set up a special font, use the following syntax. This entire command must be entered on one line:

PRINTER <printer number> FONT <font number> = <start>, <end>
NAME "<menu name>"

The <printer number> can be 1 to 4; <font number> can be 1 to 5; <start> and <end> are the starting and ending escape codes for the font (see Printer Control Codes later in this chapter); "< menu name>" is a descriptive name in quotes to appear on the **Words/ Style** menu when the corresponding printer is selected in quotes (only the first 26 characters will appear in the menu). See the Installing Printer Drivers section earlier in this chapter for more information.

Specifying Programs to Run from Called Surfaces

The Config.db lines specifying the programs to run from called surfaces resemble the following:

PRG<surface> = <entry>,<exit>,<layout>,<field>,<execute>

Program names are not surrounded by quotation marks. You can include a path. You do not need to specify all possible programs for each surface; if you skip a definition, no program will be run when that event occurs, but you must add delimiting commas (PRGFORM = <entry>,,,<field>). You do not need to add trailing commas.
See *Programming in dBASE IV* for a complete description of programs that run from called surfaces.

Config.db Statement Examples

The following are examples of the above commands. Spaces before the = sign are optional:

AUTOSAVE = ON

SHIFT-F8 = "COPY STRUCTURE TO"

```
FUNCTION     = 2, "ASSIST;"
PRINTER 1    = HPLAS260.PR2 NAME "HPLJ2 60 lpp" DEVICE LPT1
PRINTER 2    = FX85_1.PR2 NAME "Epson FX-85" DEVICE LPT2
PDRIVER      = HPLAS260.PR2
PRGCC        = , ,MYASSIST
```

Printer Control Codes

Printer control codes are sequences of ASCII characters, often but not always beginning with the **Esc** character, that control printer options, such as changing to another font or selecting page size. The codes are different for different printers and are obtained from the printer documentation.

Some characters, such as **Esc**, can be written by enclosing the name in curly braces: {ESC}. Any character can be written by enclosing its decimal ASCII number in curly braces: {27} (**Esc** is ASCII character 27). For example, the escape code sequence for begin-superscript/characters in the Epson FX-85 printer is **Esc**-50, and the code to return to normal characters is **Esc**-T. The first of these can be entered as {ESC}50, {27}50, {27}{53}{48}, and so forth (characters "5" and "0" are ASCII characters 53 and 48, respectively).

See the ??? command in Chapter 2 of *Language Reference* for more information on escape code sequences.

PostScript Printers

PostScript is a *page-description language*, by which a program such as dBASE IV tells a PostScript printer how to print a document.

PostScript is independent of the physical printer; you only need one driver to print on any PostScript printer that is 100% compatible with the Apple LaserWriter, no matter who manufactures it.

That is the theory. In practice, you may need to make slight adjustments. This section tells you exactly how to do that, step by step.

Selecting a PostScript Printer in DBSETUP

Select Postscript from the **Drivers** menu in DBSETUP.

You can set up PostScript fonts, which will be available from the dBASE IV **Words** menu, when you print. There are three fonts: 1FONT, 2FONT, and 3FONT, which by default correspond to Courier, Helvetica, and Times-Roman, respectively.

To make these fonts available, configure as discussed in the Installing Printer Drivers section. The starting code is the font name, followed by the ASCII code for a space, {32}. For example, the starting code for Helvetica is 2FONT{32}. The ending code is the default font, usually 1FONT{32}.

Printer Command Macros

Using a PostScript printer, you can type a command macro from within the dBASE IV menu system or from the dot prompt to affect printing. See Chapter 10 of *Using dBASE IV* and the ??? command in Chapter 2 of *Language Reference* for complete instructions.

Changing PostScript Defaults

If you find yourself constantly typing the same printer macro over and over, you may wish to make it the default by editing the Postscri.dld file.

Postscri.dld is a regular text file that sets up a PostScript printer. You can edit it with any programming editor that does not put in a 1Ah character at the end of the file. Do not use the dBASE editor.

Postscri.dld is also a PostScript program. It comprises two sections, the user-adjustable parameters and the program itself. The first section, labeled **User-adjustable parameters** in the file, sets a number of parameters, such as the number of lines per page, the page size, and so on. This is the only section you should edit.

Page Size and Orientation

In PostScript, lines that begin with a slash (/) and end with *def* are definitions. The first such line in Postscri.dld is the following:

```
/paper 1 def          %(1=letter, 2=legal)
```

NOTE
PostScript ignores anything on a line following the % character. The comment to the right of the % is for your information only.

This line makes the default paper size {LETTER }. If you customarily print onto legal-size paper, you can make {LEGAL } the default paper size by changing the 1 to a 2:

```
/paper 2 def          %(1=letter, 2=legal)
```

The next line makes {PORTRAIT } the default page orientation:

```
/orient 1 def         %(1=portrait, 2=landscape)
```

To make the default mode landscape (printing across the length of the page), change the line to:

```
/orient 2 def      % (1=portrait, 2=landscape)
```

/nLPP stands for "number of lines per page." In portrait mode, set this to a default of 60 or 66 lines per page on letter-size paper, or 78 lines per page on legal. In landscape mode, set /nLPP to 45 lines per page for both letter and legal.

```
/nLPP 66 def          % lines per page (60, 66, or 78 for portrait; 45 for landscape)
```

NOTE
If you change the page length parameter, remember to set the **Length of page** *option on the* **Print** *menu to the corresponding number: 60, 66, 78, or 45.*

WARNING
With some PostScript printers, printing many lines per page will exceed the memory in the printer, causing some of the data to not be printed. If this happens, decrease the number of lines per page.

Print Offset

If the text is not centered correctly on your page, you will need to adjust the *x-offset* and *y-offset* parameters in Postscri.dld.

There are four pairs of offset numbers:

- tpxoff, tpyoff: Letter size, Portrait
- gpxoff, gpyoff: Legal size, Portrait
- tlxoff, tlyoff: Letter size, Landscape
- glxoff, glyoff: Legal size, Landscape

By default, the first two are set as follows:

```
/tpxoff 18 def        % x (letter)
/tpyoff 28 def        % y (letter)
```

The numbers themselves are in points; 72 points equals 1 inch.

To move the image to the right on the page, increase the x-offset number. To move the image to the left, decrease the x-offset number.

To move the image up on the page, increase the y-offset. To move the image down, decrease the y-offset. Both the x-offset and y-offset can be negative values. Precede the number with a single dash:

```
/tpxoff -6 def % x (letter)
```

The four modes of printing are independent of one another; if you change the x- and y-offsets of any one of them, it will not affect the others in any way.

For example, suppose the text prints too high and too far to the left. Measuring the page with a ruler, you decide to lower the text ½-inch and move it ¼-inch to the right. You should *decrease* the y-offset by 36 and *increase* the x-offset by 18.

Since you are printing in portrait mode on letter-size paper, the two offsets you want are *tpxoff* and *tpyoff*. Change the lines to read:

```
/tpxoff 36 def          % x (letter)
/tpyoff -8 def % y (letter)
```

Now print the page again to see if further adjustment is needed. When the printing is properly centered, repeat the procedure for landscape, and for both portrait and landscape in legal size.

Printer Fonts

PostScript can print any predefined font in any size. The two most common font sizes are 12-point and 10-point. It is a common convention to call these two standard sizes *Pica* and *Elite*, respectively.

The next two lines in the Postscri.dld file set these definitions:

```
/PicaPoint 12 def         % Pica
/ElitePoint 10 def        % Elite
```

To fit more words on the page, you often want a compressed font. The next line sets it to 8-point:

```
/CompressedPoint 8 def  % compressed
```

The next line declares one of these three point sizes to be the default:

```
/CurPoint PicaPoint def   % initial pointsize
```

You can change this to **ElitePoint** or **CompressedPoint**, and you can change any of the three point size definitions to any point size you want. The **Report** function, however, assumes they are 8, 10, and 12 points, and columns may not properly line up if you change sizes.

Font Names

The Postscri.dld file sets up the following three fonts as defaults:

- Courier (1FONT)
- Helvetica (2FONT)

- Times-Roman (3FONT)

The initial (start-up) font is selected by the CurFSet line:

```
/CurFSet 1 def          % default font set
```

This selects font 1 (Courier) upon start-up; you can change it to 2 or 3, if you want.

NOTE
Courier is the default font because it is monospaced, *and the other fonts are* proportionally spaced. *With a monospaced font, every character takes exactly the same width to print, as on a typewriter. With a proportionally spaced font, thin characters such as* i *take less space than fat characters such as* W. *The* ***Report*** *function expects a monospaced font when it aligns columns. If a proportionally spaced font is used, columns might not align.*

You can change the fonts themselves, if you prefer another font or if your printer does not have one of the three defaults. For example, you can specify the font Bookman in place of Helvetica. The Helvetica fonts are defined in the CurFSet 2 eq statement (which is seven lines long).

NOTE
This example assumes that Bookman is a resident font *on the printer, as with an Apple LaserWriter II. A resident font is a permanent part of the printer; a downloaded font is sent to the printer from the computer. You cannot change to a downloaded font in dBASE IV. The resident fonts for each LaserWriter compatible printer are listed when you run the LaserWriter Font Utility that comes with Apple LaserWriters.*

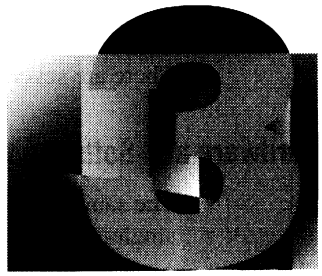
You must know the actual names used by the font for normal, bold, italic, and bold-italic styles (your font documentation will tell you this).

For Bookman, they are called Bookman-Light, Bookman-Demi, Bookman-LightItalic, and Bookman-DemiItalic. Thus, /Helvetica changes to /Bookman-Light, and so forth. Change the CurFSet 2 eq part of the /FSet definition to read:

```
CurFSet 2 eq
  {
   /n      /Bookman-Light def
   /b      /Bookman-Demi def
  /i       /Bookman-LightItalic def
  /bi      /Bookman-DemiItalic def
  } if
```

The remainder of the Postscri.dld file consists of the actual PostScript program to set up the printer according to the definitions. *Do not change this part of the file.*

Setting Up dBASE IV on a Network



The steps required to install dBASE IV onto a local area network (LAN) are structurally the same for each network. The specifics, however, differ from network to network.

The basic installation outline is as follows:

1. Log in to the server as network administrator or supervisor with full access rights to the parent directory into which you will install dBASE IV.
2. Install multi-user dBASE IV on the server.
3. "Share" directories (make them usable to other users) that contain dBASE IV files, data files, sample files, and SQL files as necessary, then link these files to the workstations.
4. Ensure that each workstation has the appropriate Config.sys and Config.db files. These are both simple ASCII (text) files you can create with any text editor. Config.db can also be created by the Dbsetup program.

NOTE
If the network users will be using the Applications Generator to design applications, you should copy the files Dbase2.res and Dbase3.res to each user's start-up directory (generally the user's home directory), so that each user can have unique defaults. If the user start-up scripts move them to the dBASE IV directory on the server to start up, you should create a separate, readable and writable directory for each user, then add that directory to the user's PATH statement (in Autoexec.bat) before *the shared dBASE IV directory (which should always be in the PATH itself).*

The rest of this chapter gives details about each of the above steps.

Before You Begin

There are several requirements to check before installing dBASE IV onto a LAN.

Hardware and Software Requirements

The same hardware and software requirements exist for multi-user dBASE IV as for single-user. They are listed in Chapter 2, "Installing dBASE IV."

If you use the SHARE command, DOS 4.01, or the FASTOPEN directory cache, insert the following line into your Config.sys file:

```
FCBS=16,16
```

Networks Supported

The following LANs are supported by dBASE IV:

- IBM PC LAN version 1.3
- IBM OS/2 LAN Server version 2.0
- Microsoft LAN Manager version 2.1
- Artisoft Lantastic version 4.1
- Novell NetWare versions 2.2 and 3.11
- 3Com 3+Share version 1.6.0
- 3Com3+Open version 2.0
- Banyan Vines version 5.0
- DEC Pathworks for DOS version 4.1
- AT&T Stargroup version 2.1a

Step 1: Install Multi-User dBASE IV on the Server

Log in as network administrator, then follow the installation instructions in Chapter 1 of this manual. Make sure you have all necessary rights to create, delete, and write files before beginning. This step is the same for each type of network supported.

NOTE
Do not attempt to install dBASE IV to the root directory (\) of your network drive, since access can be inherited to subdirectories, and the root directory can only hold a limited number of files. dBASE IV should be installed to a shared subdirectory.

Step 2: Share the dBASE IV Directory and Create Links to Workstations

Sharing a directory means making it accessible to other users. Once a directory is shared, you must create a path for each workstation, which tells it how to find the shared directory.

3Com 3+Share Network

First log in as the network administrator, then run the 3F SHARE command by typing:

```
3F SHARE ? ↵
```

The SHARE command displays a series of prompts (shown in boldface type below), to which you should respond as follows:

Sharename? <dBASE IV directory> ↵
Path? <path on server>↵
Password? [<password1>] ↵
Access (/RWC)? ↵

You can repeat this process to share a directory that contains shared dBASE IV files, sample files, or SQL files.

<password1> is an optional password; if you enter anything other than ↵, users will have to type the same password to access the dBASE IV directory. This is a network-level security process, and does not affect whether or not you set up a password system for dBASE IV itself using PROTECT.

The 3F program responds by displaying:

\\Server:Domain:Org\DBASE shared

To link workstations to the shared directories, type the following command into the Autoexec.bat file of each workstation for each shared directory:

```
3F LINK <drive letter>: \\<server name>\<shared directory> [/PASS=<password1>]
```

Finally, add the new drive letters to the PATH statement in each workstation's Autoexec.bat file.

Novell Network

To install dBASE IV on a Novell Local Area Network, you must have Supervisor rights (or equivalent rights to all required drives and directories). Otherwise, all the required files will not be copied to the server.

Log in to the network as the network administrator and run the Syscon utility to set up log-in scripts and trustee assignments for each Novell workstation. Each log-in script should resemble the following:

map search1:=<volume>:public
map search2:=<volume>:login
map search3:=<volume>:<dBASE IV directory>
map <drive letter>:=<volume>:<dBASE IV directory>
map <drive letter>:=<volume>:<database files directory>
map <drive letter>:=<volume>:<sample files directory>
#capture l=2 ti=1

Next, make your Novell trustee assignments. Enter the following assignments in Syscon for each workstation user:

<volume>: <dBASE IV directory>
read
open
search
create [parental*]

<volume>: <database files, sample files, SQL files directories>
read
write
open
search
create
delete

<volume>: <CONTROLPATH= directory> (often the dBASE IV directory)
read
write
open
search
create**
delete***

* use parental rather than create privilege if you are using Novell NetWare version 3.11
** create privilege only needed when entering dBASE IV the first time
*** delete privilege required for proper operation of PROTECT utility

Finally, type the following commands into the Autoexec.bat file of each workstation to allow users to log in to the network:

IPX
NETX (if using Novell Netware version 3.11 and DOS versions 3.x to 6.0*; if using Novell Netware version 2.2, enter NET3 for DOS version 3.x, NET4 for DOS version 4.01, or NET5 for DOS version 5.0.
<network drive letter>:
LOGIN <username>

* To use NETX with DOS version 6.0, you must first include the following line in each workstation's Config.sys: DEVICE=C:\DOS\SERVER.EXE

IBM PC LAN, IBM LAN Server, Lantastic, 3Com3+Open, and MS LAN Manager Networks

All these networks, except for IBM PC LAN, use DOS's Share program for file-sharing capability. For information on using the Share command, refer to your DOS manual.

For the IBM PC LAN, type the following command in the server's Autoexec.bat file:

NET SHARE DBASE=<dBASE IV directory path>: /RWE ↵

The parameter /RWE tells the network that the directory is open for reading, writing, and executing programs. Type a similar command for each shared directory (containing shared dBASE IV files, sample files, or SQL files) you plan to make available, using a different name instead of DBASE.

To link workstations to shared directories for all the networks, type the following into the Autoexec.bat file of each workstation. Repeat for each shared directory, using the appropriate names instead of DBASE:

NET USE <drive letter>: \\<server name>\DBASE ↵

Finally, add the new drive letters to the PATH statement in each workstation's Autoexec.bat file.

Banyan Vines

Use DOS's Share program to install file-sharing capability. Then, to link workstations to the shared directories, type the following statement in each workstation's Autoexec.bat file:

Z:SETDRIVER DRIVE:"<server name> <SharedFiles>@<Group name>@<Organization name>"

DEC Pathworks

Use DOS's Share program to install file-sharing capability. Then, to link workstations to the shared directories, type the following statement in each workstation's Autoexec.bat file:

USE DRIVE: \\<server>\<path>

Step 3: Configure Server and Each Workstation

Set up the Config.db file at each workstation as desired (the users can do this at their own workstations). See Chapter 2 of this manual for instructions. This step is the same for each type of network supported.

Next, check the Config.sys file of each workstation to ensure it contains the appropriate commands from the following sections.

All Supported Networks

Ensure that the following is contained in each workstation's Config.sys file:

```
FILES=99
BUFFERS=15
```

IBM PC LAN, IBM LAN Server, MS LAN Manager, Banyan Vines, AT&T Stargroup, and 3Com Networks

In addition to the above FILES and BUFFERS commands, each workstation's Config.sys file should also contain the following line:

```
LASTDRIVE=<drive letter>
```

Set <drive letter> to a valid drive letter between A and Z which corresponds to the highest-letter drive you plan to use (the default is E).

DEC Pathworks Network

In addition to the above FILES and BUFFERS commands, each workstation's Config.sys file should also contain the following lines:

```
DEVICE=C:\<path>Protman.sys /I: C:\<path>

DEVICE=C:\<path>ELNKMC.DOS
```

IBM PC LAN and IBM LAN Server Networks

To set device drivers, type the following statements in each workstation's Config.sys file:

```
DEVICE=<drive>:\dxma0mod.sys

DEVICE=<drive>:\dxmc0mod.sys

DEVICE=<drive>:\dxmt0mod.sys

DEVICE=<drive>:\dxme0mod.sys

DEVICE=<drive>:\smcmac.dos (use for Western Digital cards)
```

Running dBASE IV

Running dBASE IV on a network is almost identical to running it on a single-user PC. Some operations may be slower depending on the number of users, whether another user is exclusively accessing a file you need, and on the amount of memory available in the server and workstation.

There are also certain protections that can be implemented to ensure the security of each user's database files, and to make sure that two users do not attempt to alter the same record simultaneously; see Chapter 4, "Multi-User dBASE IV," for more information.

Starting dBASE IV

If the linked drive letter has been added to your path, you can use the same command you would use to start dBASE IV on a single-user system. If you want dBASE IV to use a Config.db file on your own workstation when it starts up, use the -c parameter (see Chapter 1 for further information).

As usual, the name of a program (application) file, such as Invoices.prg, can be added to run upon start-up; if this program is on the server, be sure to add the path name. Type any start-up parameters to this program directly after the program name:

```
DBASE -c C:\DBASE\Config.db Invoices "SMITH" {5/23/85} ↵
```

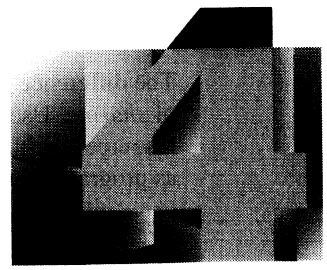
If the drive has not been added to your path, either precede the command with the appropriate drive letter, or change to that drive before running dBASE IV.

NOTE
If the system has been PROTECTed, a valid user login must be entered before the dBASE IV copyright screen appears.

Adding and Subtracting Users

The total number of users who can run dBASE IV at the same time is called the *user count*. This concept is fully explained in Chapter 4, "Multi-User dBASE IV," along with full instructions for adding and subtracting users from the user count, and setting up user accounts in the dBASE IV PROTECT utility.

Multi-User dBASE IV



This chapter describes how to use the multi-user aspects of dBASE IV. A multi-user environment allows several users to access dBASE IV from workstations connected to one or more servers.

This chapter is divided into three sections:

- For the User
- For the Developer
- For the System Administrator

Users of multi-user dBASE IV are divided into three groups. *Users* enter data, activate indexes, print reports or labels, and run application programs. *Developers* create database files, indexes, reports, and forms, and may even develop application programs. *System administrators* have the responsibility of administering and maintaining the multi-user environment, which includes setting up accounts, setting passwords, performing backups, and so forth. Note that one individual may belong to more than one group; that is up to the system administrator.

All users of a multi-user dBASE IV system should read the first section, developers should also read the second section, and system administrators should read all three sections. In general, information in an earlier section is not repeated in a later section, with some minor exceptions.

NOTE
It is assumed that you will be familiar with single-user dBASE IV before *you read this chapter, whether you are a system administrator, developer, or user. In particular, this chapter frequently refers to the complete procedural discussions in* Using dBASE IV *and the complete command descriptions in* Language Reference.

For the User

The difference between server environments and single-user systems is the principle of *sharing*. Users on terminals or workstations can access both shared hardware resources such as server CPUs, disk drives, and printers, and software resources such as programs and data.

Generally, the network allocates hardware resources, and the application (dBASE IV in this case) allocates software resources.

In addition, any workstation that has storage capability (such as a hard disk) can store local data: database files, indexes, report forms, applications, and so forth. Locally stored dBASE IV executable and overlay files can speed up multi-user operations; see Chapter 1, "Installing dBASE IV."

What It Looks Like

Physically, each user sits at a *workstation*, which may be a computer or a diskless terminal. Each workstation is hardwired to a *network*, which comprises a central computer, called a *server*, and some number of printers, disk drives, tape drives, and so forth.

dBASE IV is normally only stored on the server. When you want to run the program, your workstation will load the server's copy into local memory, just as if dBASE IV resided on the workstation's local hard drive. From the user's point of view, there is virtually no difference in how dBASE IV looks and responds in a multi-user environment, with the following exceptions: first, data may physically reside anywhere on the server and may have been moved or altered by another user; second, a file you need to use may have been copied by another user — thus, multiple versions of essentially the same file might exist; and third, another user may be using a file at the same time you are, making changes that affect your own edits.

Logging In to dBASE IV

If the system administrator has set up a security system using the PROTECT command, you will see a log-in screen when you type DBASE ↵, rather than the dot prompt or Control Center.

Type your group name, user name, and password, all three of which will be assigned to you by the system administrator. Your password will not appear on-screen as you type it.

If you type the wrong information three times, the system will terminate with an **Unauthorized log in** error message.

Multi-User Compared to Single-User dBASE IV

There are a few general differences between single-user and multi-user dBASE IV, as described in the following sections.

Finding Files

Before you can open a file, you must know where it is. This is a significant issue in multi-user dBASE IV, since you not only have to know the server drive and directory in which the file resides, you might also have to be aware of other copies of the file in order to ensure that you are editing the correct one (though it is generally good practice to allow only one copy of each file to exist).

Locked Files — What You See

With multi-user dBASE IV, it is possible that someone else may be using a particular database or other file or specific piece of hardware when you need to use it.

Generally, the network itself takes care of hardware conflicts. But with dBASE IV, the system administrator can establish rules for file access. Users *lock* (take temporary exclusive control of) a file or record when they execute certain commands; correspondingly, you may be denied access to a file or record because another user has locked it.

If you try to execute a locking command on a file or record already locked by another user, dBASE IV informs you that the file or record is in use, and attempts to lock it until you press the **Esc** key or take other actions specified by an error procedure in an application.

Using Transactions

A dBASE IV transaction is a sequence of actions, generally in a dBASE IV program, which is considered a single task. If one of the actions cannot be completed, for example if a database file cannot be opened, then you can ROLLBACK all previous actions in the transaction to their original state.

Because you are dealing with multiple users accessing the same files, it is more important to perform operations within a transaction than it is in single-user dBASE IV. The use of transaction processing in multi-user dBASE IV is discussed later in this section. For further transaction processing information, see the BEGIN/END TRANSACTION and ROLLBACK commands in Chapter 2 of *Language Reference*.

dBASE IV Data Security

Data security means protecting data from both accidental loss and unauthorized viewing or changing. There are three levels of data security when using multi-user dBASE IV:

- DOS security. DOS itself provides a certain level of security against accidental data loss with BACKUP (or an equivalent backup utility) and the ATTRIB command.
- Network security. With most networks, the network administrator (often the same person as the dBASE IV system administrator) can set up passwords, various levels of users, groups of users (such as ACCTNG, PAYROLL, SALES), and directories or entire disk drives that only certain types or groups of users can access. Networks can also ensure that two users do not use the same device simultaneously, or try to save the same file in the same location simultaneously.
- dBASE IV security. Multi-user dBASE IV provides a great deal of security using the PROTECT utility, which is fully described in Chapter 14 of *Using dBASE IV.* Briefly, PROTECT is used by the dBASE IV system administrator to set up a password system, independent of the network passwords, to allow particular file operations by specific users.

The rest of this section explains what dBASE IV tasks the user can perform, and how they differ from equivalent tasks in single-user dBASE IV.

Starting Multi-User dBASE IV

To start multi-user dBASE IV, enter the same command you would at the DOS prompt:

```
DBASE ↵
```

If you want to run a program immediately upon starting, append the program name:

```
DBASE <application name> ↵
```

NOTE
In most cases, the system administrator has set up each workstation so that you do not need to change to the server drive or directory to run dBASE IV; if you receive the error message **Bad command or file name**, *talk to your system administrator to determine the drive and directory where dBASE IV is stored.*

About the Current Session

Before you begin work, or at any subsequent time, you may want to determine which users are logged in and your own dBASE IV status — which database is currently selected, which files and records you have locked, your working drive and directory, and so forth. The commands LIST/DISPLAY USERS and LIST/DISPLAY STATUS perform these tasks.

LIST USERS and DISPLAY USERS

Two commands display the names of users currently logged in to dBASE IV: LIST USERS and DISPLAY USERS. The commands are identical, except that LIST USERS does not pause periodically as DISPLAY USERS does and is thus more convenient for printing. The two commands are generally shown as one, LIST/DISPLAY USERS.

LIST STATUS and DISPLAY STATUS

Two similar commands provide information about the current dBASE IV session on the network: LIST STATUS and DISPLAY STATUS. Type either command at the dot prompt; add the parameter TO PRINT to direct the output to the currently defined printer (defined by SET PRINTER) in addition to the screen. The two commands are generally shown as one, LIST/DISPLAY STATUS.

LIST/DISPLAY STATUS shows which database files have been locked; if the database file has been CONVERTed, LIST/DISPLAY STATUS also shows who locked the file last.

Here is the part of a sample LIST STATUS that shows locking information:

```
Currently Selected Database:
Select area: 1, Database in Use: M:\DBASE411\TEMP1.DBF Alias: TEMP1
   Lock list:       1,      3,      5,      9  locked
```

The database has records 1, 3, 5, and 9 locked. LIST/DISPLAY STATUS also displays other information, including the current values of various SET commands and key assignments.

Updating Files and Records

In a multi-user environment, a file or record you want to update may already be in use by another user. This does not necessarily prevent you from using that file or record as well; the commands each user can execute depend upon other commands being executed that pertain to the same database.

This section tells you how to open a file for shared use or for your exclusive use, which commands require the file to be opened exclusively, and how automatic locking works.

NOTE
REFRESH can be executed before using BROWSE or EDIT to ensure that changes made by other users will be updated on your screen at specified intervals. See the SET REFRESH command in Language Reference *for detailed information.*

Multi-User Files: Mode, Lock Status, and Protected Status

There are three conditions that affect whether you can access a file or record in multi-user dBASE IV: the *file open mode, file* or *record lock status,* and whether the file has been *encrypted* using PROTECT.

File Open Mode

When you USE a file, it is opened in one of two modes: exclusive or shared. The mode affects both the user who opened the file, and other users:

- A file opened as exclusive cannot be accessed by other users.
- If a file is opened as shared, you cannot perform certain commands detailed in the Exclusive Mode section later in this chapter (they require exclusive use of the file).

The file mode remains in effect until you either CLOSE the file, USE a different file in the same work area, or quit dBASE IV.

If you attempt to USE a file that is already in exclusive use by another user, or if you attempt to open a file exclusively that is already in use either exclusively or shared by another user, you will see the message **File is in use by another**.

Lock Status

Data collision is avoided in dBASE IV by allowing only one user at a time to perform certain commands on a file or record, such as changing data. While such commands execute, the file or record is locked, and other users are not able to execute another locking command on the same file or record until the first such command completes.

File locking is normally handled automatically by dBASE IV. You can also explicitly lock a file or record by using the FLOCK( ) or RLOCK( ) functions. Locking is described in detail in the Locking Files and Records section later in this chapter.

If you attempt to execute a command that requires access to a file or record that another user has either explicitly or automatically locked, you will see the message **Record** (or **File**) **is in use by another. Retrying lock, press ESC to cancel**, or, if the file has ever been CONVERTed, **Record** (or **File**) **in use by <user name>. Retrying lock, press ESC to cancel**.

Encryption

You may attempt to use a file and see the error message **File is encrypted**. This means that the system administrator has created a group of users, and only members of that group have been issued passwords for access to that file. You should ask the system administrator for access to such a file.

Exclusive Mode

A file opened with USE...EXCLUSIVE can only be read or edited by the person who opened it. It is the equivalent of locking the file as soon as it is opened and not releasing it until it's closed.

There are two ways to open a file for exclusive use with the USE command:

- Type USE <filename> EXCLUSIVE and press ↵. This command opens the named file for exclusive use.
- First type SET EXCLUSIVE ON ↵ (or put the EXCLUSIVE=ON line in your Config.db file and restart dBASE IV), then type USE <filename> ↵. After entering this SET command, USE opens each file for exclusive access until you SET EXCLUSIVE OFF. Note that SET EXCLUSIVE does not affect files already opened.

Commands That Require Exclusive Mode

There are some commands (such as MODIFY STRUCTURE) that require exclusive use. If one user were deleting fields while another user was trying to add data into those fields, dBASE IV would have no place to put the new data. Thus, to execute any of these commands, you must close the file (if it is open) and open it for exclusive use, assuming no other user already has it open.

The commands that require exclusive USE are as follows:

- CONVERT
- COPY INDEX
- COPY TAG
- INDEX ON...TAG <tag name> (where <tag name> is in an .mdx file)
- INSERT [BLANK]
- MODIFY STRUCTURE
- PACK
- REINDEX
- RESET
- ZAP

Locking Files and Records

Locking prevents multiple users from simultaneously writing to the same file or record. Locking occurs automatically when you execute certain commands, or an explicit lock can be executed. You can lock either an entire file or the individual record or group of records you are currently using.

Automatic Locking

Records are automatically locked when you use BROWSE or EDIT (or CHANGE, the synonym of EDIT) and press any key that changes a record, that is, any key except a navigational or menu key. dBASE IV attempts to lock the particular record being updated and any related records (see the Automatic Locking While USEing Multiple Files and Explicit Locking While USEing Multiple Files sections later in this chapter). This lock is released when you move from the record.

When you execute a command that modifies the entire file, dBASE IV automatically attempts to lock the file. Table 4-1 lists all commands that cause an automatic lock (if there is not an explicit lock already in place), and shows which lock the record and which the entire file. Note that if you specify a record number as the <scope> for DELETE, RECALL, and REPLACE, an automatic record lock is performed instead of a file lock.

Table 4-1 Commands that cause an automatic lock

Command	Locks	Command	Locks
APPEND FROM	File	JOIN	File
AVERAGE	File	LABEL FORM	File
BROWSE*	Record	PROTECT	File
CALCULATE	File	RECALL <scope>	File
CHANGE*	Record	REPLACE <scope>	File
COPY	File	REPORT FORM	File
COPY STRUCTURE	File	SORT	File
COUNT	File	SUM	File
DELETE <scope>	File	TOTAL	File
EDIT*	Record	UPDATE	File
INDEX	File		

*dBASE IV attempts to lock the record when a key is pressed that would change the data.

After you complete your update and press a navigation key to move to another record, dBASE IV automatically releases the file or record so other users can update it.

NOTE
You can also release the lock manually by selecting **Record unlock** *from the* **Records** *menu, or by pressing* **Ctrl**-O *if in full-screen mode.*

Locking does not prevent you from viewing a file or a record that another user is updating.

Automatic Locking While USEing Multiple Files

Automatic locking while BROWSEing multiple files using SET RELATION TO or CREATE/MODIFY QUERY is a straightforward process. Suppose you had two database files, one called States.dbf that contained three fields (STATE, CAPITAL, and POPULATION), and another file called Cities.dbf that contained the fields STATE, CITY, and MAYOR. Both files have indexes on their STATE field. Consider the following program, which is called Statecit.prg:

```
USE States.dbf IN SELECT() ORDER State ALIAS States
USE Cities.dbf IN SELECT() ORDER State ALIAS Cities
SELECT States
SET RELATION TO STATE INTO Cities
SET SKIP TO Cities
SET FIELDS TO States->STATE, Cities->CITY
BROWSE
```

The result would be a two-column BROWSE table that resembles the following:

STATE	CITY
AK	Anchorage
AK	Sitka
AK	Juneau
AL	Montgomery
AL	Birmingham
AL	Mobile
AR	Little Rock

Each row actually consists of two records: the STATE field from States and the CITY field from Cities. (You must use the SET SKIP TO command in order to see all the cities for each state.)

In this one-to-many RELATION, each record in States.dbf points to several records in Cities.dbf. Yet BROWSE will treat this as if it were a single file table for locking purposes. Thus, if you move the cursor to the first row, first column and begin editing, the entire row will be locked, including the AK record in States and the Anchorage record in Cities. When you move down to another row, these locks are released.

NOTE
Explicit locking (using the RLOCK() function) works differently. See Explicit Locking While USEing Multiple Files later in this chapter.

Disabling and Enabling Automatic Locking

You can use SET LOCK to enable or disable automatic locking for some of the commands listed in Table 4-1. The default is for locking to be enabled; to disable automatic locking for the following commands, you must issue SET LOCK OFF before you use any of them:

- AVERAGE
- CALCULATE
- COPY
- COPY STRUCTURE
- COUNT
- INDEX ON <key exp> TO <.ndx file>
- JOIN
- LABEL FORM
- REPORT FORM
- SORT
- SUM
- TOTAL

WARNING
These commands will work without locking, but data integrity is not guaranteed if automatic locking is disabled. If you SET LOCK OFF, there is a greater risk of possible inconsistent analysis, due to field values being changed by another user while you are performing calculations on them. See Transaction Problems in the For the Developer section of this chapter.

Some of these commands have two phases, reading and writing. For these commands, SET LOCK OFF works only on the reading phase. When you are writing to the file, it is automatically opened for exclusive use. These commands are:

- COPY
- COPY STRUCTURE
- INDEX ON <key exp> TO <.ndx file>

- JOIN
- SORT
- TOTAL

If a Record or File Is Already Locked

If you attempt to use one of the above commands on a record or file that another user has already locked, you see the message **Record** (or **File**) **is in use by another. Retrying lock, Press ESC to cancel** (in an application, a response other than **Esc** might be expected; ask the application developer if the error message is not clear). Press the **Esc** key to discontinue the attempted command. If the file has ever been CONVERTed, you will see the message **Record** (or **File**) **is in use by <user name>. Retrying lock, press ESC to cancel**. <user name> is the network log-in name of the user who initiated the lock. See CONVERTing Files for Network Use in the next section of this chapter for more information on CONVERT.

If the lock is successful, the system determines if any of the fields have changed since you accessed the record. If any have, the system displays the message **Record may have been changed (press SPACE)**. Press the **Spacebar** to continue.

Explicit Locking

As a user, you should treat manual locking in dBASE IV as an exception you will use rarely, if at all. A developer might use explicit locking more frequently. Manual (explicit) locking using the FLOCK( ) and RLOCK( ) functions is discussed in the For the Developer section of this chapter.

Printing in a Shared Environment

Printing in multi-user dBASE IV essentially works like printing in the single-user environment. The only difference is that you must choose whether to route the print job to a local printer or a shared printer.

PRINTER sends dBASE IV printer output to a local or shared printer. To send output to a shared printer, you must tell dBASE IV the name of the server, the printer, and the server port to which the printer is attached. This is network-dependent, and is covered in detail for each certified network in Chapter 3 of this manual, "Installing dBASE IV Onto a Network." The following example syntax assumes an IBM PC or 3Com 3+ network structure:

SET PRINTER TO \\<server name>\<printer name>=<port>

You must provide the server and printer names, which are assigned by the system administrator, and the particular port to which the printer is attached. For instance, if the server is called SERVER3, and the printer is EPSON_FX attached to LPT2 of SERVER3, the commands to print the report CENSUS are as follows:

```
. SET PRINTER TO \\SERVER3\EPSON_FX=LPT2
. REPORT FORM CENSUS TO PRINT
```

To send the report to the printer port at your local workstation, type a normal SET PRINTER command:

```
SET PRINTER TO <device>
```

The destination established by SET PRINTER remains in effect until you change it, or until you QUIT dBASE.

Transaction Processing

In general, a program intended for multi-user use must be slightly more complex than the same program intended only for use on a single workstation. For example, with other users having access to the same database files, lengthy database processes can be delayed or interrupted; a multi-user program must include some method of resolving this problem, should it occur.

dBASE IV provides a transaction processing capability, which keeps track of all changes to database files so you can restore them in the event that an updating process is interrupted. If recovery becomes necessary, you use the transaction log file to restore the values in your database files to those that existed before the transaction began.

You can break a lengthy dBASE IV session into several different transactions, which you can ROLLBACK to an intermediate step in the event of a problem. BEGIN TRANSACTION, END TRANSACTION, and ROLLBACK, which work equally well in single-user dBASE IV, are discussed in *Language Reference*. To use transaction processing in SQL, refer to Chapter 31 of *Programming in dBASE IV.*

Transaction Log File

The only difference between transaction processing in a multi-user and a single-user environment is the name of the transaction log file, which contains the record of changes made to every database file and related files since the last BEGIN TRANSACTION command.

NOTE
You can open a scratch file or lookup file without including it in the transaction log file with the USE...NOLOG command and keyword. In this case, if you issue a ROLLBACK command, the NOLOG database file will not be restored. See Chapter 2 of Language Reference *for details.*

In multi-user dBASE IV, the log file is named <workstation name>.log, rather than Translog.log (the single-user name). The <workstation name> is the same name you see associated with your workstation when you enter a LIST/DISPLAY USERS command.

Transaction Processing Example

The following is a complete example of a transaction in a multi-user environment. Every transaction processing command is used, and the example is annotated.

```
ON ERROR DO err_proc        && alternatively-ON ERROR ROLLBACK
BEGIN TRANSACTION           && start the transaction
 USE accounts               && all these commands contained within
 USE invoices IN SELECT()   && the transaction are "tentative" until
  *                         && it completes. A ROLLBACK command will
  *                         && restore all these records and files to
  *                         && their original values.
END TRANSACTION             && end the transaction
         *
         * The following CASE statement checks the outcome of the
         * transaction. It should immediately follow each transaction.
         *
DO CASE
 CASE .NOT. ROLLBACK()
         * ROLLBACK in err_proc was attemped and failed.
  @ 21,15 SAY "The rollback was not successful. You must"
  @ 22,15 SAY "restore from backup before continuing."
  RETURN TO MASTER
 CASE .NOT. COMPLETED() .AND. .NOT. ISMARKED()
         * ROLLBACK successful or not attempted (to get past previous
         * case), but transaction failed (.NOT. COMPLETED()).
         * Database file is unchanged, probably due to ROLLBACK.
         * Note that .NOT. COMPLETED() .AND. ISMARKED() can only occur
         * within the procedure err_proc. See err_proc below.
  RETURN TO MASTER          && maybe try the operation again
 CASE COMPLETED()
         * No action required, your transaction was completed.
  @ 21,15 SAY "Updates successfully completed."
ENDCASE
 *
 *
RETURN

PROCEDURE err_proc
choice = " "
DO CASE
 CASE ERROR() = 108         && "File is in use by another"
  IF .NOT. COMPLETED() .AND. ISMARKED()
         * Note that you need to check whether you are in the middle of a
         * transaction using this kind of IF statement because the same
         * error procedure is called from other parts of the program.
  @ 21,15 SAY  "One of the files that you need is in use"
  @ 22,15 SAY  "by someone else.  Retry? (Y/N) " GET choice
  READ
  IF UPPER(choice)='Y'
         RETRY
  ELSE
         ROLLBACK
    ENDIF
  ENDIF
 *
 CASE ERROR() = 109         && Other error responses follow a similar format.
 *
ENDCASE
RETURN
```

Table 4-2 lists the transaction functions' return values.

Table 4-2 Transaction functions' return values

Transaction status	COMPLETED()	ISMARKED()	ROLLBACK()
In the middle of a Transaction*	.F.	.T.	.T.
After a completed Transaction**	.T.	.F.	.T.
After a completed ROLLBACK**	.T.	.F.	.T.
If the ROLLBACK fails**	.F.	.T.	.F.

*Tested in an error procedure (after changes made to file)
**Tested after the END TRANSACTION

NOTE
If you use transaction processing in a multi-user environment, you cannot have two users logged in to the operating system with the same name creating transaction log files in the same drive and directory.

dBASE IV Security (PROTECT Command)

The PROTECT command is used to create and maintain security on a dBASE IV system. PROTECT is a menu-driven command issued within dBASE IV by the system administrator, who is responsible for data security.

PROTECT is fully described in the Restricting Access to Confidential Files section in Chapter 14 of *Using dBASE IV.* What follows is a brief description of concepts important to the multi-user environment.

Types of Security

PROTECT includes three distinct types of database protection:

- *Log-in security*, which prevents access to dBASE IV by unauthorized personnel
- *File and field access security*, which is used to define what files, and fields within files, each user can access
- *Data encryption*, which encrypts dBASE files so that unauthorized users cannot read them, even using disk utility programs

The system administrator creates a log-in name and password (a *user account*) for each user. This allows you to log in to dBASE IV. Note that you need a log-in name and password even to use database files on your own workstation hard drive, unless you have a local copy of dBASE IV as well.

You are assigned a user level from 1 to 8, with 1 being the highest level. Some files may be encrypted, which means only users with a particular user level can access them; in addition, access to particular fields within files can also be restricted to a higher user level.

You may have more than one user account, but each account is assigned to one and only one group.

The system administrator will assign you to a group for file and field access security. Your group must be specified at login.

In addition, each file is assigned to one and only one group. If the file group and your group do not match, you cannot access the file.

NOTE
To access files in two different groups, you must have two different user accounts, one for each group. To switch from one to the other, LOGOUT and log in using the other ID.

By using access levels within the group, the system administrator can give different users different kinds of access to the application program files.

Access by User Level

If you receive the message **Unauthorized access level** when you attempt to USE a file, then either the file requires a higher user level than you have to read it, or it is associated with a group different from your user account group. If you need to access files associated with more than one group, the system administrator will give you a different user account ID for each group (they can all have the same password, if the system administrator agrees).

If you are able to view the file (using either BROWSE or EDIT), but are unable to change any data, or are unable to add or delete a record, then the file requires higher access level than you have for that operation.

If you view a file and a particular field is missing from the view, then that field requires a higher access level than you have to read it. If the field is visible, but you cannot change it in any way, then the field requires a higher access level to modify it.

If you need access to a particular database file or field, contact your system administrator, who can change your user profile, increase your user level, or issue you an additional user account.

For the Developer

In general, developing applications for the multi-user environment is very similar to developing their single-user counterparts. You do need to take into consideration the fact that files or records may be temporarily unavailable due to locking or exclusive use by another user.

All but a few commands in your applications remain unaffected by moving from single-user to multi-user environments. The exceptions are noted in this section, as well as the changes in development style required when moving to a multi-user, networked environment.

Testing for a Network

An application intended to run in both multi-user and single-user dBASE IV should test to see which environment is currently running.

The NETWORK( ) function tests for the presence of multi-user dBASE IV. It returns a logical true (.T.) if the file Dbase415.acc exists in the CONTROLPATH= directory, or, if no such directory is defined, in the start-up directory. The following example tests for multi-user, then prompts to determine if the user wants to know who is logged in.

```
IF NETWORK()
  ACCEPT "Do you want to see who else is logged in?" TO answer
  IF UPPER(answer)="Y"
    DISPLAY USER
  ENDIF
ENDIF
```

Moving a Single-User Program to a Multi-User Environment

Because of automatic locking, dBASE IV single-user application programs can be moved to a multi-user environment with a minimum of programming effort. Automatic locking places a lock on a file or a record before executing any command that updates the database, and automatically releases the lock after the update. You generally need not place an explicit lock (using RLOCK( ), LOCK( ), or FLOCK( )) before commands that update a database file, and you need not execute an explicit UNLOCK after an update, unless you explicitly LOCKed a database file or record.

Before you run a single-user application program on a multi-user system, you must do the following:

- Enter multi-user timing mechanisms, where appropriate, to retry a lock on a file or record a specified number of times. See SET REPROCESS in *Language Reference*.
- Enter a SET LOCK OFF command, where needed, to disable automatic locking. See SET LOCK in *Language Reference*.

- When using a command that requires EXCLUSIVE (keyword) use, you must SET EXCLUSIVE ON before you USE the file, or use the keyword EXCLUSIVE with USE. See EXCLUSIVE USE of Files later in this chapter for a list of commands that require EXCLUSIVE use.

Optionally, you can also CONVERT the database file to multi-user. If you choose to do so, you will be able to use LKSYS( ), CHANGE( ), and other multi-user functions to determine information about users of the file and its lock status.

The rest of this section explains how multi-user dBASE IV programming differs from single-user programming, and how applications should be modified.

Application Considerations

Multi-user systems face considerations that do not occur in a single-user system. In a multi-user environment, users have access to shared data and program files, and developers often need to control this access.

Data Collision

One potential problem, *data collision*, is avoided by the use of automatic (or, on occasion, explicit) locking. Data collision occurs when two or more users attempt to alter the same data simultaneously.

Deadlock

Deadlock can happen when two users contend for each other's already locked files or records. For example, a deadlock occurs if user A needs to lock the file or record previously locked by user B, and user B attempts to lock the file or record already locked by user A.

A standoff occurs: neither user can access all the files or records necessary to complete a transaction. Instead of exiting, each application repeatedly attempts to reaccess an unavailable file or record. The transactions of both users enter into a potentially infinite loop, each waiting for the lock held by the other to be released.

A deadlock can also occur if two users contend for each other's files. This happens when multiple users are serially accessing each other's files and each is using a program sequence that includes SET EXCLUSIVE ON and USE.

The way to correct the deadlock is for one of the applications to recognize the situation and back out of it. Once the deadlock breaks, one user is able to access all necessary files and records to complete the transaction, after which the second user can proceed.

Resolving a Deadlock

In dBASE IV, the automatic retry facility helps users to recognize when they might be in a deadlock situation and to release their lock attempts.

The developer issues SET REPROCESS TO in the application before the potentially deadlock-producing commands (see Chapter 3 of *Language Reference* for a complete description of SET REPROCESS). SET REPROCESS can be set to one of three types of values: –1, 0 (the default), or a positive integer:

–1 dBASE IV will continue to retry the lock or USE until the other user releases the file or record. A message at the bottom of the screen indicates what has happened, but you cannot stop the retry process; it will continue until the record or file is released, or until you turn off your computer. This is the potentially infinite retry setting.

0 When you try to edit a locked record or database, dBASE IV retries infinitely, but displays a message indicating that another user has locked the file or record and allows you to press the **Esc** key to cease retrying. This is the default SET REPROCESS value.

n If you SET REPROCESS to a positive integer *n*, dBASE IV attempts to lock or USE n times. If it is unsuccessful, the message **Unable to LOCK (press SPACE)** is displayed at the bottom of the screen. When you press the **Spacebar**, the command is canceled.

SET REPROCESS With ON ERROR

If you have defined an ON ERROR routine, it will not be initiated until the SET REPROCESS loop terminates. Thus, if your application issued the command SET REPROCESS TO 30, and included an ON ERROR routine, the custom error routine would not initiate until after the lock or USE had been tried thirty times.

Note that this means if you SET REPROCESS TO –1, your ON ERROR routine will never execute, since the system retries until it is successful.

NOTE
An attempt to USE a database that another user has USEd EXCLUSIVEly takes much more time than an attempt to lock a database file or record that another user has already locked. SET REPROCESS accordingly.

Transaction Problems

Two or more transactions are *concurrent* if they run at the same time on two different workstations, generally initiated by two different users. See Chapter 14 of *Programming in dBASE IV* for a description of transaction processing and recovery.

Two problems can adversely affect concurrent transactions and produce incorrect results: lost update and inconsistent analysis:

- Lost update occurs if two users access and update the same record at approximately the same time. The first user may then make another change to the same record, unaware that it no longer matches the value that was just entered, and get an incorrect result.
- Inconsistent analysis occurs if user A calculates a series of database values (adds, subtracts, averages, and so forth) and user B changes any of the data that affects those values before user A completes the calculation. User A ends up with an incorrect calculation because of user B's interference.

Resolving Transaction Problems

Within a transaction, or within a program that is intended to run in a multi-user environment (even if not using transaction processing), there are two methods of resolving these problems: explicit locking, and the EXCLUSIVE USE of files. Both are described below.

Explicit Locking

In dBASE IV, locking prevents collision. In most cases, the automatic lock initiated by certain commands is sufficient (see Table 4-1 for a list of commands that initiate a file or record lock automatically). However, an application or transaction might need to explicitly lock a file or record for a series of updates to prevent another user from changing data before the first user finishes.

You can use explicit locking interactively or implement it within an application program.

For example, explicit locking can solve all three transaction-related problems above: if user A's application explicitly locks the record or file before the first update, then user B cannot change the data until user A's application finishes and unlocks it.

Locking Functions: FLOCK( ) and RLOCK( )

A locking function allows dBASE IV to determine whether a file or record is locked by another user, and if not, to lock it. Use FLOCK( ) to lock the entire file, and RLOCK( ) or LOCK( ) to lock the current record and all records which depend upon the current record. For complete syntax and instructions, see Chapter 4 of *Language Reference*.

If not previously locked, the locking functions:

- Return a logical true (.T.)
- Place a lock on the file or record

If the locking function returns .F., the file is already locked by another user, or the lock failed for some other reason.

Unlocking an Explicitly Locked File or Record

There are two methods of releasing a locked file or record:

- Using the UNLOCK command (described in Chapter 2 of *Language Reference*)
- Closing a database file with CLOSE, USE, CLEAR ALL, or QUIT

Full-Screen Locking

When in full-screen mode, you may lock the current record from the keyboard by pressing **Ctrl**-O. Press **Ctrl**-O again to unlock the record. You can also open the **Records** menu and select **Record lock**. Finally, if you simply begin typing in a record that is not already locked, you will get a message telling you to press the **Spacebar** to continue. When you press the **Spacebar**, the record locks until you move off of the record.

In full-screen mode, when a record is locked, **RecLock** is displayed in the multi-user section (the fifth section) of the status bar.

Explicit Locking While USEing Multiple Files

When you have multiple files connected by SET RELATION TO or a query, the RLOCK( ) function may lock more than one record.

If the file currently in use is the parent of a related file, then locking the current record also locks one of the related records in the child file. If the current record in the parent is related to more than one child record, then if SET SKIP is not set, the first related child record (using the index tag established for the child prior to linking) is also locked; if SET SKIP is set to the child, then the last related child record is locked instead.

For example, if the following program set up the relation between a parent called States.dbf and a child called Cities.dbf:

```
USE States.dbf IN SELECT() ALIAS States
USE Cities.dbf IN SELECT() ORDER State ALIAS Cities
SELECT States
SET RELATION TO STATE INTO Cities
```

If you are USEing States, and the current record is AL, then the RLOCK( ) command would not only lock the AL record in States, but also the Montgomery record in Cities. This is because Montgomery is the first city associated with Alabama when Cities.dbf is ordered using the State index tag, which can be seen by BROWSEing the linked files:

STATE	CITY
CA	Los Angeles
AL	**Montgomery**
AL	Birmingham
AL	Mobile
NY	Poughkeepsie

An expression can use a lock function to qualify or expand it, which enhances its usefulness to dBASE IV multi-user developers. An expression using FLOCK( ), RLOCK( ), or LOCK( ) is a logical expression with a value of either true (.T.) or false (.F.). For example:

```
SET REPROCESS TO 0
DO WHILE .NOT. RLOCK()
  DO timer
  IF times_up
    ? CHR(7)
    DO show_msg WITH "Time-out while waiting for record to unlock"
    ? CHR(7)
    ON KEY LOOP
    DO show_msg WITH "Press ESC to exit, space to continue waiting..."
  ENDIF
ENDDO
```

NOTE
With a different SET REPROCESS statement, the above DO WHILE loop would, each time, attempt the lock the number of times specified by SET REPROCESS before *moving to the DO statement. For most accurate results, ensure that REPROCESS is set to 0 (the default) before executing the above procedure. If REPROCESS is set to –1, the entire loop is redundant, since RLOCK() will either return .T. or will retry indefinitely; it will never return .F..*

EXCLUSIVE USE of Files

As an alternative to explicitly locking a file or record, your application can open a database file for use. There are two commands that do this:

- USE <filename> EXCLUSIVE. This command opens the file for exclusive use.
- SET EXCLUSIVE ON. After entering this command, all files opened will be opened for EXCLUSIVE use until you SET EXCLUSIVE OFF.

You can only open a file for EXCLUSIVE use if no other user is already using the file. If the file is already in use, you will see the message **File in use by <user name>** or **File in use by another**.

If a file is successfully opened for use, no other user may USE the file at all, not even to view it, until you CLOSE it. A file opened with the EXCLUSIVE file open mode need not be locked (nor do any records need to be locked) before executing any commands.

The following commands require the EXCLUSIVE file open mode before execution:

- CONVERT
- COPY INDEX

- COPY TAG
- INDEX ON...TAG <tag name> (where <tag name> is in an .mdx file)
- INSERT [BLANK]
- MODIFY STRUCTURE
- PACK
- REINDEX
- RESET
- ZAP

Editing Records in Multi-User dBASE IV

In a multi-user environment, in either the Browse or Edit (CHANGE) screens, the fifth section of the status bar displays the following file or record information, as appropriate:

File or record information	Status bar display
File USEd exclusively	**ExclLock**
File read only	**ReadOnly**
File locked	**FileLock**
Record locked	**RecLock**

Both BROWSE and EDIT (and CHANGE, the exact synonym of EDIT) attempt to lock the current record when you press any key that tries to update a record, that is, any key other than a navigation key. In addition, when two or more files are related in different work areas using the SET RELATION command, a lock is attempted on either the first or the last record in each relation chain, depending on whether SET SKIP is set.

If the locks are successful, and if the file has not been CONVERTed, the system displays the following message: **Record may have changed (press SPACE)** when you attempt to change data.

When you press the **Spacebar**, the system refreshes the screen, which now displays the current information, including any changes made since the last refresh prior to your update.

If the file has been CONVERTed, a link is continuously maintained between the file and the server, so the system will know whether the record has been changed or not since the last refresh.

See Chapter 2 of *Language Reference* for full information on BROWSE, CONVERT, and EDIT.

CONVERTing Files for Network Use

If particular files will be frequently used by multiple users, you can modify those files to give more multi-user information by using the CONVERT command. CONVERT adds a special, non-displaying field called _dbaselock to the database file. This field is between 8 and 24 characters long, the length specified by issuing CONVERT TO <expN>, where <expN> is an integer from 8 to 24 characters (16 characters is the default).

The field contains the following information for each record:

- The date and time that record was last locked
- Whether it has been changed since the last version displayed at your workstation
- The full or partial ID (depending upon what number you specify with the CONVERT command) of the user who last locked or changed the record

You can use LKSYS() to determine the date and time the record was last locked, and assuming you set _dbaselock to a number greater than 8, the user who last locked the record. Note that you might intentionally set the size of _dbaselock to 8 in order to prevent a user from determining the ID of the user who last locked the record.

CHANGE() returns either a .T. (if the file has been changed by another user), or an .F. (the file is unchanged).

NOTE
Once you have CONVERTed a file, you will no longer see the message **Record may have been changed (press SPACE)** *when you attempt to update a record. Instead, the system will use the _dbaselock information to determine whether the record has been changed. If it has not, you see no message; if it has, you see the* **Record has been changed (press SPACE)** *message. Also, if you attempt to lock a record that is already locked, and if the _dbaselock field is larger than 8 characters, the system will tell you the full or partial ID of the user who has locked the record.*

See Chapter 2 of *Language Reference* for full information on the CONVERT command, and Chapter 4 of *Language Reference* for full information on LKSYS( ) and CHANGE( ).

Updating the File View

Use SET REFRESH to specify the interval after which the file view is updated, to take into account data that may have been changed by other users. The default value of SET REFRESH is 0, which means the display is not automatically updated until you attempt to change a record.

See Chapter 3 of *Language Reference* for full information on SET REFRESH.

Errors and Error Recovery

There are four types of errors that can occur in an application running in a network environment:

- User error — the user enters an incorrect response.
- Program error — the application does not behave the way the developer expects.
- Network error — the network software or hardware improperly processes data transfer.
- Operating system error or hardware error.

Operating system, hardware, and network errors must usually be solved by the network administrator (who may or may not be the same person as the system administrator). This section will discuss recovery from the first two types of errors only.

Error Processing

How errors are processed in dBASE IV depends on whether an ON ERROR routine is being executed. If SET REPROCESS is set to any value other than 0, a file opening or accessing error condition will not be generated until the automatic retry facility is exhausted. Note that if SET REPROCESS is set to –1, an error condition will never be generated.

When dBASE IV generates an error condition, either it is captured through an ON ERROR procedure, or else dBASE IV displays the error in an error box or at the bottom of the screen.

CERROR(), ERROR(), FERROR(), and MESSAGE()

dBASE IV provides several functions for returning error values: CERROR(), FERROR(), ERROR(), and MESSAGE(). CERROR() returns the number of the last compiler error; FERROR() returns the error status of a low-level file input or output operation; ERROR() returns the error code; and MESSAGE() returns the message text associated with the error code.

CERROR() is updated each time the dBASE IV compiler executes; when a compiler error occurs, ERROR() returns 360, MESSAGE() returns **Compilation error**, and CERROR() returns the error number of the compilation error.

ERROR() and MESSAGE() are usually used within an error processing procedure. In a multi-user environment, you use the ERROR() function to trap recoverable error conditions, such as attempts to lock an already locked file or record. Note that you must know the error number of these recoverable errors.

Chapter 4 of *Language Reference* details the above functions with examples of their use.

RETRY

The RETRY command restores control to the calling program or procedure (parent) at the same line that called the child program. RETRY is frequently used within an ON ERROR procedure to re-execute a command following an error. RETRY differs from RETURN because RETRY executes the *same* line in the calling program, while RETURN executes the *next* line.

Chapter 3 of *Language Reference* describes the RETRY command and gives examples.

NOTE
The RETRY command is only used for capturing errors when a command file is being executed.

Restricting Access to Confidential Files

The PROTECT command is used to create and maintain security on a dBASE IV system. PROTECT is a menu-driven command issued inside dBASE IV by the database administrator who is either the system administrator, or anybody else delegated responsibility for maintaining the PROTECT system. The database administrator is responsible for data security. PROTECT can also be accessed using the **Protect data** option on the **Tools** menu.

PROTECT is optional: you don't have to use it. Once you have used PROTECT, however, the security system will always control access to database files.

PROTECT includes three distinct types of database protection:

- *Log-in security*, which prevents access to dBASE IV by unauthorized users
- *File and field access security*, which allows the system administrator to define what files, and fields within files, each user can access
- *Data encryption*, which enciphers dBASE files so that unauthorized users cannot read them

The dBASE IV security system is fully described in Chapter 14 of *Using dBASE IV,* "Using the Tools Menu." The rest of this section highlights certain developer aspects of the PROTECT command.

Encrypted Data

Confidential data can be enciphered so that it is unreadable without the password to decipher it. This process is call *encryption.*

Data encryption can only occur when the PROTECT security system is in place, which must be initiated by the system administrator. After initiating PROTECT, the system administrator may choose to give the PROTECT password to other users. Only a user who has the PROTECT password (different from the user's own password) will be able to encrypt a file or change access levels for particular fields or for the file itself.

Using the SET ENCRYPTION command, you can control when copied files are encrypted. SET ENCRYPTION OFF means that if you COPY the file within dBASE IV, the copy will not be encrypted. You must have access to the file (determined by the PROTECT utility) in order to COPY an encrypted file.

NOTE
In order to EXPORT, COPY STRUCTURE EXTENDED, MODIFY STRUCTURE, or use options of the COPY TO command, you must make an unencrypted copy of the file, as above.

Index files are only encrypted when you REINDEX or create them with an encrypted database file.

How Encryption Works

Some system administrators prefer to handle all encryption and file access settings themselves; others distribute the PROTECT password and allow each developer to encrypt and set access levels for the developer's own files. Every person who has access to the PROTECT password is a database administrator.

The system administrator should create all new user profiles (one for each account). Once that is done, each authorized user is assigned a user level from 1 to 8; 1 is the highest user level (and the default, unless changed), and 8 is the lowest user level. Each user is assigned to a group. Creating user profiles and assigning users to groups is explained in the For the System Administrator section later in this chapter.

Once user levels have been set, each individual .dbf file can be encrypted using the **Files** menu of the PROTECT utility.

Encrypting a File

1. From the **Files** menu in PROTECT, select **New file**.
2. Select the name of the file to be encrypted.

NOTE
You cannot select a file which is already encrypted. If you need to change the file access levels for a file that is already encrypted, return to the dot prompt, type SET ENCRYPTION OFF, *and copy the file to another filename. The new file will not be encrypted, and you can select it in PROTECT. Be sure to SET ENCRYPTION ON again.*

3. Select **Group name** and type the name of the group to which the file will belong. (Each file belongs to a single user group. Only members from that one group can access the file.)

4. Select **File access privileges** and set the minimum level required to access the file to **Read**, **Update**, **Extend**, or **Delete** it. These can be different, for example user level 5 or higher may be allowed to read the file, but user level 3 may be required to update it. You can also separately set access levels for each field in the file (see step 5 below).

   There are eight potential access levels, not all of which need be used. The system administrator can tell you which access levels have been assigned to users.

   If you want to set separate access levels for individual fields, then for each valid user level, do the following:

5. Select **Access level**.
6. Type the access level.
7. Select **Establish field privileges**.
8. For each field in the database, highlight the field name and press ↵ until the appropriate rights are displayed. The three possible rights are **FULL**, **R/O** (read-only), and **NONE**.
9. When all access levels are set for each field, select **Store file privileges**.
10. Proceed to the next file to be encrypted.

When you exit from PROTECT, all selected files are encrypted.

NOTE
Encryption time depends on the size of your files. Files are encrypted when you select **Save** *on the PROTECT* **Exit** *menu or exit PROTECT. If you are saving file privilege schemes of large files, it may take some time to exit PROTECT.*

Note that the database file is not yet encrypted. PROTECT creates an encrypted copy of the file, called <filename>.crp; if there is an associated memo file, the encrypted copy has the name <filename>.cpt. First, make an unencrypted backup copy of each .dbf and .dbt file; then check the unencrypted copies of the database and memo files; finally, copy the encrypted versions to <filename>.dbf and <filename>.dbt, respectively, and delete the .crp and .cpt files (otherwise you will use excess disk space).

Other Developer Security Considerations

When a PROTECT security system is in place, there are two functions which become available:

- USER( ), which returns the log-in name of the current user.
- ACCESS( ), which returns the access level of the current user.

These can be used to avoid access errors by denying access to users of an unauthorized level before attempting to access the file itself.

Note that the LOGOUT command logs the current user out (if PROTECT has been initiated) and returns the user to the log-in screen.

See Chapter 2 of *Language Reference* for complete descriptions of PROTECT and LOGOUT, Chapter 3 for SET ENCRYPTION, and Chapter 4 for USER() and ACCESS(). See also Chapter 14 of *Using dBASE IV* for a related description of the PROTECT system.

Other dBASE IV Multi-User Commands

Table 4-3 lists and describes other commands of particular interest in a multi-user environment. These commands are detailed in *Language Reference*.

Table 4-3 Other multi-user commands

Command	Description
CONVERT	Provides information for the CHANGE( ) and LKSYS( ) functions. Also provides change detection with automatic locking.
LIST/DISPLAY STATUS	Displays lock status along with the information displayed in a single-user environment.
LIST/DISPLAY USERS	Displays the names of users currently using dBASE IV on the network.
SET ENCRYPTION	Controls whether copied files are encrypted when created.
SET EXCLUSIVE	Determines the file open attribute of all database files USEd (after entering the command) during the dBASE IV session.
SET LOCK	Enables and disables automatic locking for a subset of commands.
SET PRINTER	Redirects printer output to shared or local printers.
SET REFRESH	Updates viewed database information if changed by another user.
SET REPROCESS	Sets the number of times dBASE IV tries a multi-user command or function before producing an error message.
USE EXCLUSIVE	Opens a database file and specifies that access is not shared.

Summary of Multi-User Commands and Functions

This summary describes each dBASE command and function used in programming for a multi-user environment.

Some of the commands and functions described are unique to multi-user programming, and some are used in both single-user and multi-user programming. If used in both, basic features are described in *Language Reference*, and multi-user functionality is described here. Before reading this section, you should be familiar with *Language Reference*.

Each of the commands and functions described in this section falls into a command class. Table 4-4 lists the various classes, while Tables 4-5 and 4-6 summarize the commands and functions themselves.

Table 4-4 Command classes

Command Class	Description
Database Security	Introduces security elements at the programming level
Editing of Data	Used to edit the data within a database
Server Access	Used to access a database server
Locking Control	Ensures data integrity in multi-user applications
Environmental Control	Establishes a dBASE IV environmental setting
Programming	Assists in the control and usage of command files
User Assistance	Provides on-line information
Opening Databases	Specifies how .dbf, .mdx, and .ndx files are opened

Table 4-5 Summary of multi-user programming commands

Command	Command Class	Intended Environment	Description
BEGIN TRANSACTION	Database Security	Single- and multi-user	Marks the beginning of a transaction
CHANGE/EDIT	Editing of Data	Single- and multi-user	Used to alter the contents of a record in the active database file
CONVERT	User Assistance	Multi-user	Provides information for the SET REFRESH command, the CHANGE( ) function, and the LKSYS( ) function

(continued)

Table 4-5 Summary of multi-user programming commands *(continued)*

Command	Command Class	Intended Environment	Description
LIST/ DISPLAY STATUS	User Assistance	Single- and multi-user	Provides information about the current session
LIST/ DISPLAY USERS	User Assistance	Multi-user	Provides information about the current dBASE IV users on the network
END TRANSACTION	Database Security	Single- and multi-user	Completes transaction and causes it to be committed (used with BEGIN TRANSACTION)
LOGOUT	Server Access	Single- and	Forces a user Security multi-user logout and allows a new user to log in
RESET	Editing of Data	Single- and multi-user	Changes the integrity tag in a database file
RETRY	Programming	Single- and multi-user	Re-executes a command that caused an error
ROLLBACK	Database Security	Single- and multi-user	Used with transaction processing to undo changes to a database file
SET	Environmental Control	Single- and multi-user	Displays and changes the current values of SET commands
SET AUTOSAVE	Database Security	Single- and multi-user	Automatically updates thedisk file and directories after I/O operations
SET ENCRYPTION	Database Security	Single- and multi-user	Establishes whether encrypted files are encrypted when copied

(continued)

Table 4-5 Summary of multi-user programming commands *(continued)*

Command	Command Class	Intended Environment	Description
SET EXCLUSIVE	Environmental Control	Multi-user	Determines the file open attribute of all database files subsequently opened during the session
SET LOCK	Locking Control	Multi-user	Enables and disables automatic locking for read-only commands
SET PRINTER	Environmental Control	Single- and multi-user	Redirects printer output to a network or local device
SET REFRESH	Editing of Data	Multi-user	Sets how frequently screen changes to reflect updates by another user
SET REPROCESS	Environmental Control	Multi-user	Sets the number of times dBASE IV retries a multi-user command or function before producing an error message
UNLOCK	Locking Control	Multi-user	Releases record and file locks
USE EXCLUSIVE	Opening Databases	Multi-user	Opens a database file and related files in the selected work area for exclusive use
USE NOLOG	Opening Databases	Multi-user	Opens a database file in the selected work area that will not be logged in the transaction log file

NOTE
See the EXCLUSIVE USE of Files section, earlier in this chapter, for a list of commands that require the database to be opened for exclusive use.

Table 4-6 Summary of multi-user programming functions

Function Name	Description	Output Data Type
ACCESS( )	Returns a user access level	N
CHANGE( )	Determines whether a record has been changed*	L
COMPLETED( )	Determines if a transaction has been completed	L
CERROR( )	Returns the compiler error code	N
ERROR( )	Returns the error code	N
FERROR( )	Returns the file I/O error code	N
FLOCK( )	Attempts to lock the database file and returns a logical .T. or .F.	L
ID( )	Returns the name of the current user	C
ISMARKED( )	Determines if a transaction is in progress	L
LKSYS( )	Determines who has locked a record or file and the date and time it was locked*	C
MESSAGE( )	Returns an error message character string	C
NETWORK( )	Determines if a network is present	L
RLOCK( )/LOCK( )	Attempts to lock a record and returns a success value*	L
ROLLBACK( )	Determines whether the last ROLLBACK command was successful	L
USER()	Returns the name of the user logged in to a protected system	C

The Sample Application

The sample application is an order entry program. The Library.prg program in that application illustrates basic multi-user programming concepts. This program includes procedures that do the following:

- Define a pop-up menu
- Add, change, and delete records

- Use automatic record locking
- Time out after waiting for a record to unlock
- Demonstrate use of the RETRY command

Library.prg contains code that is common to all modules in the order entry application. Multi-user code is set off by rows of asterisks. The entire application will run on a single-user or multi-user system, but all multi-user code is contained in one core module, Library.prg. The other modules in the application do not require, nor do they contain, any multi-user commands.

Follow these steps to run the order entry application:

1. Copy the sample files from their master directory to your own directory. If you have not copied the sample files, follow the procedures for using DBSETUP in Chapter 1 of this manual.
2. Change to the SAMPLES directory, then type DBASE and press ↵.
3. From the dot prompt, type DO BUSINESS and press ↵.

For the System Administrator

The primary functions of the system administrator are to install, set up, and maintain the shared systems, including dBASE IV; to create user profiles, maintain passwords, and set file access levels; and to perform regular backups and other administrative tasks.

Installing dBASE IV Onto a Network

Installation is highly dependent upon the specific network on which dBASE IV will be installed. Full instructions for all certified networks are found in Chapter 3 of this book.

WARNING
Do not attempt to install dBASE IV onto the root directory of your server: if you give users rights to the root directory, they might inherit the same rights for all subdirectories. Also, the operating system may only allow a limited number of files to be copied to the root directory of a server.

Installing and configuring printers is also network-dependent. Refer to Chapter 3 of this manual.

In addition to the directories associated with single-user dBASE IV, multi-user dBASE IV also requires a directory to hold the user-count file (Dbase420.acc) and, if you initiate the PROTECT system, the security file Dbsystem.db. Normally, this is the dBASE IV directory, but you can put these two files in any directory as long as you also change the CONTROLPATH= line in Config.db to point to the right directory.

This directory must be both readable and writable by all users.

NOTE
If a user has already installed single-user dBASE IV, then a copy of Dbase420.acc was copied to the user's local hard disk. When dBASE IV is started, it will look at the local Dbase420.acc instead of the shared file created when the system administrator installed multi-user dBASE IV. Thus, even if you give this user a valid user name and group affiliation (stored in the shared Dbase420.acc), it will not be used by the single-user dBASE IV, and files encrypted with PROTECT will be unreadable.

To force the single-user dBASE IV to read the shared Dbase420.acc, set the CONTROLPATH= line in the user's Config.db file to the drive and directory where the shared Dbase420.acc is stored (generally the dBASE IV directory on the server).

User Count

Any number of users may have accounts on a multi-user dBASE IV system (see Setting Up User Accounts later in this chapter). However, the number of users who can use the system *simultaneously* is limited by the installed *user count* contained in Dbase420.acc.

When you first install multi-user dBASE IV, only one user at a time is authorized to use the system. For every copy of the dBASE IV LAN Access Pack version 2.0 that you install, you increase your allowed user count by the number specified for the product.

Follow the instructions below to install or uninstall user counts. Adduser4.exe both installs and uninstalls user counts. It can be run in either menu or command line mode.

Adduser4 Menu Mode

1. From the operating system prompt in the dBASE IV directory (or the directory that contains the file Dbase420.acc), type ADDUSER4 and press ↵.
2. From the menu that appears, select **1. Increase User Count** or **2. Decrease User Count** by typing 1 or 2.
3. Type the letter of the drive into which you will insert the LAN Access Disk, for example, a.
4. Insert a LAN Access Disk and press any key to continue. To add user counts, use a disk whose user counts have not already been added; to uninstall user counts, use a disk whose user counts have already been added to the current copy of dBASE IV version 2.0.

   You can also insert a dBASE IV version 2.0 System Disk 1 instead of a LAN Access Disk. Follow the screen prompts and type in the requested LAN Access number printed on your disk label.

5. Press **Esc** to return to the main menu screen when the user counts have been added or subtracted.

6. When finished increasing or decreasing user count, select **3. Quit** from the main menu by typing 3.

Adduser4 Command Line Mode

You can bypass the Adduser4 menu by using the command line mode. Use the following syntax:

```
ADDUSER4 [-action] [[Dbase420.acc drive and directory]
        [LAN Access Disk serial number]]
```

where -action is:

–i to *increase* user count

–d to *decrease* user count

–r to *report* (display) user count

–m to display the Adduser4 menu, as in the previous section (same as no -action parameter)

For example, to increase the user count in the Dbase420.acc file in D:\DB4_ACCT, you would type the following at the operating system prompt:

```
ADDUSER4 -I D:\DB4_ACCT <serial number> ↵
```

After you enter the command line, you will be prompted to insert the LAN Access Disk. The user count in Dbase420.acc is increased or decreased as in step 4 above.

Setting Up User Accounts

Using the PROTECT system, you can limit use of the copy of dBASE IV loaded onto the server to specific individuals for whom you have set up a *user account.*

If you do not use PROTECT, anybody can use dBASE IV from any workstation that has been given access rights to the server's dBASE IV directory. Even without PROTECT, the number of users who can use dBASE IV simultaneously is still limited by the user count.

You will generally have more user accounts than actual users attempting to access the system simultaneously. This is because not everybody needs to use the system all day, every day, and also because the same person can have several different user accounts (one for each group to which the user belongs, if you are using PROTECT).

Only one user can be logged in to a single workstation at a given time.

If you initiate the PROTECT system, users are required to type a valid group and user name (account) and the corresponding password in order to use the copy of dBASE IV loaded on the server. Thus, you maintain strict control over who can use the system.

In addition, each such user receives a user level, which determines which files the user is able to read, update, extend, or delete, and which specific fields in those files the user can read and change.

Initiating PROTECT

From the dot prompt, type PROTECT and press ↵, or select **Protect data** from the **Tools** menu at the Control Center.

The system will ask for a password. From this point forward, the password must be used by anyone who needs to access the PROTECT utility, so do not lose it. The password can be up to 16 characters long; for security purposes, it is recommended that you use all 16 characters for your password. Press ↵ when you finish typing the password.

PROTECT asks you to repeat the password; subsequent use of the utility only requires the password to be typed a single time. PROTECT is now initiated.

Adding Users

To add a user account, do the following:

1. From the **Users** menu, select **Login name** and create a log-in name from one to eight characters long. Press ↵.
2. Select **Password** and type a password from 1 to 16 characters long. Press ↵.
3. Select **Group name** and type a valid user group name. You can have any number of group names, but each file that you encrypt will be usable only by members of the one group you associate with that file. Any number of users may belong to the same group, but each user may only belong to one group. To belong to more than one group, an individual needs more than one user account (one for each group).

   To use an encrypted file associated with a group (for example, ACCOUNTS), a user logs in, entering the user ID that is a member of ACCOUNTS. If the user is already logged in under a user ID that belongs to a different group, the user must first LOGOUT, then log in again using the ACCOUNTS user ID.
4. Select **Full Name** and type the full name of the person who will use this account.
5. Select **Access level** and type an integer from 1 to 8. This number controls what rights the user will have for encrypted files. 1 is the highest user access level, and 8 is the lowest.

NOTE

There is no intrinsic meaning to the access level numbers other than the relative values (a file can only be accessed by users with user level of n *or higher) and the values you assign to files and fields using the* **Files** *menu. You do not have to use all possible access levels; you could set up a system that used only levels 2 and 4, for example. Each person would be either a 2 or a 4, file access levels would be either 2 (usable by level 2 users only) or 4 (usable by everybody), and only field access levels 2 and 4 need be set for encrypted files.*

6. Select **Store user profile**, and move to the next user. Remember to set up an account for yourself (give yourself user level 1).

To delete or edit a user account, type the **Login name**, **Password**, and **Group name** (press ↵ after each entry). The system will recognize that the user is already known, and will ask whether you want to edit the user's profile. Answer **Y**, then either change information or select **Delete user from group**.

Security Considerations

If you use the PROTECT utility, it is extremely important that you make frequent backups of the Dbsystem.db file. This file stores all the file and field level protection requirements you have set up.

Can Dbsystem.db Be Read?

Dbsystem.db is encrypted, and can only be read or properly altered by a user who has access to the PROTECT password.

Any user with sufficient access to the dBASE IV directory (or the directory specified in the CONTROLPATH= line in Config.db) can corrupt the file or delete it altogether, unless your network allows file-specific delete/no-delete rights. With proper preparation, recovery from such a situation is not difficult.

If Dbsystem.db Is Deleted or Corrupted

If Dbsystem.db is deleted, then any user will be able to run dBASE IV, subject only to the user count limits on simultaneous users. However, any files which have been encrypted will still be unreadable — not only to unauthorized users, but to authorized users as well.

The first indication of a deleted or corrupted Dbsystem.db file occurs when you type DBASE ↵ and do not see the log-in screen before dBASE IV starts.

If you have a current backup of Dbsystem.db, copy it to the dBASE IV directory or CONTROLPATH directory. The system will now function exactly as before.

You can still restore the system if you do not have a current backup, but it is a more detailed process.

If There Is No Backup for Dbsystem.db

The key for encrypted files depends not only upon the PROTECT utility password, but also upon the group name associated with the file. Thus, you must retain a record of all group names associated with encrypted files, as well as a list of what file and field level access rights have been granted to each encrypted file.

The easiest way to do this is to make frequent printouts of the **User Information** and **File Information** choices from the **Reports** menu of PROTECT. Be sure to store these printouts in a secure location, as the user information report includes a full list of all user account IDs, group names, and passwords. The file information report comprises encryption information for each file.

To restore the security system in the absence of a current backup of Dbsystem.db, do the following:

1. Initiate PROTECT again. Type the same password as before, but twice (dBASE IV thinks it is a new PROTECT system because Dbsystem.db is missing).
2. Create a user profile for yourself for each group associated with encrypted files. Give each account user level 1.
3. Exit PROTECT.
4. For each encrypted file, USE the file, SET ENCRYPTION OFF, and COPY the file to another filename. This is necessary because you can only add encryption to an unencrypted file.
5. Verify the integrity of the resulting unencrypted file before you DELETE the original file.
6. DELETE the original file — you can RENAME the unencrypted file to the original filename after you delete the original (encrypted) file.
7. Enter PROTECT again. From the **Files** menu, select **New file** and proceed as you normally would to retype all file and field level protection for the file.
8. Do this for each encrypted file. When you finish, set up user accounts for all users again, assigning them to the appropriate groups.

The system is fully restored. Be sure to make a backup of Dbsystem.db.

If a User Cannot Log In

The most likely explanation is that the user is typing an incorrect group name, user name, or password. If three attempts to log in are made unsuccessfully, the system terminates with an error message, **Unauthorized log in**.

When the user informs you of the inability to log in, compare the information in the **User Information** report to the information the user has. If necessary, you can change any element of the user profile by editing it. See Adding Users earlier in this section.

If a User Cannot Access a File or Field

The three possibilities are that the user has an insufficient user level to access the file or field, is a member of the wrong group, or read or write access is denied to the file by the operating system or network.

The first and second possibilities are far more likely than the third; if the user requires access, but is of too low an access level, you can edit the user profile and increase the user level. Check **File Information** (from the **Reports** menu of PROTECT) to determine what user level is required for each type of file or field access.

You can change the user's group affiliation, or create an additional user profile with the necessary group affiliation.

If a file has the operating system attribute of read only, dBASE IV treats it as if no one has sufficient access level to do anything but view it. If it has the operating system attribute of hidden and it is already in a catalog, you can still USE it. You must change the operating system hidden attribute from DOS. See your DOS manual ATTRIBUTE command to change the file attribute. If it is hidden and not in a catalog, dBASE IV will not see it and cannot edit it.

Protecting and Recovering from Problems

Three commands, MODIFY STRUCTURE, MODIFY COMMAND/FILE, and CONVERT, actually create a backup as they operate. In addition, two other commands, PROTECT and REPLACE ALL, utilize methods to avoid data errors.

- MODIFY COMMAND/FILE and MODIFY STRUCTURE create backups of the files edited. The backups have the same names as the original file, with the extensions listed in Table 4-7.

Table 4-7 File extensions

Original file extension	Backup file extension
MODIFY COMMAND/FILE	
.prg	.bak
.txt	.bak
MODIFY STRUCTURE	
.dbf	.dbk
.dbt	.tbk
.mdx	.mbk

- CONVERT first copies the original .dbf file to a backup file with the extension .cvt, then creates a new .dbf file with the _dbaselock field.

- When PROTECT encrypts a file, it leaves the original .dbf and .dbt files intact and creates encrypted copies with the extensions .crp and .cpt, respectively. You have to subsequently copy the .crp file to the .dbf file and the .cpt file to the .dbt file in order to USE the database file in encrypted mode.

  Thus, you can make multiple encrypted copies of a single database file with different group affiliations and access levels (stored in different directories, or given different names), so long as you do not copy one of the encrypted files back to the .dbf file. Once you do, you must decrypt it by SETting ENCRYPT OFF and COPYing it to another filename before you can again encrypt the file (PROTECT cannot be used with a file that is already encrypted).

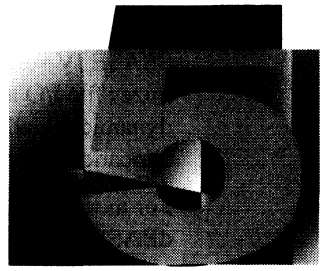
- REPLACE ALL (but not any other scope of REPLACE) does not actually write the replaced records in a database file until either all the records have been replaced in memory or the 64K REPLACE buffer runs out of memory. Thus, if REPLACE ALL is interrupted, and the file is less than 64K, either all records will have been replaced or no records will have been replaced. With any other scope (even REPLACE FOR .T.), it is possible that an interruption will result in some, but not all, records being replaced.

What's New

What's New in dBASE Version 2.0

Changes Since dBASE III PLUS

What's New in dBASE IV Version 2.0



dBASE IV version 2.0 performance has been dramatically increased with enhancements that will significantly improve the productivity of both end-users and developers in both single and multi-user environments.

This chapter describes the changes and enhancements from dBASE IV version 1.5 to version 2.0.

Improved Performance

dBASE IV version 2.0 uses new filter optimization and memory management techniques to improve performance significantly. dBASE IV uses these new techniques dynamically: no special programming or configuration is required.

New High-performance Filter Optimization

New automatic filter optimization techniques let dBASE IV access sets of records very efficiently. You can query a large database for a small set of records and expect dBASE IV to return the results faster than before. For more information, read Chapter 34, "Optimizing Your dBASE Applications," in *Programming in dBASE IV.*

Virtual Memory Management

dBASE IV version 2.0 uses a Virtual Memory Manager (VMM), which lets dBASE access 5MB or more of extended or virtual memory. With VMM, you can run dBASE IV on systems with as little as 1MB of extended memory. VMM runs automatically when you start dBASE IV and evaluates your system's configuration. It then uses internal configuration values to determine how to manage your system's extended memory and disk space for the best performance. If you prefer, you can change VMM's default settings to create a more custom-fit environment for your application development.

For more information about virtual memory and VMM, read Chapter 33, "Optimizing dBASE IV," in *Programming in dBASE IV.*

Data Buffer Manager

dBASE IV version 2.0 provides a new data-buffering management system that optimizes file I/O and eliminates the need for external disk-caching software. It dynamically allocates memory to dBASE's data buffers. You can customize memory allocation to this buffer by using the DOS environmental variable, DBASEIV_BUFF.

For more information about configuring the data buffer, read Chapter 33, "Optimizing dBASE IV," in *Programming in dBASE IV.*

DOS Protected Mode Interface (DPMI) Support

dBASE IV version 2.0 supports DPMI when running under Windows and OS/2.

Language Enhancements

dBASE IV version 2.0 adds many new commands and functions that are described in the following subsections.

New and Enhanced Menu Commands

The twelve new or enhanced menu commands let you define more actions for menu bars and pads. You can now execute commands when users select (highlight), choose (activate), or exit specific menu bars and pads. You can also assign the same command to several or all menu bars or pads.

The following are the new and enhanced menu commands:

ON BAR

Executes a command when users select (highlight) a specific bar in a pop-up menu.

ON POPUP

Executes a command when users select any bar in a pop-up menu that has not been assigned an ON BAR command.

ON PAD (enhanced)

Executes a command when users select a specific pad in a bar menu. Previously, ON PAD only activated a pop-up menu.

ON MENU

Executes a command when users select any pad in a bar menu that has not been assigned an ON PAD command.

ON SELECTION BAR

Executes a command when users choose (activate) a specific bar in a pop-up menu either by pressing Enter when the cursor is on the bar, or by clicking the bar.

ON SELECTION POPUP (enhanced)

Executes a command when users choose any bar in a pop-up menu that has not been assigned an ON SELECTION BAR command. Previously, an ON SELECTION POPUP command applied to all bars in the pop-up menu.

ON SELECTION PAD (enhanced)

Executes a command when users choose a specific pad in a bar menu. In dBASE IV version 2.0, you can use ON SELECTION PAD with ON SELECTION MENU to define unique actions for individual pads and a "global" action for all other pads in a bar menu.

ON SELECTION MENU

Executes a command when users choose any pad that has not been assigned an ON SELECTION PAD command.

ON EXIT BAR

Executes a command when users move the cursor off a specific bar in a pop-up menu.

ON EXIT POPUP

Executes a command when users move the cursor off any bar that has not been assigned an ON EXIT BAR command.

ON EXIT PAD

Executes a command when users move the cursor off a specific pad in a bar menu.

ON EXIT MENU

Executes a command when users move the cursor off any pad in a bar menu that has not been assigned an ON EXIT PAD command.

For more information about the new and enhanced commands, read Chapter 2 of *Language Reference*.

New Menu Functions

dBASE IV version 2.0 adds three new menu functions that return information about menu pads and bars.

BARCOUNT()

Returns the number of bars in the active or specified pop-up menu.

BARPROMPT()

Returns the prompt text that appears in a specific bar of the active or specified pop-up menu.

PADPROMPT()

Returns the prompt text that appears in a specific pad of the active or specified bar menu.

For more information about the new functions, read Chapter 4 of *Language Reference*.

New Mouse Commands and Functions

dBASE IV version 2.0 adds new mouse commands and functions to give you more control over mouse actions.

SET MOUSE ON/OFF

Enables or disables the mouse cursor. Use SET MOUSE OFF to remove the mouse cursor from the screen and ignore all subsequent mouse actions. You can specify the mouse setting in the Config.db file.

ON MOUSE

Detects a left mouse button click, then executes a command after the button is released.

ISMOUSE()

Returns a logical true if a mouse driver is installed. Otherwise, it returns false.

MROW()

Returns the row position of the mouse pointer on the screen.

MCOL()

Returns the column position of the mouse pointer on the screen.

For more information about the new mouse commands and functions, read the *Language Reference*.

Other New and Enhanced Commands and Functions

SET IBLOCK TO

Lets you specify the size of the index block that dBASE IV uses when it creates new index (.MDX) files. You can set IBLOCK in the Config.db file.

SET MBLOCK TO

Lets you specify the size of the memo blocks that dBASE IV uses when it creates new memo field (.DBT) files. You can set MBLOCK in the Config.db file.

SET LDCHECK ON/OFF

Enables or disables dBASE's capability to check for language driver compatibility. You can specify the LDCHECK setting in the Config.db file.

SET DISPLAY TO (enhanced)

Lets you specify a monochrome or color monitor, and the number of lines your screen should display. dBASE IV version 2.0 adds three more options: VGA25, VGA43, and VGA50.

AT() (enhanced)

Returns a number that shows the starting position of a character string (substring) within another string or memo field. AT() starts the search from the first character of the string or memo field being searched.

In dBASE IV version 2.0, you can add a third argument which returns the nth occurence of the substring.

RAT()

A variant of AT(), RAT() returns a number that shows the starting position of a character string (substring) within a larger string or memo field. RAT() starts the search from the last character of the string or memo field being searched.

FLDLIST()

Returns the fields and calculated field expressions of a SET FIELDS TO list.

KEYMATCH()

Returns a logical true (.T.) or false (.F.) to indicate whether or not a specified expression is found in the index keys of a specified index. A primary use of KEYMATCH() is to check for duplicate values during APPEND.

MEMORY() (enhanced)

Returns the amount of memory, in 1024-byte increments, that is available in or allocated to various memory regions.

In dBASE IV version 2.0, you can enter eight different values (0 - 7) to get different types of information about your system's memory allocation.

LASTKEY() (enhanced)

Returns the ASCII value of the last key pressed.

In dBASE IV version 2.0, you can use LASTKEY() to detect if the last action during a keyboard wait state was a mouse click. If it was, LASTKEY() returns -100.

TIME() (enhanced)

Returns the system time. In dBASE IV version 2.0, you can specify a valid expression to get the time in HH:MM:SS.hh format.

Other New Features

Increased Array Sizes

You can now declare an array with as many elements as your system's virtual memory will support, as long as one dimension does not have more than 65,535 elements. The previous version of dBASE allowed a total of 1,170 array elements.

New VGA Video Modes

dBASE IV version 2.0 supports VGA mode displaying the following number of lines: 25, 43, and 50.

New Block Size Limit

dBASE IV version 2.0 extends the maximum block size limit you can specify with the SET BLOCKSIZE command from 32 to 63.

New Debugger Keys

You can use the following new keys in the Debugger:

Ctrl-T — toggles the breakpoint on the selected line

Ctrl-L — executes the selected command line

PgUp, PgDn, Ctrl-PgUp, Ctrl-PgDn — pages through the source code

Improved Error Trapping

dBASE IV version 2.0 extends its error trapping capabilities to handle the following errors:

ERROR #	Message
56	Disk full when writing file:<filename>
210	Production .MDX file not found:<mdxname>
289	Production .MDX file is damaged
526	Index expression needs missing memory variable:<mvarname>
527	Memo file not found:<name:dbtfile>

Support For Two Code Pages

You can now use dBASE IV version 2.0 on computers configured with DOS code page 437 or 850. Previously, the US version of dBASE IV supported only code page 437. Code page 850 is the default code page used in most systems in other countries. For more information about code pages, refer to your DOS manual.

Compatibility-checking of Language Drivers

dBASE IV uses different language drivers to support different DOS code pages (character sets) and language requirements. dBASE IV version 2.0 adds new language driver ID-checking functionality that warns you of conflicting language drivers. For example, if you try to open a dBASE file created with a different language driver, dBASE displays a message and prompts you for an action.

This language driver-checking functionality is important for users who share dBASE files created with other international versions of dBASE, or different dBASE configurations. The configurations, set through DBSETUP, that affect the language drivers are: LANGTABLES ON/OFF and ASCIISORT ON/OFF.

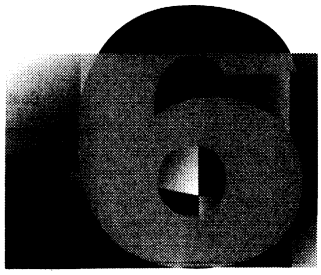
The correct language driver is required in order for dBASE to create indexes properly, know which characters are valid, and convert characters from uppercase to lowercase (and vice versa).

For more information, see the SET LDCHECK command in Chapter 3 of *Language Reference*.

Options that Have been Removed

dBASE IV version 2.0's new virtual memory and data buffer managers replace DBCACHE, HyperDisk, and DBHEAP.

Changes Since dBASE III PLUS



Using This Chapter

"Changes Since dBASE III PLUS" is designed to introduce users of earlier versions of the dBASE product to dBASE IV version 2.0. There are four possible paths such users may follow:

- Upgrading to version 2.0 from version 1.1
- Upgrading to version 2.0 from version 1.0
- Upgrading to dBASE IV version 2.0 from dBASE III PLUS or dBASE III®
- Upgrading from an older dBASE product

For those upgrading from a dBASE product earlier than dBASE III or dBASE III PLUS, there are so many improvements and enhancements that it is easier to consider it an entirely new product than to try to learn all the "changes."

This chapter addresses those upgrading either from dBASE III PLUS or from dBASE IV version 1.0. In both cases, there are many changes that apply to one path but not to the other. The best approach is to read whichever of the two sections applies to your own situation; if you also read the other section, bear in mind that it is written for a different audience and may not apply to you.

NOTE
Because the dBASE IV object code format is changed, you need to recompile all object code files from all earlier versions of dBASE IV.

Figure 6-1 shows the major milestones for each version of dBASE IV.

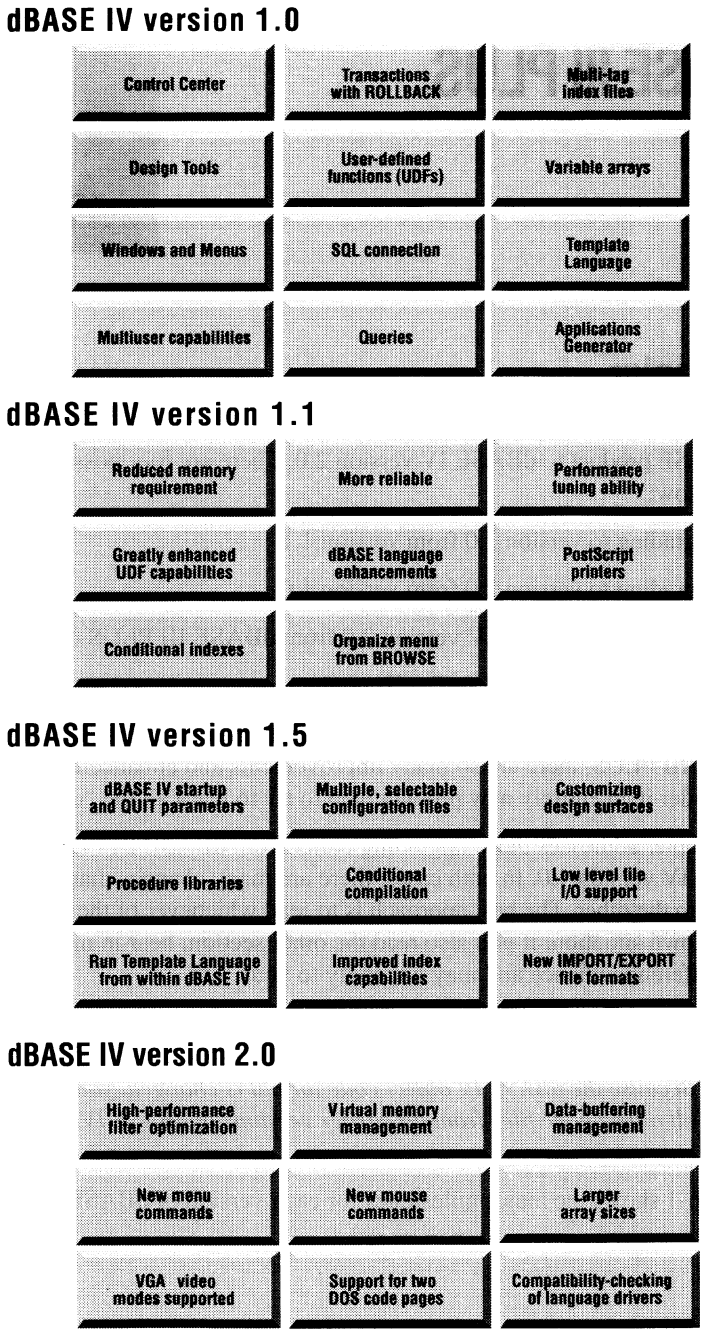


Figure 6-1 Major milestones for each version of dBASE IV

Upgrading From dBASE III PLUS

dBASE IV has many new features that enhance its power for both users and applications developers. This section highlights many of the basic differences between the general working environment of dBASE IV and dBASE III PLUS.

The changes divide into four basic sections:

- Performance improvements
- Interface enhancements
- General functionality improvements
- New functions and command enhancements

Performance Improvements

dBASE IV version 2.0 uses new filter optimization and memory management techniques to improve dBASE's performance significantly.

Filter Optimization

New filter optimization techniques let dBASE IV access sets of records very efficiently. You can query a large database for a small set of records and expect dBASE to return the results quickly.

Virtual Memory

dBASE IV's new virtual memory manager automatically determines how to manage your system's extended memory and disk space for the best performance.

Data Buffer Management

dBASE IV includes a new data-buffering management system that optimizes file I/O and eliminates the need for external disk-caching software.

Interface Enhancements

A completely new interface allows you to harness the features and power of dBASE IV without needing to write statements at the dot prompt. Every full-screen component of dBASE IV, from APPEND to SET, features a full set of pull-down menus for ease of operation. These new menus are supplemented by the new dBASE IV keyboard macro feature.

The screen designers for queries, forms, labels, and reports share a common "what you see is what you get" (WYSIWYG) approach to designing. With the improved catalog system, you can easily keep track of which forms, labels, and reports work with which database files.

BROWSE and EDIT are more closely related, with a special function key to flip between these two methods of displaying data.

The dBASE III PLUS approach to handling views and queries has been enhanced to offer a direct and powerful way to join, filter, and project data from multiple files.

The Control Center

The Assistant has been replaced by the more powerful Control Center. You can use the Control Center to open and close files, and access the menu system to establish views, reach design screens, run programs, and manage files and catalogs. See *Using dBASE IV* for a complete discussion of the Control Center.

Design Tools

dBASE IV has introduced a number of design tools to help you quickly create and modify the view queries, forms, reports, and labels you need. All of the following tools are described in *Using dBASE IV*, and complete instructions are given for their use:

Queries and Views

The queries design screen offers a new work surface for creating the queries that define views. You can more easily join files, filter their records, and choose which fields to display. You can also create calculated fields and global filters for any query. Special update queries help you automatically change many records at once.

Both the fields used as selection criteria and the fields to be displayed are shown at the same time on the screen. Select fields for viewing with a single keystroke.

Forms

Using the forms design screen, you can lay down all the fields and field names at once with an automatic layout, more easily move objects around the screen, and copy text, fields, and boxes to other places on a form. Calculated fields are also available. You can even create the form first, then create a database file that will fit the form you just designed.

Reports

The reports design screen is a WYSIWYG layout surface. Reports can include calculated fields, summary fields, and special fields such as the system date or a running page number. A full menu of text styling options can be used to add bold or italics, as well as different type fonts and pitches.

Labels

The labels design screen uses the layout surface common to dBASE IV. You simply place fields where you want them, just as you do in the reports design screen. The labels generator offers full support of text styling and print settings, including automatic setup for the nine most common label sizes.

Fields can be added from the underlying database file or view. You can also use new calculated fields and special fields such as the current date or time.

Applications Generator

The Applications Generator is a menu-driven tool for developing dBASE IV applications without having to write a single line of code. You create the structures, objects, and connections between them using the menu system, and let dBASE IV generate all the code for you. See *Using dBASE IV* for complete instructions.

Browse and Edit

In dBASE IV, you can toggle directly between the Browse and Edit screens using the **F2 Data** key.

Help System

Context-sensitive Help boxes provide information for using the menu system and the Applications Generator. They also offer syntax and examples for all the commands and functions in dBASE IV.

General Functionality Improvements

Changes to many aspects of database files make dBASE IV more powerful and efficient than dBASE III PLUS. See *Language Reference* and *Programming in dBASE IV* for more information on the following improvements.

Increased Database Structure Limits

Database structures can now contain up to 254 fields per record. dBASE IV supports floating point numbers (type F).

Data Encryption

The PROTECT utility is now immediately available from the dot prompt and Control Center, and is no longer an external utility. In addition, memo fields may now be encrypted, either explicitly through the PROTECT command or automatically when the related database is encrypted.

Data Restoration

dBASE IV now supports full transaction processing with rollback capability. If an error should occur during the transaction, any data and index files involved may be automatically restored with the ROLLBACK command. See *Language Reference* for information about transaction processing.

Error Dialog Boxes

Error handling from the dot prompt has been improved. When an error occurs, you may now select options to edit the erroneous command line, get help, or cancel the command.

Indexing

dBASE IV has implemented several major enhancements to indexes:

Multiple-tag Index Files

dBASE IV uses a new method of indexing that is able to maintain up to 47 index tags in a single file called an .mdx (multiple index) file. Each index tag is similar to an old .ndx file, and consists of a tag name and key expression. When a database structure is created or modified, you can specify any (or all, up to 47) fields as an index field. If you do, a *production* .mdx file is created with the same name as the database file, and automatically opened when the .dbf file is USEd. You may have more than one .mdx file active at once, which allows you to have literally hundreds of indexes open at the same time.

dBASE IV maintains complete compatibility with dBASE III PLUS .ndx files.

Conditional Indexes

In dBASE IV, you can create an index tag containing entries only for records that satisfy a condition. Using a conditional index to limit database records can be faster than using a normal index to access the same records.

The effect of a conditional index is similar to filtering a database using the SET FILTER command. Refer to Chapter 5 of *Using dBASE IV*, or to Chapter 2 of *Language Reference*.

Set BROWSE Range Using an Index

An alternative way to very quickly display a range of records is the SET KEY command. This utilizes an active index to quickly set the top and bottom records of a BROWSE, without having to reindex to change the endpoints (as you would with a conditional index). Because it uses the powerful dBASE IV indexing capabilities, it is much faster than SET FILTER when dealing with a range of records, as opposed to individual records scattered throughout the database file.

See Chapter 3 of *Language Reference*.

Customization and Configuration

An expanded configuration file and new commands make system configuration easier and more flexible in dBASE IV.

DBSETUP is used to modify the existing hardware setup, transfer other dBASE IV files to your hard disk, and create or modify your Config.db file.

DBINFO provides a complete listing of your current hardware and software settings on the screen.

If you are using MS Windows, DBSETUP and DBINFO are iconized for you.

International Currencies and Dates

dBASE IV provides a number of functions to support international currency and date formats.

Number of Open Files Increased

The maximum number of simultaneously open files has been increased from 15 to 99 files. To use all of the additional files, the DOS Config.sys file must contain FILES=99.

Printer Drivers

dBASE IV now supports the use of optional installable printer drivers to control print functions. The printer drivers support such print attributes as bold, italic, underlined, superscript, subscript, and print font. In addition, you can control many of the internal print variables such as page number, line spacing, and page offsets with system memory variables.

Keyboard Macros

dBASE IV has its own complete keyboard macro capability. You can create and save keyboard macro files for playback at any time.

Multiple-Child Relations

A major improvement in dBASE IV is the added support of multiple-child, multiple-file relationships. A single file can now be related to more than one file simultaneously.

Multi-User Features

Automatic Multi-User File and Record Locks

dBASE IV now provides automatic file and record locking for a number of dBASE IV commands. Multiple users can simultaneously access and update different parts of the same database file without issuing explicit file and record locks. Thus, applications may now be used in single- and multi-user environments with a minimum of modification.

SET REPROCESS TO is used to specify the number of times a multi-user command is attempted before dBASE IV returns a network conflict error.

Automatic Refresh of Changed Data

When using BROWSE or EDIT, SET REFRESH TO specifies the number of seconds to wait between automatic checks of the database file in USE to see if another user has altered the data. dBASE IV will automatically display the changed data on your screen.

Structured Query Language (SQL) Capability

SQL is now an integral part of the dBASE command and programming language, allowing both dBASE and SQL commands to access data in .dbf files. Both modes share the common features of the dBASE IV environment, such as the compiler, the windowing system, and the debugger.

Mouse Support

dBASE IV now supports the use of a mouse on all design screens and in the Control Center. Also, the mouse is available during @...GET data input.

Programming Enhancements

Compiler

dBASE IV includes a built-in pseudo-compiler that greatly increases execution speed by interpreting and parsing command, procedure, and format files into intermediate code saved in object (.dbo) files. Subsequent execution of the program will be much faster.

SET DEVELOPMENT ON causes the compiler to compare .dbo and .prg file dates and times. If the .prg file has a more recent save date than its corresponding .dbo file, it is recompiled.

Data Validation

The @ <row>, <col> SAY...GET command has a new REQUIRED option that forces validation; otherwise, VALID only validates data when it is changed.

Debugger

A full-featured program debugger has been added to dBASE IV. It provides windows for debugger status, editing, and breakpoints with up to 10 conditions; it displays the program file and procedure name being executed, current line number, work area, database, and index file in use, current record number, and other information. The **Edit** window displays the program source file being executed. You can make and save changes to the file during debugging.

Procedures

When a source file is compiled into object code (a .dbo file), procedures encountered in the program file are added to a procedure list for that object file, and are then available to any subsequently called program file. Thus, all program files are now effectively procedure files.

The total number of procedures per procedure file has been increased from 32 to 963. A procedure can contain a maximum of 64K of generated object code.

dBASE IV still supports the SET PROCEDURE TO command for procedures that are contained in a file not activated by a DO <filename> statement, and also supports using more than one procedure with SET LIBRARY TO.

Template Language

The template language is a programming language that the developer can use to customize menu applications created with the Applications Generator, or the forms, labels, and reports produced by the design tools. The template language also provides the ability to automatically document applications created with these new tools.

User-Defined Functions

dBASE IV supports user-defined functions (UDFs) for flexibility and power in the command language.

User-Defined Menus

dBASE IV now supports user-definable menus in the programming language and Applications Generator, with which you can create and modify menus.

Windowing

dBASE IV fully supports up to 20 windows at once, either from the dot prompt or within the programming language

The rest of this section describes the new functions and improved commands of dBASE IV.

Work Areas

The maximum number of work areas has been increased to 40.

CALCULATE Command

CALCULATE can process and evaluate any of its functions in a single pass through the database, rather than having to make a separate pass for every function. This greatly speeds up processing time. There are also a number of new financial and mathematical functions in dBASE IV, including trigonometric and statistical functions.

Increased Command Line Length

dBASE IV supports command lines up to 1,024 characters long in program files. Command lines from the dot prompt are 254 characters.

Date Delimiters

dBASE IV can use curly braces ({}) to specify date literals.

Memo Format Change

Memo files in dBASE IV have been structurally changed from dBASE III PLUS. If you read an old memo field and change it using the dBASE IV internal editor, or if you read it without modification using an external editor, it will be resaved to the disk in dBASE IV format, which cannot be read by dBASE III PLUS.

If you want to run both dBASE IV and dBASE III PLUS using the same .dbt memo file, you can COPY the altered file back to the older format by using the DBMEMO3 option of the COPY command. See *Language Reference*.

Memory Variables and Arrays

dBASE IV has expanded support of memory variables and a full implementation of memory arrays. See *Language Reference* for complete information.

Arrays

dBASE IV supports one– or two–dimensional arrays of memory variables with the following commands and functions: APPEND FROM ARRAY, AVERAGE, COPY TO ARRAY, DECLARE, PUBLIC, REPLACE FROM ARRAY, STORE, and SUM.

Increased Memory Variable Limits

The maximum number of memory variables has been increased to 15,000.

SCAN/ENDSCAN Command

dBASE IV provides a SCAN/ENDSCAN programming construct to find and process database records meeting a specific condition. SCAN/ENDSCAN loops are easier to program and require fewer statements than equivalent DO WHILE/ENDDO loops.

UDFs and User Procedures

dBASE IV now supports UDFs and user procedures with a variable number of parameters.

Upgrading From dBASE IV Version 1.0

This section summarizes the enhancements in versions 1.1, 1.5 and 2.0, for those users who are upgrading directly from dBASE IV version 1.0.

Performance Improvements

dBASE IV version 2.0 uses new filter optimization and memory management techniques to improve dBASE's performance significantly.

Filter Optimization

New filter optimization techniques let dBASE IV access sets of records very efficiently. You can query a large database for a small set of records and expect dBASE to return the results quickly.

Virtual Memory

dBASE IV's new virtual memory manager automatically determines how to manage your system's extended memory and disk space for the best performance.

Data Buffer Management

dBASE IV includes a new data-buffering management system that optimizes file I/O and eliminates the need for external disk-caching software.

Performance Tuning Ability

To tune the performance of dBASE IV, you can set the TMP, DBTMP, and DBASEIV_BUFF operating system environment variables before starting dBASE IV. TMP and DBTMP identify a directory where dBASE IV creates temporary files. DBASEIV_BUFF modifies the way that dBASE IV allocates memory to data buffers.

New Menu Commands

New menu commands in dBASE IV version 2.0 let you define more actions for menu bars and pads. You can execute commands when users select (highlight), choose (activate), or exit menu bars and pads. You can also assign the same command to several or all menu bars or pads.

New Mouse Commands

dBASE IV version 2.0 adds new mouse commands and functions to give you more programming control over mouse actions.

Increased Array Sizes

With dBASE IV version 2.0, you can declare an array with as many elements as your system's virtual memory will support, as long as each dimension does not have more than 65,535 elements.

Ability to Create Conditional Indexes

In dBASE IV, you can create an index tag containing entries only for records that satisfy a condition. Using a conditional index to limit database records can be faster than using a normal index to access the same records.

The effect of a conditional index is similar to filtering a database using the SET FILTER command, but yields a dramatic performance improvement.

Refer to Chapter 5 of *Using dBASE IV*, or to Chapter 2 of *Language Reference*.

Access to Organize Menu from Browse and Edit

While viewing database records on a **Browse** or **Edit** screen in dBASE IV, you can use the options of the **Organize** menu to create a new index, change or drop an existing index, organize records using an index or sorted file, unmark records marked for deletion, or erase marked records. You can turn this option off using the NOORGANIZE option of BROWSE, EDIT, CHANGE, INSERT, or APPEND.

Refer to Chapter 5 of *Using dBASE IV*.

Ability to Size Query Columns

In dBASE IV version 1.0, each file skeleton column on the queries design screen had a fixed length of 21 characters, and only four columns could be displayed at once. In the current version of dBASE IV, each file skeleton column is as wide as its field name. As you type in data, column width expands to accommodate the data. You can adjust column width to display more than four columns at a time. Refer to *Using dBASE IV*.

Fewer Restrictions on Using UDFs and ON Commands

In dBASE IV version 1.0, you could not use user-defined functions (UDFs) and ON commands that interrupted the environment in which a command was executing. You also couldn't use a UDF that performed certain recursive actions (such as BROWSEing a calculated field that called a UDF that performed a BROWSE of the same database file).

In the current version of dBASE IV, you can remove most of these restrictions by including the DBTRAP = OFF command in your Config.db file or setting DBTRAP OFF at the dot prompt. If DBTRAP is ON, a UDF cannot:

- CLOSE, PACK, MODIFY STRUCTURE, USE, or ZAP a database file that is open in the current work area
- CLEAR, DEACTIVATE, DEFINE, MOVE, RELEASE, or RESTORE an active window
- Recursively use BROWSE, LIST, DISPLAY, EDIT, CHANGE, INSERT, or APPEND to address the same database file

When DBTRAP is OFF, these restrictions no longer fully apply. For complete information about using DBTRAP, refer to Chapter 1 of *Language Reference*.

Support for PostScript Printers

dBASE IV supports PostScript printers, such as the Apple LaserWriter, and compatibles. Refer to Chapter 2, "Configuring dBASE IV."

SQL Improvements

The following improvements have been made to the SQL capability in dBASE IV:

- Control Center Functions — You can use Control Center operations in SQL mode (for example, to browse, create queries, and run reports and labels using SQL tables).
- dBASE commands and functions — You can enter additional dBASE commands and functions on the SQL command line and in SQL program files.
- dBASE catalogs — A number of SQL commands, including CREATE TABLE and DROP TABLE, CREATE DATABASE and DROP DATABASE, SELECT...SAVE TO TEMP, and START DATABASE, add or remove information about SQL database files from the current dBASE catalog file if SET CATALOG is ON.
- User-defined functions — dBASE UDFs are now available in SQL program files.

These improvements are described in the sections that follow.

Control Center Functions

While in SQL mode, you can use menu system functions at the Control Center much as you can in dBASE mode.

You cannot use the menu system to create a SQL table or modify the structure of an existing table. These operations must be performed using SQL commands. However, you can use the menu system to create and modify dBASE database files that can later be defined as SQL tables.

Using the Control Center, you can also use table data to do the following:

- Create, modify, and print reports and labels
- Create and run SQL applications
- Create dBASE queries

dBASE Commands and Functions

In SQL mode, dBASE commands now provide an alternative to using SQL commands to define and manipulate data. You can use either set of commands, depending on your familiarity with each language and the task that you want to perform.

You can use most dBASE commands and functions in SQL mode to access non-SQL and SQL database files. This includes the following dBASE commands that perform the same tasks as SQL commands:

- Open a database file (USE)
- Position the record pointer (GOTO, SKIP)
- Display and update data (BROWSE, EDIT/CHANGE)
- Locate data (SEEK, DISPLAY)

- Add data (APPEND, without the BLANK option)
- Create a database file (CREATE/MODIFY STRUCTURE, used only for defining dBASE database files, not SQL tables)
- Create queries (CREATE/MODIFY QUERY)

Other dBASE commands and functions that you can use in SQL mode include the following:

- SET CATALOG and SET TITLE to record and maintain SQL objects in dBASE catalog (.cat) files
- SET INSTRUCT to control the display of prompts and SET WINDOW to edit memo fields of non-SQL database files
- ASSIST to access the Control Center and the menu system
- CREATE/MODIFY APPLICATION, CREATE/MODIFY LABEL, CREATE/MODIFY REPORT, CREATE/MODIFY SCREEN, REPORT FORM, and LABEL FORM to initiate Control Center design and print functions
- SET DESIGN to prevent an application user from accessing the dot prompt or one of the Control Center design screens
- SELECT to access SQL and non-SQL database files
- PROTECT to implement security for dBASE IV (use SQL GRANT and REVOKE commands to control access to SQL objects)
- SET to display and change the current values of SET commands
- ALIAS() to return the alias name of a specified work area
- SELECT() to return the number of the highest unused work area (cannot be used in a SQL command)

dBASE Catalogs

SET CATALOG is now supported in SQL mode. The following SQL commands affect the current dBASE catalog (.cat) file if SET CATALOG is ON:

- START DATABASE — adds entries for all of the tables in the database to the catalog. (SQL system catalogs are not added to the catalog.)
- DROP DATABASE — removes entries for all of the tables in the database from the catalog.
- CREATE TABLE — adds an entry for the new table to the catalog.
- DROP TABLE — removes the table entry from the catalog.
- SELECT...SAVE TO TEMP — adds an entry for the temporary table to the catalog. If you specify KEEP, the temporary table is not removed when the table is later discarded, but remains in the catalog as a non-SQL database file.

User-Defined Functions

You can use a dBASE user-defined function (UDF) in a SQL program file, with the following restrictions:

- A UDF can contain only the dBASE IV commands and functions that are available in SQL mode. A UDF cannot contain any SQL command.
- A UDF cannot be called from within a SQL command.

A UDF in a .prs file can call a UDF in a .prg file and vice versa, just as .prs and .prg files can call one another.

Summary of Commands and Functions Modified Since dBASE III PLUS

The following is a list of all commands, SET commands, and functions that have changed in the dBASE product since dBASE III PLUS. All the listed entries were added or changed in dBASE IV versions 1.0, 1.1, or 1.5.

NOTE
For information about new or changed commands and functions in version 2.0, read Chapter 5, "What's New in dBASE IV Version 2.0."

Commands

? and ?? now support AT, FUNCTION, PICTURE, and STYLE options to specify the format and style of displayed or printed output.

??? sends output to the printer bypassing the installed printer driver. ??? can be used to send printer control strings directly to the printer.

@...FILL TO...COLOR is used to change the display color attributes of a rectangular portion of the screen.

@...PICTURE functions that have been added are as follows: B, to left-align fields; H, to stretch fields horizontally (?/?? only); I, to center fields; J, to right-align fields; L, to display leading zeros; T, to trim spaces; V, to stretch fields vertically (?/?? only); $, to precede a numeric value with a currency symbol and separate thousands with commas; the semicolon (;), to expand embedded semicolons to line breaks in multi-row fields; and ^, to format an expression to appear or edit in exponential or scientific notation. The M function allows a multiple choice list for a single GET.

@...SAY...GET has been greatly enhanced to provide extended data validation, help, and custom prompts using the new clauses. VALID and RANGE check for validation when you move out of a field only when the REQUIRED keyword is present.

@...SCROLL shifts the contents of a specified region of the screen within the region itself.

@...TO has been enhanced so that you can select the characters used to draw boxes on the screen and specify the color of the box.

ACTIVATE MENU displays a previously DEFINEd horizontal bar menu and transfers control to that menu.

ACTIVATE POPUP displays a previously DEFINEd pop-up menu and transfers control to it.

ACTIVATE SCREEN directs screen output from a previously ACTIVATEd window to the global screen without removing the windows from the screen.

ACTIVATE WINDOW <window name list>/ALL displays and activates the windows contained in the name list or all windows. When a window is activated, all screen input and output takes place in that window.

APPEND includes improvements listed under BROWSE.

APPEND FROM ARRAY is a new command that appends records to a database from a named array. APPEND FROM also offers a REINDEX option that forces reindexing to wait until the command has finished APPENDing.

APPEND MEMO...FROM appends the entire contents of <filename> to the specified memo field in the current record. If the OVERWRITE option is specified, the memo field is erased before the file is read into the memo field.

ASSIST activates the dBASE IV Control Center.

AVERAGE replaces the values of one or more array elements in the named array with the computed averages of the specified elements.

BEGIN/END TRANSACTION provides for database changes to be made tentatively, then if there is a problem (a file is unavailable, for instance), ROLLBACK restores all files to the way they were when BEGIN TRANSACTION was issued.

BLANK can be used to fill fields and records with blanks and check for blank fields and records.

BROWSE has been much improved with an enhanced menu bar and the ability to edit memo fields. It can operate within a user-defined window, automatically pan fields, and display calculated fields. When using BROWSE in a multi-user environment, dBASE IV now performs a record lock only when you attempt to modify information contained in the record without first attempting a manual record lock. If another user changes information in a record that you are browsing, dBASE IV automatically displays the changed information on your screen. BROWSE now also provides a NOORGANIZE option to suppress the Organize menu.

CALCULATE processes records in the active database file for the <option list> functions. These are the financial function NPV() (net present value), and the statistical functions AVG() (arithmetic mean), CNT() (count records), MAX() (maximum), MIN() (minimum), STD() (standard deviation), SUM() (sum of values), and VAR() (variance).

CALL is now able to pass up to seven optional parameters to the CALLed routine. The .bin file size has been increased to 64K.

CLEAR MENUS releases from memory all horizontal bar menus not in use. CLEAR MENUS also clears all associated ON SELECTION and ON PAD commands.

CLEAR POPUPS releases all pop-up menus not in use from memory. CLEAR POPUPS also clears all associated ON SELECTION commands.

CLEAR SCREENS clears all screen memory variables.

CLEAR WINDOWS deactivates all windows, erases them from the screen, and releases them from memory. Any text obscured by an activated window is restored.

COMPILE searches for a program source (.prg) file and generates a program object (.dbo) file with the same filename. Program object (.dbo) files are pre-parsed, so execution speed is greatly increased with their use. If the RUNTIME option is specified, the compiler will print an error when it encounters a command that is not legal for RunTime. A number of compiler preprocessor directives now allow for conditional compilation. You can also now specify file extensions besides .prg. See Programming in dBASE IV.

CONVERT adds a field to a database structure that contains information required to do multi-user lock holder identification and change detection. Upon execution, CONVERT adds the _DBASELOCK field to the selected database. _DBASELOCK is used by the new CHANGE() and LKSYS() functions.

COPY INDEXES/TAG TO creates .mdx tags in <filename> or an .ndx file with the same key expression as each .ndx filename or .mdx tag specified.

COPY MEMO TO copies the named memo field to a new file, or adds the memo to an existing file if the ADDITIVE keyword is used.

COPY TAG TO creates a new .ndx file with the same key expression as the .mdx tag.

COPY TO ARRAY replaces the values of one or more array elements in the existing named array with fields from the database file currently in USE.

COPY TO now supports Framework II®, dBASE II®, RapidFile®, and Lotus 1-2-3 file types, and dBASE III PLUS memos. The WITH PRODUCTION keyword allows you to also copy the production .mdx.

COPY STRUCTURE EXTENDED produces a database file containing the new index option database field attribute in addition to field number, name, type, and size. The WITH PRODUCTION keyword allows you to also copy the production .mdx.

CREATE APPLICATION invokes the Applications Generator from the dot prompt.

CREATE FROM now creates a database structure containing the index database field attribute in addition to the field name, type, length, and decimal places (if applicable).

CREATE now accepts both N and F data types for binary coded decimal and floating point numbers, and also accepts the new index field database field attribute. See the Numeric Types section in Chapter 1 of *Language Reference* for more details.

CREATE VIEW FROM ENVIRONMENT now includes a query clause (?) to list all .qbe and .vue files on disk.

DBLINK is a stand-alone program capable of linking up to 256 related program object (.dbo) files into a single object file for faster execution.

DEACTIVATE MENU deactivates the active horizontal bar menu and erases it from the screen, but does not release it from memory.

DEACTIVATE POPUP deactivates the active pop-up menu and erases it from the screen, but does not release it from memory.

DEACTIVATE WINDOW <window name list>/ALL deactivates and clears the specified windows from the screen. Any text obscured by a window when the window was activated is restored. Any previously activated window is reactivated, or dBASE IV reverts to full-screen input and output. Deactivated windows are not cleared from memory, and may be reactivated at any time.

DEBUG invokes the new built-in debugger on the specified file.

DECLARE creates the arrays specified in the array list and sets aside a memory variable location for each element of each array. Arrays can be one– or two–dimensional.

DEFINE BAR is used to specify a single option on a pop-up menu.

DEFINE BOX establishes parameters for a graphic box to be drawn with subsequent lines of output.

DEFINE MENU is used to define a horizontal or vertical bar menu.

DEFINE PAD is used to define a single horizontal bar menu option.

DEFINE POPUP is used to easily create pop-up menus.

DEFINE WINDOW is used to specify the name, size, colors, and border pattern of a window, and its location on the screen. Up to 20 windows can be DEFINEd.

DELETE TAG removes the specified index tag from the specified .mdx files.

DEXPORT creates a binary named list (BNL) file from Control Center design objects, instead of the .npi file, for use with the DGEN() function. Use this command in place of the DTL_TRANSLATE environment variable, which has been removed.

DISPLAY FILES has been enhanced to support the TO PRINTER/TO FILE <filename> option.

DISPLAY has been enhanced to support the TO PRINTER/TO FILE <filename> option.

DISPLAY HISTORY now supports the TO PRINTER/TO FILE <filename> option.

DISPLAY MEMORY has been enhanced to show window names, pop-up menu names, horizontal bar menu names and pads, screens saved using SAVE SCREEN, and the memory consumed by each. This is in addition to memory variable names, data types, and contents, and the amount of available memory left. DISPLAY MEMORY now also supports the TO PRINTER/TO FILE <filename> option.

DISPLAY STATUS has been modified to display both .mdx and .ndx information and now supports the TO PRINTER/TO FILE <filename> option. It also displays some low-level file information.

DISPLAY STRUCTURE has been enhanced to display a database structure in a work area other than one currently selected, and now supports the TO PRINTER/TO FILE <filename> option.

DISPLAY USERS has been added to display the users currently logged onto the system.

DO supports an optional list of parameters to be passed to the called program or procedure file. DO can also pass a variable number of parameters to a precedure. See PCOUNT() later in this chapter.

EDIT supports all the features described under the BROWSE command. It works in a window if a format file is in use and the @...GETs are inside the window.

EJECT PAGE advances the printer to the first line of the next logical page or activates an ON PAGE routine.

END TRANSACTION terminates an active transaction. Please see BEGIN TRANSACTION for more information.

ENDPRINTJOB sends the ending printer codes (defined by the _pecodes system memvar) to the printer, performs an EJECT if required by the _peject system memvar, and loops back to the PRINTJOB statement for the number of copies required by the _pcopies system memvar. See PRINTJOB for more information.

ENDSCAN closes off an active SCAN loop. See SCAN for more information.

END TRANSACTION closes a transaction; see BEGIN/END TRANSACTION, earlier in this list.

EXPORT supports FW2 (Framework II), FW3 (Framework III), FW4 (Framework IV), RPD (RapidFile), DBASEII (dBASE II), and WKS (Lotus 1-2-3) file types in addition to the PFS file type.

FUNCTION is used to declare a user-defined function. User-defined functions must begin with the FUNCTION statement.

GO/GOTO has been enhanced to move the record pointer in a work area other than the one currently selected.

IMPORT now supports FW2 (Framework II), FW3 (Framework III), FW4 (Framework IV), RPD (RapidFile), DBASEII (dBASE II), and WKS and WK1 (Lotus 1-2-3) file types in addition to the PFS file type. IMPORT now tries to convert to the correct field types whenever possible instead of converting all fields to character type.

INDEX has been vastly enhanced to allow for the creation of .mdx files and conditional indexing.

KEYBOARD can be used to enter any non-negative value returned by the INKEY() function. You use KEYBOARD in a program to enter ASCII character values in the type-ahead buffer. dBASE IV then reads the values as if you had pressed the actual keys. KEYBOARD supports character strings, CHR() function, key label mnemonics, and numeric ASCII characters.

LABEL FORM now supports the TO FILE <filename> option which directs label output to the named text file.

LIST now supports TO FILE <filename> in every instance.

LIST MEMORY now supports all improvements listed under DISPLAY MEMORY.

LIST STATUS now also displays some low-level file information.

LIST STRUCTURE now supports all improvements listed under DISPLAY STRUCTURE.

LIST USERS has been added to list the users currently logged onto the system.

LOAD now supports .bin files of 64K.

LOGOUT closes a current PROTECT session.

MODIFY APPLICATION invokes the Applications Generator from the dot prompt.

MODIFY COMMAND also now supports the WINDOW <window name> option, allowing the editor to operate within a user-defined window area.

MODIFY COMMAND/FILE will now search for a program object (.dbo) file of the same name as a program source (.prg) file and delete the old .dbo file when a new version of the .prg file is saved. In this way, the .dbo files on disk correspond to the latest version of a .prg file.

MODIFY STRUCTURE has been enhanced so that, if a database structure is modified with index files open, the indexes are now checked for validation and maintained accordingly. MODIFY STRUCTURE also accepts the new database index field attribute.

MOVE WINDOW moves the named window to the new absolute position specified by <row>, <col>, or the new relative position specified by <delta row>, <delta column>. MOVE WINDOW also updates the window definition in memory to indicate the final coordinates.

ON KEY has been enhanced to trap specific keys. Function keys, shifted function keys, Alt- function keys, Ctrl- function keys, and all other special keys available on the keyboard can be trapped with ON KEY by using unique key labels. The new LABEL keyword prevents the <command> from being interpreted as the <key label name>.

ON PAD defines which pop-up menu will be activated when the selection bar is positioned to the prompt pad of the specified menu.

ON PAGE allows you to specify the action to be taken at a page break.

ON READERROR executes the command when invalid RANGE or VALID conditions occur, or when an invalid date is entered into a date variable.

ON SELECTION PAD indicates the action to be taken when a selection is made from a horizontal bar menu previously DEFINEd and ACTIVATEd.

ON SELECTION POPUP indicates the action to be taken when a selection is made from a pop-up menu previously DEFINEd and ACTIVATEd.

PLAY MACRO replays the designated keyboard macro.

PRINTJOB defines the beginning of a print job. PRINTJOB sends the starting printer codes (defined by the _pscodes system memvar) to the printer and performs an EJECT if required by the _peject system memvar.

PROTECT is no longer an external utility, and can be used from both the dot prompt and Control Center.

PUBLIC can now declare both memory variables and arrays to be PUBLIC.

QUIT WITH causes dBASE IV to return a value to the calling program (such as a batch file) when it terminates, allowing the calling program to check for successful operation.

REINDEX has been enhanced to support .mdx files and tags. Do not confuse with the new REINDEX option (see the Keywords section, later in this chapter).

RELEASE MENUS releases from memory the named horizontal bar menus and any associated ON PAD and ON SELECTION commands not in use.

RELEASE MODULES releases all modules when a specified list of items is omitted, rather than none.

RELEASE POPUPS releases from memory the named pop-up menus and any associated ON SELECTION commands not in use.

RELEASE SCREENS can be used to remove stored screen displays from memory, in addition to CLEAR ALL.

RELEASE WINDOWS deactivates the specified windows, restores any text obscured by the windows, and releases the window definition from memory. When a specified list of items is omitted, all windows are now RELEASEd, rather than none.

REPLACE FROM ARRAY performs the inverse of the COPY TO ARRAY command function: it updates one or more existing records in the active database file with values from an array.

REPLACE...WITH now automatically converts between character strings and memo fields. In addition, the ADDITIVE clause can now be used to build up memo fields from several character strings. REPLACE...WITH now includes a new REINDEX option to force reindexing to wait until the command finishes.

REPORT FORM now supports the TO FILE <filename> option which directs report output to the named text file.

RESET resets the database file header integrity flag in the specified work area to show that a transaction is no longer in progress.

RESTORE MACROS FROM retrieves the macros contained in the specified macro file.

RESTORE SCREEN moves a screen image saved with SAVE SCREEN from memory back to the screen display.

RESTORE WINDOW restores either ALL or the specified window definitions previously saved with the SAVE WINDOW command.

RETURN has been enhanced to return an expression when used in conjunction with a user-defined function.

ROLLBACK attempts to restore each database and index file involved in the current transaction to its state prior to the initiation of the last BEGIN TRANSACTION command. It closes all files created during a transaction. ROLLBACK uses either the current transaction log file, or a file you specify when you enter the command. Because of file format changes in version 1.5, ROLLBACK cannot use a transaction log file from a previous version of the dBASE product.

SAVE MACROS TO saves any defined macros to the specified macro file. If no file extension is given, the file is given the default .key extension.

SAVE SCREEN stores the current screen image in a named memory location. Use RESTORE SCREEN to replace the current screen with the saved image.

SAVE WINDOW saves either ALL or the named window definitions to the specified window file. If no file extension is given, the file is given the default .win extension.

SCAN...ENDSCAN establishes a loop to process database records meeting the specified condition. SCAN...ENDSCAN is similar to DO WHILE...ENDDO, but affords more flexibility in programming and offers cleaner syntax.

SHOW MENU displays a previously defined horizontal or vertical bar menu on the screen without activating it.

SHOW POPUP displays a previously defined pop-up menu on the screen without activating it.

SKIP has been enhanced to move the record pointer in a work area other than the one currently selected.

STORE now initializes array elements previously created.

SUM replaces the values of one or more memory variables or array elements in an existing array with the computed sums of the specified database elements.

TYPE has a new option to insert line numbers in a displayed or printed file. A page heading showing the filename, date, and page number is automatically included unless SET HEADING is OFF. TYPE also supports a TO FILE option.

UNLOCK has been enhanced to release record or file locks in a work area other than the one currently selected.

UPDATE includes a new REINDEX option to force reindexing to wait until the command finishes.

USE has been enhanced to support .mdx files and to open databases in a work area other than the one currently selected. USE has a new ORDER option to specify the master or controlling index when the database is opened, and a new NOUPDATE option, which opens the database for reading only.

USE NOLOG does not record the changes to the .dbf into the transaction log file.

USE NOSAVE opens the .dbf file for temporary use and deletes it from the disk when you close it.

USE AGAIN opens a second copy of the same .dbf in another work area.

SET Commands

NOTE
For commands that are SET either ON or OFF, the default setting is shown in uppercase letters.

SET ALTERNATE TO now supports an optional ADDITIVE keyword to extend existing text files with new output. Use of the ADDITIVE keyword appends output to the end of an existing ALTERNATE file instead of creating a new file.

SET AUTOSAVE ON/off saves each record to disk after any I/O operation.

SET BELL TO is used to modify the tone and duration of the bell.

SET BLOCKSIZE TO determines the block size of newly created index and memo files. <expN> may range from 1 to 32, and the actual block size is 512 bytes.

SET BORDER TO allows you to change the characters used to define the borders of windows, boxes, pop-ups, menus, and lines.

SET CARRY TO selects the fields to be copied forward when using APPEND, INSERT, and BROWSE. Using the ADDITIVE keyword adds the specified fields to those already specified in a previous SET CARRY TO command.

SET CLOCK on/OFF determines whether the system clock is displayed on the screen. The system clock is always displayed when using the Control Center, or the forms, labels, program, queries, or reports design screens.

SET CLOCK TO turns on the system clock and indicates where it is displayed.

SET COLOR OF is used to select the colors used in text, headings, boxes, and the like.

SET CURRENCY LEFT/right determines whether the currency symbol appears to the left or right of currency amounts.

SET CURRENCY TO changes the symbol used for the currency character. CURRENCY defaults to a dollar sign ($), but may be any string up to nine characters long.

SET CURSOR ON/off displays or hides the cursor.

SET DATE now supports a greatly expanded variety of date formats.

SET DBTRAP ON/off activates or deactivates a set of self-reference safety restrictions.

SET DESIGN ON/off is used by developers to prevent users from entering database, query, form, label, program, or report design modes. SET DESIGN is now local, so that if a sub-procedure turns it off (for example), then when the sub-procedure finishes, SET DESIGN is returned to its original value.

SET DEVELOPMENT ON/off controls whether dBASE IV will compare the creation date and time of an existing program object (.dbo) file with its related program source (.prg) file and recompile the .prg file if the object file is older than the source file.

SET DEVICE TO now supports the FILE <filename> option in addition to SCREEN and PRINTER to send the result of @ <row>, <col> SAY... commands to a text file.

SET DIRECTORY TO specifies the dBASE IV default drive and directory.

SET DISPLAY TO specifies the type of video display monitor being used.

SET DOHISTORY on/OFF has been retained for compatibility with dBASE III PLUS, but performs no function in dBASE IV. SET DOHISTORY is no longer necessary thanks to the enhanced debugger.

SET EXCLUSIVE on/off defaults to ON for the single-user version of dBASE IV; the default for the multi-user version is OFF.

SET FIELDS TO now can use calculated fields and designate a field as read-only. SET FIELDS TO also now supports the use of a wildcard skeleton to select which fields are active.

SET FIXED on/OFF has been retained for compatibility with dBASE III PLUS, but performs no function in dBASE IV. Since dBASE IV has floating point numeric support, the SET FIXED command is unnecessary. Use the SET DECIMALS command to determine the number of decimal places displayed.

SET FULLPATH on/OFF determines whether the result of a function that returns a filename will include the full path name in addition to the drive letter and filename, for compatibility with dBASE III PLUS.

SET FUNCTION now provides programmable Ctrl- and Shift- function keys in addition to the standard function keys.

SET HOURS TO determines whether a 12-hour or 24-hour system clock format is used.

SET INDEX TO has been enhanced to support .mdx files. It also supports an ORDER option to designate which index file or tag will be the master or controlling order.

SET INSTRUCT ON/off displays or hides a prompt box of actions at the Control Center when a file is selected. If SET INSTRUCT is OFF and SET TALK is ON, dBASE IV code is displayed on-screen as it is generated in REPORT FORM and LABEL FORM.

SET KEY TO uses the controlling index to establish a sequential range of records, without having to either examine each record with a filter or reindex with a conditional index whenever the endpoints change.

SET LIBRARY establishes a library file of procedures to be used throughout a program or application, thus reserving SET PROCEDURE for local operations.

SET LOCK ON/off controls whether dBASE IV performs automatic file and record locking before processing read-only commands (AVERAGE, CALCULATE, COPY [STRUCTURE], COUNT, JOIN, INDEX, LABEL, REPORT, SORT, SUM, and TOTAL) in a multi-user environment.

SET MARK TO changes the delimiter used to separate the month, day, and year when dates are displayed or entered. The default character depends on the setting of the SET DATE command; for example, SET MARK TO establishes the slash character (/) as the delimiter to be used if the SET DATE TO USA command is in effect.

SET MENUS ON/off has been retained for compatibility with dBASE III PLUS, but performs no function in dBASE IV. The enhanced Help system in dBASE IV replaces SET MENUS in dBASE III PLUS.

SET MESSAGE contains a new clause, AT <expN>. If SET STATUS is ON, the command displays the user-defined message at the bottom of the screen; if SET STATUS is OFF, the command displays the message at the screen position specified by the AT coordinates.

SET NEAR on/OFF determines the position of the record pointer after an unsuccessful SEEK or FIND command. Following an unsuccessful SEEK or FIND with SET NEAR OFF, dBASE IV, like dBASE III PLUS, positions the record pointer to the end-of-file and sets the EOF() flag to a logical true (.T.). SET NEAR ON, however, positions the pointer to the first record following the computed position of the desired search key in the controlling index.

SET ORDER TO has been enhanced to support .mdx files as well as .ndx files. When an .mdx file is open in the current work area, you must use tag names instead of numeric expressions. SET ORDER TO...NOSAVE causes indexes created as a result of a query not to be saved when the .mdx file is closed.

SET PAUSE on/OFF controls the display of the SQL SELECT command and stops the display of data after each full screen.

SET POINT TO changes the symbol *displayed* as the decimal point character (internally, dBASE IV always uses a period).

SET PRECISION TO determines the number of significant digits, including the decimal point and sign, of type N numbers in dBASE IV. The precision may be set between 10 and 20. The default precision is 16. A different default precision may be specified in the Config.db file. SET PRECISION has no effect on type F numbers in dBASE IV, which always use 16 significant digits.

SET PRINTER TO affords control over the destination of printed output and network spoolers. New features include TO <computer>, TO <spooler>, and TO FILE.

SET REFRESH TO works in conjunction with BROWSE and EDIT to set the number of seconds between automatic checks of the database file in USE for changes made by another user. If another user has changed the data, dBASE IV automatically displays the changed data on your screen.

SET RELATION TO now supports multiple-child relations with a single command.

SET REPROCESS TO is used to select how many times a multi-user process should be attempted before dBASE IV returns a network-related error. The maximum number of retries for SET REPROCESS is 32,000.

SET SEPARATOR TO changes the symbol displayed as the numeric separator character (between thousands). The default numeric separator is a comma.

SET SKIP TO changes the way the record pointer is positioned when processing related database files. When used with SET RELATION, SET SKIP causes the record pointer in the specified database files to be advanced before the record pointer in the primary database file is advanced. This provides access to multiple occurrences of the same key in linked files.

SET SPACE ON/off controls whether a space is generated between fields separated by a comma in the ? [<expression list>] and ?? [<expression list>] commands.

SET SQL on/OFF controls the new interactive SQL mode from the dot prompt.

SET TRAP on/OFF controls whether the debugger is invoked when an error occurs or the **Esc** key is pressed during program execution and no ON ESCAPE statement is active. SET TRAP ON to invoke the debugger automatically when one of these conditions occurs.

SET VIEW TO now works in concert with CATALOG. When a dBASE III PLUS .vue file or dBASE IV .qbe file is updated in dBASE IV, the resulting file is saved as a query object (.qbo) file. The updated .qbo files cannot be used with dBASE III PLUS.

SET WINDOW OF MEMO TO sets the default window for editing memo fields to <window name> when using the BROWSE, EDIT, APPEND, CHANGE, READ, and INSERT commands.

Functions

ACCESS() returns a user's access level (1 to 8) when dBASE IV was started using the PROTECT log-in security features.

ACOS() returns the smallest positive angle size in radians whose cosine matches the specified value.

ALIAS() returns the alias name of the specified work area as an uppercase character string. There are 40 work areas available in dBASE IV.

ASIN() returns the smallest positive angle size in radians whose sine matches the specified value.

AT() can now take a memo field name as its second argument.

ATAN() returns the smallest positive angle size in radians whose tangent matches the specified value.

ATN2() returns the smallest positive angle size in radians whose sine and cosine match the two specified values.

BAR() returns the bar number of the most recently selected pop-up menu option.

BOF() can now be passed an alias.

CALL() executes one of the up to 16 binary modules previously loaded into dBASE IV through the LOAD command. CALL() returns a character expression with the same length as the input character expression. The value of the expression, however, can be modified by the CALLed module. CALL() is able to pass up to seven optional parameters to the CALLed routine.

CATALOG() returns the name of the current active catalog file.

CEILING() returns the smallest integer greater than or equal to the specified numeric expression.

CERROR() reports the number of the last compile-time error. You can use CERROR() for error checking in programs that generate and compile dBASE IV code.

CHANGE() tests whether the data in the current record has changed since it was read from disk when dBASE IV is running on a network. CHANGE() returns a logical true (.T.) if the count or checksum value of the current record is different than the value stored in the _DBASELOCK field created by the CONVERT TO command; CHANGE() returns a logical false (.F.) if the record has not been changed since first read from disk. The optional <alias> argument can be used to test a record in another work area. See also the CONVERT command and LKSYS() function.

COMPLETED() returns a logical true (.T.) by default and when a transaction is successfully ended. COMPLETED() returns a logical false (.F.) from the time a BEGIN TRANSACTION command is issued until an END TRANSACTION command is executed.

COS() returns the cosine of the specified angle measured in radians.

DBF() can now be passed an alias.

DELETED() indicates whether the current record is marked for deletion.

DESCENDING() returns .T. if an .mdx index tag was created with the DESCENDING keyword.

DGEN() generates a dBASE IV program out of a design BNL file, using the specified .gen file. DGEN() is now a dBASE language function.

DIFFERENCE() returns a number from zero to four indicating how similar two literal strings are to each other using the SOUNDEX code of each string. Two closely matched strings yield a difference of four; two strings with no letters or spaces in common return a difference of zero.

DMY() returns a character string representation of the date variable argument in DD Month YY format, where Month is the full month name. DMY(<expD>) returns the century and year in the form YYYY if SET CENTURY is ON. For example, DMY({11/20/58}) returns 20 November 1958 with SET CENTURY ON, and 20 November 58 with SET CENTURY OFF.

DTOR() converts the specified number of degrees to radians.

DTOS(<expD>) converts a date expression into an eight-character string suitable for use in an index without further manipulation. For example, DTOS({11/20/58}) will return the string 19581120. The century is always returned.

EOF() can now be passed an alias.

FCLOSE() closes a low-level file after FOPEN() or FCREATE().

FCREATE() creates a new low-level file in EXCLUSIVE mode.

FDATE() returns the specified file's date of last update.

FEOF() returns .T. if the file pointer is at end-of-file.

FERROR() returns the I/O error status of a low-level file I/O operation.

FFLUSH() flushes the system buffer for a particular low-level file to disk, thereby updating the file allocation table (FAT).

FGETS() reads a string of characters terminated by end-of-line from a low-level file and returns the character string.

FIELD() can now be passed an alias.

FIXED() converts a data type F floating point expression into a data type N binary coded decimal number.

FLDCOUNT() returns the number of fields in an open database file. FLDCOUNT() can be passed an alias.

FLOAT() converts a data type N binary coded decimal expression into a data type F floating point number.

FLOOR() returns the largest integer that is less than or equal to the specified numeric expression.

FOPEN() opens a low-level file in EXCLUSIVE mode.

FOR() returns the FOR conditions used to make an .mdx index tag.

FOUND() can now be passed an alias.

FPUTS() writes a character string to a low-level file.

FREAD() reads a specified number of bytes from a low-level file into a character string.

FSEEK() positions the file pointer in a low-level file.

FSIZE() returns the specified file's size.

FTIME() returns the specified file's time of last update.

FV() returns the future value for the specified payment, rate, and number of periods.

FWRITE() writes characters from a character string to a low-level file.

HOME() returns the path of the copy of dBASE IV which is currently running.

ID() returns the name of the current user as a character string.

IIF() is an "immediate IF," a shortcut function that takes the place of IF <condition> <expression 1> ELSE <expression 2> ENDIF.

INKEY([<N>]) traps a keypress or the first character in the typeahead buffer and returns an integer that represents this key. If <n> is specified, dBASE IV will pause <n> number of seconds before continuing program execution. <n> can be a value with decimal points.

ISBLANK() returns .T. if the field, memvar, array element, or other expression given as the argument is blank.

ISMARKED() returns a logical true (.T.) if any transaction is in progress that may affect the integrity of the specified database or alias, or a transaction has failed but no ROLLBACK has been executed. A logical false (.F.) is returned if no transaction is in progress and the integrity of the database file header is intact.

KEY() returns the key expression of an .ndx file or .mdx tag as a character string.

LASTKEY() returns the decimal ASCII value of the last key pressed that exited a full-screen operation such as BROWSE or EDIT.

LEFT() is now able to take a memo field name or character expression as its first argument.

LEN() is now able to take a memo field name or character expression as its argument.

LIKE() returns a logical true (.T.) or logical false (.F.) depending on whether the <pattern> compares with the <expC> argument. You have unlimited use of the * and ? wildcard symbols as part of the pattern string.

LINENO() returns the file relative line number of the current command.

LKSYS() accepts numeric arguments that range from 0 to 5, and returns either the time of lock, date of lock, or ID of the user who last locked a record or file when running in multi-user mode. LKSYS() requires the _DBASELOCK field created by the CONVERT command to operate.

LOCK() has been enhanced to support locking of multiple records with a single command line. The maximum number of simultaneous locks has been increased to 100. Unlike in dBASE III PLUS, LOCK() is now additive; it will not unlock a previously requested lock. LOCK() also now allows a locked file or record to be read while locked.

LOG10() returns the base 10 logarithm of the specified numeric expression.

LOOKUP() searches the database containing the <look-in field> for the first record where the <look-in exp> matches the <look-for field>. If the search or lookup is successful, the <return field> is returned.

LUPDATE() can now be passed an alias.

MAX() now accepts date and character expressions in addition to numeric.

MDX() returns the filename of the .mdx file containing the tag number.

MDY() returns a character string representation of the specified date variable in Month DD, YY format, where Month is the full month name. MDY(<expD>) returns the century and year in the form YYYY if SET CENTURY is ON. For example, MDY({11/20/58}) returns November 20, 1958 with SET CENTURY ON, and November 20, 58 with SET CENTURY OFF.

MEMLINES() returns the number of lines a memo field will occupy based on the current value of the SET MEMOWIDTH TO command.

MEMORY() returns the number of kilobytes of unused RAM memory available to dBASE IV.

MENU() returns the name of the active or most recently ACTIVATEd horizontal or vertical bar menu.

MIN() now accepts date and character expressions in addition to numeric.

MLINE() returns the text of the specified line in the memo field, based upon the current value of the SET MEMOWIDTH TO command.

NDX() has been enhanced to return the names of active index files in the specified work area.

NETWORK() returns a logical true (.T.) if dBASE IV has been installed for network operation and is running on a network, and a logical false (.F.) if not.

ORDER() returns the name of the .ndx file or .mdx tag controlling the order of a database file in the specified work area.

PAD() returns the name of the most recently selected horizontal or vertical bar menu option.

PAYMENT() returns the payment required to amortize a loan with the specified principal balance, interest rate, and number of periods.

PCOUNT() returns the number of parameters passed to a procedure.

PI() returns the floating point number 3.14159265358979. The value of PI() when displayed or printed is specified by the value of the SET DECIMALS TO command.

POPUP() returns the name of the active or most recently ACTIVATEd pop-up menu.

PRINTSTATUS() returns a logical true (.T.) if the printer is ready to accept output, and a logical false (.F.) if the printer is not ready.

PROGRAM() returns the name of the program, procedure, or UDF being executed.

PROMPT() returns the PROMPT string of the most recently selected horizontal or vertical bar or pop-up menu option.

PV() calculates and returns the present value of the specified payment, interest rate, and number of periods.

RAND() returns a random number between 0 and 1. RAND() can use a given numeric expression for the seed value; if the given numeric expression is negative, RAND() will generate a seed based on the system clock.

RECCOUNT() can now be passed an alias.

RECNO() can now be passed an alias.

RECSIZE() can now be passed an alias.

RIGHT() is now able to take a memo field name or character expression as its first argument.

RLOCK() has been enhanced to support locking of multiple records with a single command line. The maximum number of simultaneous locks has been increased to 50. Unlike in dBASE III PLUS, RLOCK() is now additive; it will not unlock a previously requested lock. RLOCK() also now allows a locked file or record to be read while locked.

ROLLBACK() returns a logical true (.T.) by default or when a ROLLBACK command completes successfully. When a ROLLBACK command is issued, ROLLBACK() returns a logical false (.F.) until the ROLLBACK terminates successfully.

RTOD() converts the specified number of radians to degrees.

RUN() can roll dBASE IV out of memory, leaving only a 10K kernal, so that you can run another program, then return to dBASE IV. RUN() can now return the called command's error code, rather than an error code of the operating system shell.

SEEK() is similar to the SEEK <exp> command, but returns a logical value immediately and allows the use of an alias name as a second argument. SEEK([,<alias>]) returns a logical true (.T.) if the expression sought is found in the master index of a database file, or a logical false (.F.) if not.

SELECT() with no argument returns the number of the lowest unused work area available. SELECT() with a work area alias as an argument now returns the work area number associated with the specified alias.

SET() tests the status of ON/OFF and integer SET parameters. Additional arguments can now be passed to SET(): "ATTRIBUTE", "BORDER", "DATE", "DEVICE", "INDEX", "LIBRARY", "MARK", "ORDER", "POINT", "SEPARATOR", "SKIP", and "WINDOW".

SIGN() returns 1 if the numeric expression is positive, 0 if it is equal to zero, and –1 if the expression is negative.

SIN() returns the sine of the specified angle measured in radians.

SOUNDEX() returns a phonetic character string code based on what the specified character string sounds like, which can be used in uncertain match lookups. SOUNDEX() returns the code of the specified character string in letter, digit, digit, digit format.

SUBSTR() can now operate on a memo field.

TAG() returns the tag name of the specified MDX file as an uppercase character string.

TAGNO() returns the index number for a named index.

TAN() returns the tangent of the specified angle measured in radians.

UNIQUE() returns .T. if an index was created using UNIQUE or with SET UNIQUE ON.

USER() returns the name of the user (from the Dbsystem.db file) logged into a PROTECTed system.

VARREAD() returns the name of the field or memory variable currently being edited as an uppercase character string.

WINDOW() returns the name of the currently active window.

Keywords

NOTE
Keywords used with the SET() function must be enclosed in double quotation marks.

"ATTRIBUTE" is used with SET(), and returns the dBASE IV color setting.

"BORDER" is used with SET(), and returns the current SET BORDER value ("SINGLE", "DOUBLE", "PANEL", "NONE", or a border expression).

"DATE" is used with SET(), and returns the style used for the date.

"DEVICE" is used with SET(), and returns "SCREEN", "PRINT", or "FILE <filename>".

"INDEX" is used with SET(), and returns the name of the active index.

"LIBRARY" is used with SET(), and returns the name of the active library file (see SET LIBRARY).

"MARK" is used with SET(), and returns the date delimiter.

NOLOG is used with USE, and causes changes to the file not to be recorded in the transaction log file, and the file to not be restored during a ROLLBACK.

NOORGANIZE is used with BROWSE, EDIT, CHANGE, INSERT, and APPEND, and disables the Organize menu.

NOSAVE is used with SET ORDER TO, and causes indexes created as a result of a query not to be saved when the .mdx file is closed. It is also used with USE, and causes the file to be automatically deleted when it is closed.

"ORDER" is used with SET(), and returns the name of the active index tag (or the .ndx name).

"POINT" is used with SET(), and returns the character used for the decimal point.

REINDEX is used with APPEND FROM, BLANK, REPLACE, and UPDATE, and causes these commands to rebuild the active index only after all records have been changed.

SCROLL is used with @, and causes the contents of a specified region to be shifted within the region itself.

"SEPARATOR" is used with SET(), and returns the character used as a numeric separator (usually "," or ".").

"SKIP" is used with SET(), and returns the aliases set by SET SKIP.

"WINDOW" is used with SET(), and returns the name of the memo window.

WITH is used with QUIT, and causes dBASE IV to return a value when it terminates.

Miscellaneous Changes

ARGUMENT() in Template Language returns the character string (if any) passed to DGEN() as the second (parameter) argument.

BUILD utility is no longer included with dBASE IV.

Compiler directives have been added to allow for conditional compilation. See Programming in dBASE IV for complete information.

Config.db commands have been added. See Chapter 2 of this manual for complete information about all new Config.db commands except the entry and exit program definitions, which are described in Programming in dBASE IV.

Start-up parameters can now be used with dBASE IV. A -c configuration switch tells dBASE IV to use the specified configuration file in place of Config.db, and parameters after a program name can now be passed to the program when it runs. See Chapter 1.

TOKEN() in Template Language is used by dBASE IV template programmers to easily parse the ARGUMENT() string.

CATALOG is no longer strictly in area 10, but requires the CATALOG() function for editing.

Two new start-up parameters allow you to pass programs and parameters from the operating system prompt, and to specify a different configuration file instead of Config.db.

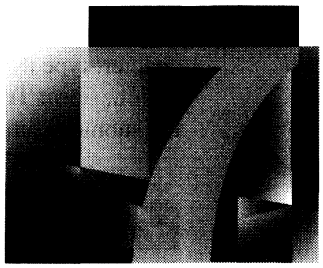
For dBASE IV on network, the CONTROLPATH= option in your configuration file allows you to specify the location of the user-count file, Dbase412.acc.

SYSPROC= allows you to set up an overriding set of procedures for use in dBASE IV programs.

dBASE IV open architecture allows support for developers who want to customize the design surfaces. See Chapter 17 of Programming in dBASE IV.

Exploring dBASE IV

Database Basics



Information is an important resource. From sports statistics to the latest stock market prices, facts put people in charge of the myriad details that affect their lives. dBASE IV puts you in charge of your facts.

dBASE IV is a *database management system* (often abbreviated DBMS), a powerful tool for managing data. dBASE IV lets you store, relate, manipulate, and retrieve vast amounts of information with speed and efficiency.

You can use dBASE IV in two ways:

- Control Center — the gateway to a system of menus that guides you through data management tasks, from maintaining files to printing reports.
- Dot Prompt — the dBASE command prompt that offers you the speed and flexibility of entering dBASE commands directly.

Depending on your needs, each method has unique advantages. However, each offers the full spectrum of dBASE IV capabilities, and you can easily move from one to the other.

This chapter introduces you to managing data with dBASE IV. You will learn:

- What a database is
- How the dBASE IV database management system works
- How to display the Control Center and use its menus and panels
- How to create a database file
- How to display, edit, add to, reorganize, filter, and print the data in a database file

While doing the exercises in this chapter and those that follow, you can refer to *Using dBASE IV* for greater detail about specific tasks.

Preparing for This Chapter

To perform the exercises in this chapter, you must have installed the sample files provided with dBASE IV, as described in Chapter 1 of this manual. Before or after dBASE IV installation, the sample files can be installed in their own directory (normally C:\DBASE\SAMPLES) on your computer's hard disk.

If you did not copy the sample files during installation, copy them before using this chapter: type INSTALL at the operating system prompt, choose the **Menu-driven** option, select **Transfer other files** from the **Install** menu, select **Sample files** from the submenu, and follow the prompts.

NOTE

- *All exercises in this and the remaining chapters assume that you are using the default settings provided with dBASE IV. If you are using a modified version of the program, some results or screen displays may not match the description in the chapter. The exercises also assume that you start dBASE IV when you begin a new chapter and quit dBASE IV at chapter end.*
- *Having a printer connected to your computer is not necessary for performing most of the exercises in these chapters, except those that deal with printing. If you have a printer, make sure that you have installed it for dBASE IV using the DBSETUP program described in Chapter 2 of this manual, and that the printer is ready to print.*
- *As you do the exercises in these chapters, don't worry about changing the data in the sample files; if necessary, you can use the Menu-driven option of the INSTALL program described in Chapter 1 of this manual to transfer the sample files again.*

What is a Database?

A database is a collection of information that is organized for ease of reference. For example, a telephone book is a database, as is a train schedule, a real estate listing, or an inventory.

Before electronic data processing, databases had to be maintained on paper stored in file folders that were kept in filing cabinets. Files had to be pulled out of cabinets, forms filled, and paper shuffled.

With dBASE IV, you can do all this electronically. Information can be filed and recalled instantly and accurately.

How Database Management Systems Work

Computers are unrivaled in their capacity to store data. A single floppy disk, for example, can hold 20 to 40 pages of the information from a typical telephone book. A hard disk can store more than a dozen full-length novels. But just as important as storage volume is the ease with which electronic data can be organized and manipulated.

Records and Fields

An efficient database system organizes information by category. Within each category, information is arranged in a certain sequence. For example, customer information for a business might be arranged alphabetically by customer last name or numerically by customer number.

The items of information maintained for each customer would be the same: name, address, telephone number, and so on. In database terminology, all information for a given customer is called a *record*. Each individual item of customer information is called a *field*.

Database Files

In dBASE IV, you use a *database file* to store and maintain records for a category of information. The fields in a database file define its *structure*.

In the database file illustrated in Figure 7-1, each record consists of Lastname, Firstname, Address, City, State, Zip, and Phone fields. In the record for Rick Lisbonn, for example, the Address field contains *1550 Keystone St.* and the Lastname field contains *Lisbonn*.

LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	PHONE
Lisbonn	Rick	1550 Keystone St.	Glendale	CA	91206	(818) 555-3344
Garnett	Lena	520 S. 8th St.	Reno	NV	89504	(702) 555-9121
Kaufman	Lisa	1960 Lindley Ave.	Reseda	CA	91355	(818) 555-0300
Johnson	Jay	14234 Riverside Dr.	Riverside	CA	92504	(714) 555-6784
Collins	Sara	303 W. Milford St.	Portland	OR	97219	(503) 555-0953
Arlich	Kim	10564 Ventura Blvd.	Long Beach	CA	90806	(213) 555-8773
Montovan	John	1034 Lorraine St.	Culver City	CA	90230	(213) 555-3403
Goreman	Vicky	203 E. 3rd St.	Mesa	AZ	85201	(602) 555-8801
Plimpton	Daniel	5934 Ocean Blvd.	Santa Monica	CA	90406	(213) 555-7150
Youngblood	Dick	1190 Fulton Pl.	Laguna	CA	90344	(714) 555-9108

Figure 7-1 Name and address file

A record can also contain blank fields. For example, this entry is also a valid record:

Garnett Lena Reno (702) 555-9121

A field has a *data type*, which is dictated by the type of data that it stores. In Figure 7-1, all fields are of the *character* data type. Data types are:

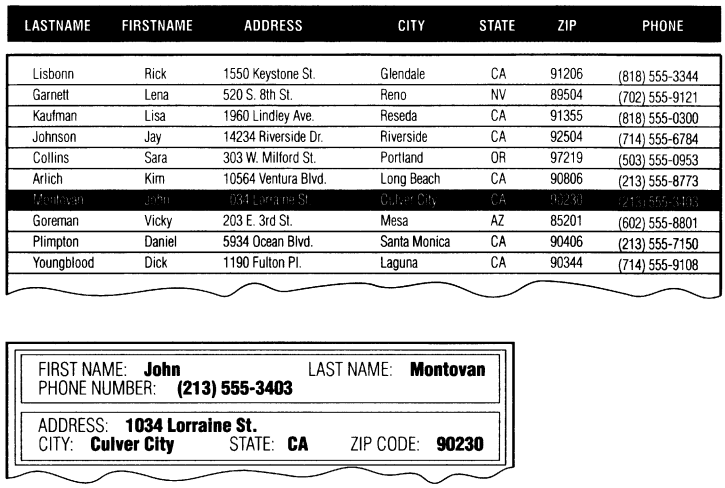
- Character type — letters, symbols, and numbers that are not used for calculation
- Numeric type — numbers that are used for calculation (type N: binary coded decimal remembers; type F: floating point binary numbers)

- Date type — date information
- Logical type — a flag (Y=yes, N=no; or T=true, F=false) indicating whether a condition is true or false
- Memo type — variable-length character information

For complete information about data types, refer to Chapter 1 of *Language Reference*.

Two Ways to Show Data

You can display information in a dBASE database file as a *table* or as a *form*, as illustrated in Figure 7-2.



LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	PHONE
Lisbonn	Rick	1550 Keystone St.	Glendale	CA	91206	(818) 555-3344
Garnett	Lena	520 S. 8th St.	Reno	NV	89504	(702) 555-9121
Kaufman	Lisa	1960 Lindley Ave.	Reseda	CA	91355	(818) 555-0300
Johnson	Jay	14234 Riverside Dr.	Riverside	CA	92504	(714) 555-6784
Collins	Sara	303 W. Milford St.	Portland	OR	97219	(503) 555-0953
Arlich	Kim	10564 Ventura Blvd.	Long Beach	CA	90806	(213) 555-8773
Montovan	John	1034 Lorraine St.	Culver City	CA	90230	(213) 555-3403
Goreman	Vicky	203 E. 3rd St.	Mesa	AZ	85201	(602) 555-8801
Plimpton	Daniel	5934 Ocean Blvd.	Santa Monica	CA	90406	(213) 555-7150
Youngblood	Dick	1190 Fulton Pl.	Laguna	CA	90344	(714) 555-9108

FIRST NAME: **John** LAST NAME: **Montovan**
PHONE NUMBER: **(213) 555-3403**

ADDRESS: **1034 Lorraine St.**
CITY: **Culver City** STATE: **CA** ZIP CODE: **90230**

Figure 7-2 Table (top) and form (bottom)

The distinction between a table and a form is illustrated by your checkbook, which is a database containing information about the checks you've written. When you're writing a check, you're filling in information on a form. When you're recording the check in the check register, you're entering a record in a table.

Displaying information in a table allows you to scan a number of records at once. Displaying information as a form lets you concentrate on one record at a time. In dBASE IV, you can switch back and forth from viewing data in a table to viewing it as a form.

Displaying the Control Center

The Control Center is a graphic display from which you select files and initiate database operations. As its name implies, the Control Center is the gateway to the dBASE IV database management system.

To start dBASE IV, type dbsample at the operating system prompt (which looks something like **C:\>**), and press ↵. The Control Center, shown in Figure 7-3, appears.

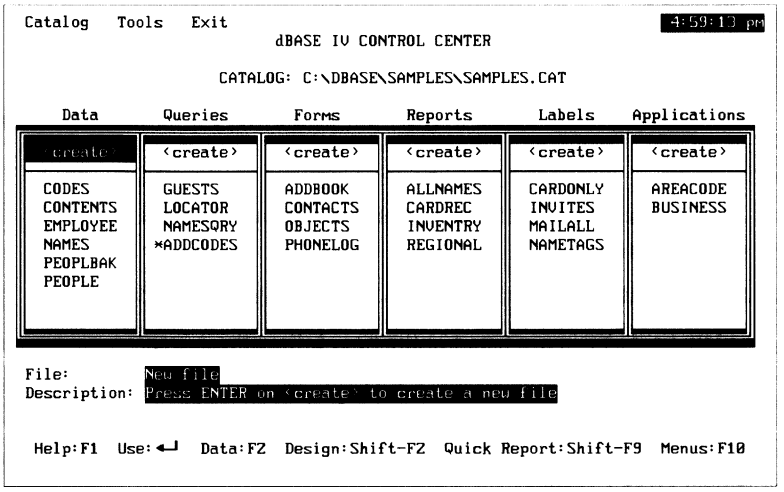


Figure 7-3 Control Center

The names of the sample files are displayed in the Control Center panels, and the catalog for the sample files (for example, C:\DBASE\SAMPLES\SAMPLES.CAT) appears in the center of the screen opposite **CATALOG:**.

NOTE

A catalog is a listing of information about each file needed for a particular job or application (for example, filename and operating system directory location). The Samples catalog lists the sample files that you will be using for the exercises in this and the following chapters. Once the Samples catalog is open and its name and path information are displayed in the Control Center, information about any file that you create is added to the catalog. Catalogs are discussed in Chapter 3 of Using dBASE IV.

If something other than the Control Center appears, do the following:

- If the dot prompt (a period character) and a cursor (a blinking underline character) appear at the lower left side of your screen, press **F2** or type assist and press ↵ to display the Control Center.
- If neither the dot prompt nor the Control Center appears, reinstall dBASE IV as described in Chapter 1 of this manual.

If, when displayed, the Control Center panels do not contain the names of the sample files and the catalog displayed is not C:\DBASE\SAMPLES, follow these steps:

1. Press **F10 Menus**.

   The **Catalog** menu appears with the **Use a different catalog** option highlighted.

2. Press ↵.

   A list of the catalogs in the current directory appears in a box on the right side of the screen.

3. If **SAMPLES.CAT** is listed, move the highlight to it and press ↵.

If **SAMPLES.CAT** is not listed in the box, follow these steps:

1. Press **Esc** once to remove the box display.
2. Press the left or right arrow key until the **Tools** menu is highlighted.
3. Select **DOS Utilities**.
4. Press **F10 Menus** (the **DOS** menu appears).
5. Select **Set default drive:directory**.
6. At the prompt, type path information for the directory that contains the sample files (for example, C:\DBASE\SAMPLES) and press ↵.
7. Press **F10 Menus**, select **Exit to Control Center** and press ↵.

How Control Center Panels are Used

In addition to database files, dBASE IV uses other types of files to help you manage data. Each Control Center panel contains the names of a different type of file. A file of a given type is created using the **<create>** marker that appears at the top of the panel.

The panels, the types of files they contain, and the file extensions are described in Table 7-1.

Table 7-1 Control Center panels

Panel	File	Purpose
Data	Database (.dbf)	Create a new database file for storing data or modify the structure of an existing database file; display and edit file data.
Queries	Query (.qbe)	Create a new query file to display selected records and fields of a database file; modify an existing query file.
Forms	Form (.scr)	Create a new form for entering or editing data in a database file one record at a time; modify an existing form file.
Reports	Report (.frm)	Create a new report format for displaying and printing the data in a database file; modify the format of an existing report file; use a format to display and print data.
Labels	Label (.lbl)	Create a new label format for printing labels using the data in a database file; modify the format of an existing label file; use a format to display and print labels.
Applications	Design (.app) Program (.prg)	Create application design and program files for performing database management tasks; modify and run an existing program.

This chapter shows you how to use the **Data** and **Queries** panels to create and work with database and query files. You'll learn more about the other panels and their files in Chapter 9.

Creating a Database File

Now you'll use the Control Center to create a simple database file for an address book:

NOTE
If you make a mistake while typing, just use the **Backspace** *key to erase your mistake and continue typing.*

1. With the highlight on the **<create>** marker in the **Data** panel, press ↵.

   The database design screen shown in Figure 7-4 is displayed with the cursor in the **Field Name** column.

2. Type lastname in the **Field Name** field.

   Notice that dBASE IV enters the field name in all capital letters even though you typed it in lowercase.

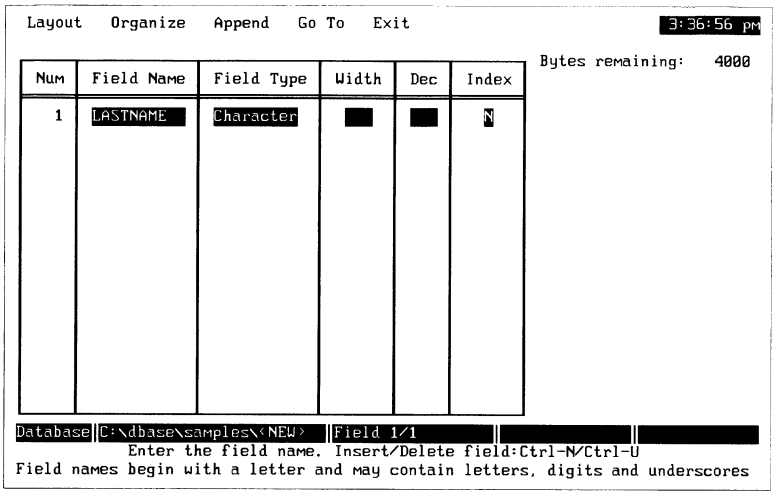


Figure 7-4 Database design screen with Lastname field

3. Press ↵ to move the cursor to the next column, **Field Type**.

   The preset field type in dBASE IV, Character, already appears in the column.

4. Because Lastname is to be a character-type field, press ↵ to move the cursor to the next column, **Width**.

5. Type 15 and press ↵.

   The cursor skips past the **Dec** column, used to specify number of decimal places for a numeric field, and moves to the **Index** column. If you were to enter Y (yes) in this column, dBASE IV would create an index based on the values entered for this field. Indexing is discussed in the Reorganizing Data section later in this chapter.

6. Press ↵ without entering anything in the **Index** column.

   The number **2** appears in the **Num** column of the next line, indicating that dBASE IV is ready for you to enter the information for a second field.

7. Repeat steps 2 through 6 to enter information for fields 2 through 7, listed in Table 7-2.

Table 7-2 Remaining field information for new database file

Num	Field Name	Field Type	Width
2	FIRSTNAME	Character	10
3	ADDRESS	Character	20
4	CITY	Character	14
5	STATE	Character	2
6	ZIP	Character	5
7	PHONE	Character	13

You've now added most of the fields for your new database. The last field that you're going to enter, named Business, will enable you to distinguish a record for a business acquaintance from that of a social acquaintance. This field has the *logical* data type and can contain only *T* or *F* (for True or False), or *Y* or *N* (for Yes or No).

8. With the cursor in the **Field Name** column for field 8, type business and press ↵.
9. Type L (for Logical) or press **Spacebar** until the Logical field type appears.

   The number 1 automatically appears in the **Width** column and the cursor moves to the next line.
10. Press ↵ with the cursor in the empty line to tell dBASE IV that you're finished entering field information.

    A prompt appears asking you for the name of the new database file.
11. Enter mynames and press ↵.

    Now you're prompted at the bottom of the screen to begin entering records into your database.
12. Type Y for Yes.

    The Edit screen illustrated in Figure 7-5 appears for the first record of Mynames with the cursor in the LASTNAME field. Notice that this is a *form* for entering data.

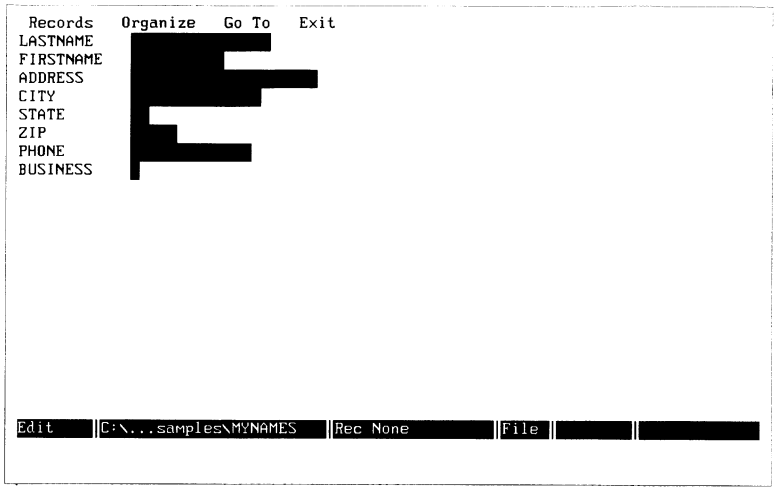


Figure 7-5 Data entry screen for Mynames

13. Enter field information for someone you know. Press ↵ to terminate each field entry.

    If you notice a mistake while typing, press ↑, ↓, ←, or → to move the cursor to the mistake and type over it. **Home** moves the cursor to the beginning of a field and **End** moves the cursor to the end of a field.

    Notice that when you've completely filled a field (for example, State, Zip, or Phone), the cursor immediately jumps to the next field. When you have entered information for the Business field (Y or N, depending on whether or not the person is a business acquaintance), an empty data entry form appears for you to enter a new record.

14. Type as many records as you like in your new database file.

    During data entry you can use the **PgUp** and **PgDn** keys to move up and down between records. If you press **PgDn** while on the last record, you are prompted to add a new record. If you type Y, a blank data entry form appears. If you type N, the cursor moves to the first field of the current record.

15. When you're finished entering records, press ↵ when the next empty data entry form appears.

    The records that you have entered are saved and the Control Center appears. The name of the database file that you just created, Mynames, is highlighted in the **Data** panel above a horizontal line, indicating that it is the active file.

Working with Database Files

Once you've created a database file, you need to update it periodically to add new records, edit existing ones, and discard obsolete ones. This section shows you how to perform all of these operations using the Names sample file. You will also learn how to reorganize your data to make it more manageable.

Displaying Data

Start by displaying the records in the Names database file, which is listed in the **Data** panel:

1. In the **Data** panel, press ↓ until **NAMES** is highlighted.
2. Press ↵.

   A prompt box appears, giving you the following choices:

   - **Use file** — activates the database file without displaying data and redisplays the Control Center.
   - **Modify structure/order** — displays the database design screen to allow you to change the structure of the database file or maintain its indexes.
   - **Display data** — displays the data in the database file.
3. Press → until **Display data** is highlighted and press ↵.

   The Edit screen shown in Figure 7-6 appears. The screen displays a form containing information for the first record in the Names database file, because you used an Edit screen earlier to enter records for your Mynames file (see Figure 7-5). Also, this form is identical to the one that you used earlier because the *structure* of the Names file (field names, types, and widths) is identical to that of Mynames.

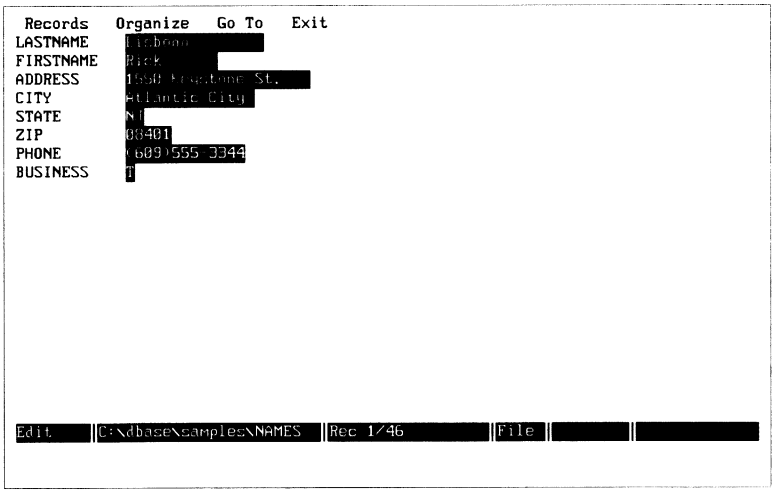
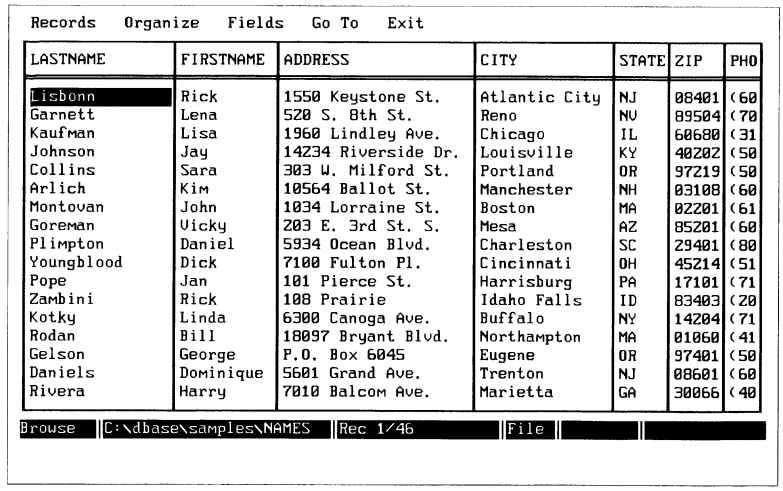


Figure 7-6 Edit screen for Names

4. To display a different view of the Names data, press **F2 Data**.

   The screen now shows the Names file data in a table, as shown in Figure 7-7.



Records Organize Fields Go To Exit

LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	PHO
Lisbonn	Rick	1550 Keystone St.	Atlantic City	NJ	08401	(60
Garnett	Lena	520 S. 8th St.	Reno	NV	89504	(70
Kaufman	Lisa	1960 Lindley Ave.	Chicago	IL	60680	(31
Johnson	Jay	14234 Riverside Dr.	Louisville	KY	40202	(50
Collins	Sara	303 W. Milford St.	Portland	OR	97219	(50
Arlich	Kim	10564 Ballot St.	Manchester	NH	03108	(60
Montovan	John	1034 Lorraine St.	Boston	MA	02201	(61
Goreman	Vicky	203 E. 3rd St. S.	Mesa	AZ	85201	(60
Plimpton	Daniel	5934 Ocean Blvd.	Charleston	SC	29401	(80
Youngblood	Dick	7100 Fulton Pl.	Cincinnati	OH	45214	(51
Pope	Jan	101 Pierce St.	Harrisburg	PA	17101	(71
Zambini	Rick	108 Prairie	Idaho Falls	ID	83403	(20
Kotky	Linda	6300 Canoga Ave.	Buffalo	NY	14204	(71
Rodan	Bill	18097 Bryant Blvd.	Northampton	MA	01060	(41
Gelson	George	P.O. Box 6045	Eugene	OR	97401	(50
Daniels	Dominique	5601 Grand Ave.	Trenton	NJ	08601	(60
Rivera	Harry	7010 Balcom Ave.	Marietta	GA	30066	(40

Browse | C:\dbase\samples\NAMES | Rec 1/46 | File

Figure 7-7 Browse screen for Names

Notice that each record is a separate row of the table. In dBASE IV, this table format is called a *Browse* screen because you can use it to browse quickly through your records.

F2 Data allows you to toggle between Browse and Edit screens. This is one of ten *function keys* that let you accomplish a dBASE IV task with a single keystroke. Chapter 8 contains more information about function keys.

Editing Data in the File

You can use either the Browse screen or the Edit screen to change data in a database file or to add new records.

Using the Browse screen, change the zip code entry for the record whose Lastname field is currently highlighted.

1. Press **Tab** to move the highlight to the Zip field.
2. Type in your home zip code.

dBASE IV beeps once to indicate that the information you added filled the entire field, and moves the cursor to the beginning of the Phone field.

To undo your change while the cursor is still on the highlighted record:

1. Press **F10 Menus** to display the **Records** menu with **Undo change to record** highlighted.
2. Press ↵ to choose **Undo change to record**.

Notice that the Zip field information that you changed is immediately redisplayed.

WARNING
Once you've moved the cursor to another record, you cannot use the **Undo change to record** option to remove changes to a previous record.

Adding a New Record

To add new records to the file using the Browse screen, you could keep pressing ↓ until you reached the end of the file and were prompted to add new records. Happily, there is a quicker method.

As you've noticed, the menu bar at the top of the screen lists menu titles. Whenever you've pressed **F10 Menus**, the first menu on the left has appeared. As you've pressed the ← or → key, each menu on the menu bar, in turn, has appeared. Each item on a menu is an *option* that you can select in one of two ways:

- Press ↑ or ↓ to move the highlight to the option and press ↵.
- Type the first letter of the option title.

Once you have finished using a menu, you can remove it from the screen by pressing **Esc**.

To add a new record to the Names file:

1. Press **F10 Menus**.

   The **Records** menu appears with the **Add new records** option highlighted.

2. Select **Add new records**.

   A blank record appears with the cursor in the Lastname field.

3. Enter information for yourself in all fields except Business.

4. Press **Esc** to redisplay the Control Center without saving the new record.

Notice that Names is now the active database file: its name appears above the horizontal line in the Data panel.

NOTE

Once you have moved the cursor from a newly entered record to another record line, the new record is automatically saved. In step 3, entering information for Business would automatically have moved the cursor to the next new record line, thereby saving the new record. To save a new record and redisplay the Control Center without moving the cursor to another record line, press **Ctrl-End**.

Reorganizing Data

When you browse or edit a database file (that is, use the Browse or Edit screen to display and edit data), records normally appear in the order in which they were entered. This is known as *natural* order. Usually, it is more convenient to have the records arranged in a logical order.

dBASE IV gives you several ways to organize records in a database file. The most common method is indexing.

A dBASE IV *index* is a file that determines the order in which the records in a database file are displayed. The actual order of the records in the database file remains unchanged.

1. Choose the Names database file on the **Data** panel by highlighting the filename and pressing ↵.

2. From the prompt box, choose the **Display data** option.

   The Browse screen appears.

3. Choose the **Order records by index** option from the **Organize** menu.

   A list of the four indexes currently defined for Names appears. As you highlight each index name, a box opposite displays the fields on which the index is based.

   Now you are going to create a new index.

4. Press **Esc** to remove the display of index names and choose the **Create new index** option.

   The **Create new index** menu appears with the **Name of index** option highlighted. Note that this menu includes instructions for creating a new index.

5. Press ↵.

   The curly braces disappear and the cursor appears in their place. The instructions at the bottom of the screen prompt you to specify a name for your new index.

6. Type FRIENDS and press ↵.

   Your entry is enclosed by curly braces and **Index expression** is highlighted.

7. Press ↵, press **Shift-F1** to display the field names for Names, select FIRSTNAME, and press ↵ again.

FOR clause is highlighted.

8. Press ↵, type .NOT. Business to restrict rows displayed to those of non-business acquaintances, and press ↵. (.NOT., a logical operator, must be delimited by periods.)

   Order of index is highlighted along with its default, ASCENDING, which specifies alphabetical order for a character field. (You could change the index order to DESCENDING, the opposite of ASCENDING, by pressing ↵ at this point.)

9. Press **Ctrl-End**.

   Your new index is saved and the Names records are listed alphabetically by first name. The FOR clause has caused records whose Business fields contain .T. not to be displayed.

10. Use **PgDn** to scroll through the records. Notice the alphabetical order in the Firstname field. Press **End** to see the F (False) values in the Business field.

11. Select Order **records by index** from the **Organize** menu.

    Notice that there are now five indexes displayed, including the Friends index that you just created.

12. Highlight one of the other indexes and press ↵.

    Notice that the Names records are reorganized by the index that you've chosen.

13. To go back to the Control Center, press **Esc**.

Filtering Data

You can use a dBASE query to display a view of a database file that looks like the same file as organized by index. A query is a file containing instructions for displaying or updating data.

Now, create a simple query on the Names database file to display only Lastname, Firstname, Phone, and Business fields for records whose Firstname fields contain "John."

1. With the Names file in use ("NAMES" is displayed above the horizontal line in the **Data** panel), select Names and then select **Close file**.

   The Control Center appears with Names displayed below the horizontal line in the **Data** panel.

NOTE
To create a query for a database file without first closing the file, refer to Chapter 9.

2. Press → to highlight the **<create>** marker in the **Queries** panel and press ↵.

   The **Queries** menu bar is displayed with the **Layout** menu open.

3. Press ↵ to select **Add file to query**.

   Two boxes are displayed, the one to the right listing the names of database files in the current catalog (Samples), the one below containing the description of the currently highlighted database file.

4. Choose Names.

   The queries design screen appears with the *file skeleton* of the Names.dbf file, as shown in Figure 7-8. The file skeleton displays the field names of the database file across the top of the screen.

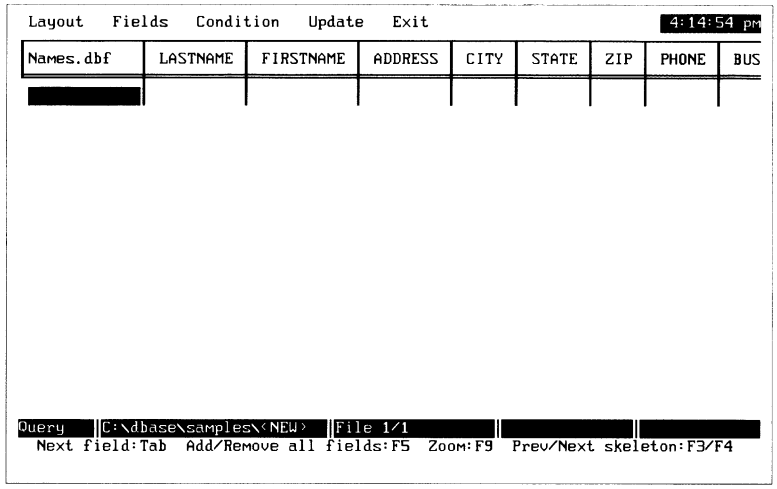


Figure 7-8 Queries design screen

5. Press **Tab** to move the cursor to the Lastname field and then press **F5 Field**.

   Another skeleton, called the view *skeleton*, is created at the bottom of the screen and the Lastname field is added to the view skeleton.

6. Press **Tab** again to highlight the Firstname field and press **F5 Field**.

   The Firstname field is added to the view skeleton.

7. Repeat the process described in steps 5 and 6 to add the Phone and Business fields to the view skeleton.

8. Press **Shift-Tab** until the Firstname field is highlighted and type "John" (include the quotation marks).

   The completed queries design screen for query John is shown in Figure 7-9.

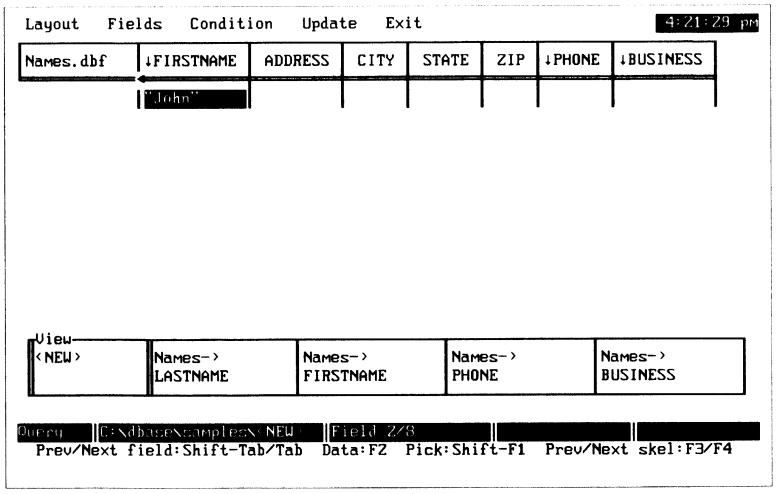


Figure 7-9 Queries design screen with completed query

9. Press **Ctrl-End** to save the query.
10. At the **Save as:** prompt, type John (without quotation marks) and press ↵.

    dBASE IV processes the query and redisplays the Control Center with the name of the new query above the horizontal line in the **Queries** panel.
11. To display the Names data that has been filtered by query John, press ↵ and select **Display data**.

    A Browse screen displays Names records whose Firstname fields contain "John." Notice that two of the records are for business acquaintances (T), the other two for non-business acquaintances (F).
12. Press **F10 Menus** and select **Exit** from the **Exit** menu to return to the Control Center.
13. Close the John query by selecting its name from the **Queries** panel and then selecting **Close view**.
14. If you like, select another query from the **Queries** panel. Select **Modify query** and study the structure of the query. Then press **F2 Data** to see how the query filters data.
15. Select **Exit** from the **Exit** menu to return to the Control Center.

Printing Data

You can print the data that you've stored and displayed using customized reports, form letters, and mailing labels. For times when you need a fast and informal way to print data, use the dBASE IV Quick Report feature.

1. Highlight Names in the **Data** panel and press **Shift-F9 Quick Report**.

   A menu of printing choices appears with the **Begin printing** option highlighted. Make sure that your printer is on.

   (If a printer is not connected to your computer, select the **View report on screen** option instead of **Begin printing**. When you use the **View report on screen** option, the first page of the Quick Report appears on the screen. Press **Spacebar** to view each page of the report. To stop the Quick Report display, press **Esc**.)

2. Press ↵ to select **Begin printing**.

A report of Names data begins printing. When the report is printed, the Control Center reappears.

NOTE
If you have trouble printing a report, press **Ctrl**-S *to pause printing (and again to resume), or* **Esc** *to cancel printing. If your printer has a buffer, printing will not stop immediately, but will continue until the buffer is empty.*

You also can organize Names data before printing a Quick Report.

1. Select Names from the **Data** panel and select **Modify Structure/order**.
2. From the **Organize** menu, select **Order records by index**.

   Note that this is the same **Organize** menu that you used earlier on the Browse screen.

3. From the box of index names, select the ZIP index. Press **F2 Data** to view the data ordered by zip code.
4. Press **Shift-F9 Quick Report** and select **Begin printing** to print Names records ordered by zip code. Press **Esc** to return to the Control Center.
5. Now, quit dBASE IV by pressing **F10 Menus** and then selecting **Quit to DOS** from the **Exit** menu. (As a shortcut, press **Alt**-E and then type q.)

Summary

You have now practiced some of the most common operations of dBASE IV. You have created a database file and entered data, displayed and edited data, and added a new record to a database file. You also have learned how to create and use an index to organize data, how to create and use a query to filter the data that is displayed, and how to print a Quick Report.

Important as they are, these operations barely tap the capabilities of dBASE IV. The chapters that follow will give you a better idea of the many ways in which dBASE IV can help you manage your data.

Below are summaries of some of the operations you performed in this chapter.

To select a menu:

Press **F10 Menus** to move to the menu bar and open a menu.

To select a menu option:

Use the → or ← key to open the desired menu. Or, press **Alt** while pressing the first letter of the menu title. Use the ↑ or ↓ key to highlight the desired option and press ↵, or type the first letter of the option title.

To create a database file:

Highlight the **<create>** marker in the **Data** panel of the Control Center and press ↵.

On the database design screen that appears, enter the name, type, width, and any other necessary information for each field in the database file. Use the ↵ key to move from column to column. Press ↵ on an empty line to finish entering field information. Enter a filename for the file at the prompt and press ↵.

To enter records right away, type Y at the prompt that appears. To go back to the Control Center without entering records, type N.

Enter each record by typing information in the fields of the Edit screen that is displayed. Use the **PgUp** and **PgDn** keys to display records above and below the current record. To finish entering records, press ↵ when the next empty Edit screen form is displayed.

To display data:

Highlight a filename in the Data panel and press ↵. Select **Display data** in the menu box that appears and press ↵.

Press **F2 Data** to switch back and forth between displaying data in a Browse screen table and displaying one record at a time in an Edit screen form.

To add or change data:

On the Browse or Edit screen, move the highlight to the record and field in which you want to add or change data and type the new data. To undo a change to the current record, select **Undo change to record** from the **Records** menu.

To add a new record:

On the Browse or Edit screen, select **Add new records** from the **Records** menu. In the blank line or form that appears, type data for the new record. To finish data entry on the Edit screen, press ↵ when the next blank form appears.

To finish data entry on the Browse screen, press **Ctrl-End** after completing the new record and before pressing ↵ to move to a new record line. (If the cursor skips automatically to a new record line, press ↑ to move the cursor to the last new record that you want to add before pressing **Ctrl-End**; otherwise, a blank record will be created.)

To create an index:

Highlight the file to be indexed in the Data panel and press ↵. Select **Modify structure/order** or **Display data** from the prompt box.

Select **Create new index** from the **Organize** menu. Type the name of the new index; type the name of the field on which you want to index, or press **Shift-F1** to select the field name from a list; if desired, enter a FOR clause condition to limit the rows that are indexed; and choose the index order (ASCENDING or DESCENDING).

Press **Ctrl-End** to save the index and reorganize the database file. If you are indexing from the database design screen, press **F2 Data** to display file records in the new order.

To reorganize data using an existing index:

Highlight the file to be indexed in the Data panel and press ↵. Select **Modify structure/order** or **Display data** from the prompt box.

Select **Order records by index** from the **Organize** menu. Select the desired index from the list that appears.

If you are reorganizing from the database design screen, press **F2 Data** to view the records in the new order.

To create a query for filtering data:

Close the database file on which the new query is to operate. Highlight the **<create>** marker in the **Queries** panel and press ↵.

Select **Add file to query** from the **Layout** menu. In the box that appears to the right, highlight the name of the database file to be queried and press ↵.

Press **Tab** or **Shift-Tab** to move the highlight to each field in the database file skeleton that is to be used by the query. While the field is highlighted, press **F5 Field** to add the field to the view skeleton being created at the bottom of the screen.

Highlight any field that is to contain a query condition and type the condition in the field. Press **Ctrl-End** to save the query. At the prompt, type the name of the query and press ↵.

To display the file data that has been filtered using the new query, highlight the query name in the **Queries** panel of the Control Center and press ↵. Select **Display data** from the prompt box.

To filter data using an existing query:

Highlight the query name in the **Queries** panel of the Control Center and press ↵. Select **Display data** from the prompt box.

To print a Quick Report:

In the **Data** panel of the Control Center, select the name of the file whose data is to be reported and press **Shift-F9 Quick Report**. Select **Begin printing** from the **Print** menu.

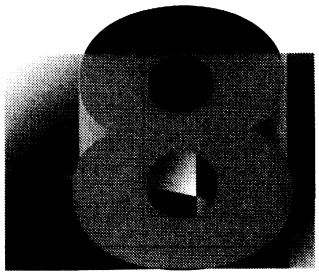
To organize data with an index before printing, highlight the filename in the **Data** panel and then select **Modify structure/order** or **Display data** from the prompt box.

Select **Order records by index** from the **Organize** menu and select the name of the index in the box that appears. If you are using the Browse or Edit screen, press **Shift-F9 Quick Report** and select **Begin printing** from the **Print** menu. If you are using the database design screen, exit to the Control Center before printing.

To view a Quick Report on your screen:

After pressing **Shift-F9 Quick Report**, select **View report on screen** from the **Print** menu instead of **Begin printing**.

The Menu System



As you have seen, dBASE IV provides a comprehensive system of menus. Menu options take you to full-screen displays that allow you to create, modify, and organize data with the help of other menu options.

This chapter briefly describes the elements of the dBASE IV menu system:

- Basic screen elements such as the menu bar, clock, status bar, navigation line, and message line
- How to display menus and select options
- How to use function keys and the Help system
- How to quit dBASE IV

Preparing for This Chapter

If the Control Center with the sample files is not already displayed, type **dbsample** at the operating system prompt.

If you have a problem displaying the Control Center and the sample files, refer to the instructions in the Displaying the Control Center section of Chapter 7.

Basic Screen Elements

Although the Control Center screen contains file panels, most of the screens that you use to design and modify database files, queries, forms, reports, and labels contain a *work surface* that you use to enter and manipulate data.

Figure 8-1 shows the elements of a typical dBASE IV screen.

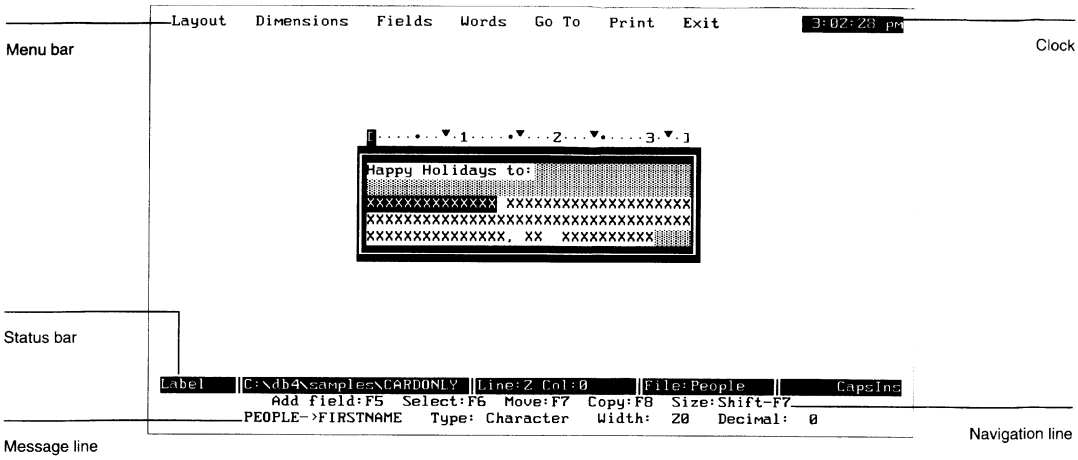


Figure 8-1 Elements of the labels design screen

Menu Bar

The menu bar displays the titles of available menus.

Clock

The clock shows the current time, as set by your operating system.

Status Bar

The status bar is composed of five segments. In Figure 8-1, each segment displays, from left to right:

1. The type of screen that you are viewing (labels design).
2. The name of the file that is currently in use and its operating system path (the Cardonly file in the Samples directory).
3. The location of the cursor on the work surface (line 2, column 0).
4. The source database file or view (the People database file).
5. The keyboard mode (**Caps** for **Caps Lock** and **Ins** for Insert mode).

Navigation Line

The navigation line shows important keystrokes that you can use on the screen.

Message Line

When a menu is open, the message line carries a brief explanation of the highlighted menu option. Otherwise, the line displays prompts or informational messages.

Using Menus

A dBASE IV menu is a list of choices that you can make for a type of file or for the activity performed on a screen. You open a menu to see its options; you press **Esc** to make the menu disappear. Figure 8-2 shows a typical menu.

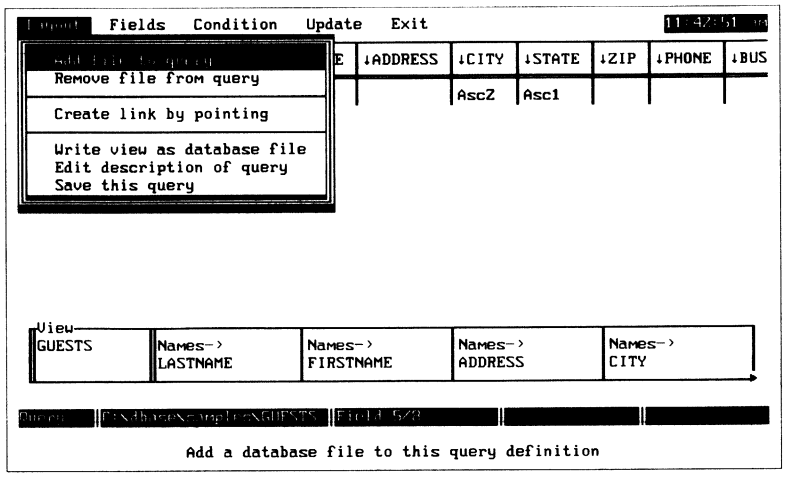


Figure 8-2 dBASE IV menu

Displaying Menus

Pressing **F10 Menus** opens a menu on the menu bar:

- If you have not yet opened a menu while viewing the current screen, the first menu on the menu bar is opened.
- If you have already opened menus on the screen, the last menu displayed is opened.

Selecting Options

Once you open a menu, you can select an option from it in one of the following ways:

- Move the highlight to the option and press ↵.
- Type the first letter of the option title.

Types of Menu Options

Selecting a menu option may have one of the following results:

- A box containing descriptive information about a currently active item appears with an instruction about what to do with it (for example, the **Edit description of catalog** option of the **Catalog** menu on the Control Center screen).
- A box containing a list of files, fields, or other items appears and you are asked to choose one of them by highlighting its name and pressing ↵ (for example, the **Add file to query** option of the **Layout** menu on the queries design screen).
- A menu related to the option (called a *submenu*) appears and you are asked to select one of the options. (Such an option is preceded by an arrowhead symbol, for example, the **Export** option of the Control Center **Tools** menu.)
- An action occurs (for example, the **Exit to dot prompt** option of the Control Center **Exit** menu).

Function Keys

Pressing a function key lets you perform an often-used dBASE IV operation without having to navigate through menus and options. For example, pressing **F2 Data** while viewing an Edit screen displays a Browse screen.

You also can hold down the **Shift** key while pressing a function key to perform a different function. For example, pressing **Shift-F2 Design** displays the database design screen for the highlighted database file with the **Organize** menu opened.

You have already used **F2 Data** and **F10 Menus**. Later chapters will introduce you to other function keys. In the meantime, use the keyboard template provided with dBASE IV as a quick guide to the actions performed by each function key.

Getting Help

While you are performing any dBASE IV operation, you can display comprehensive information about the function by pressing the **F1 Help** key.

F1 Help Key

Pressing **F1 Help** displays information about the topic suggested by the current cursor location. For example, if the cursor is on a menu option, dBASE IV shows you information about that option.

1. Highlight the **<create>** marker on the Control Center **Data** panel.
2. Press **F1 Help**.

dBASE IV displays the Help box shown in Figure 8-3. The Help box is superimposed on the display with which you are working, making it easy for you to view the Help information in context.

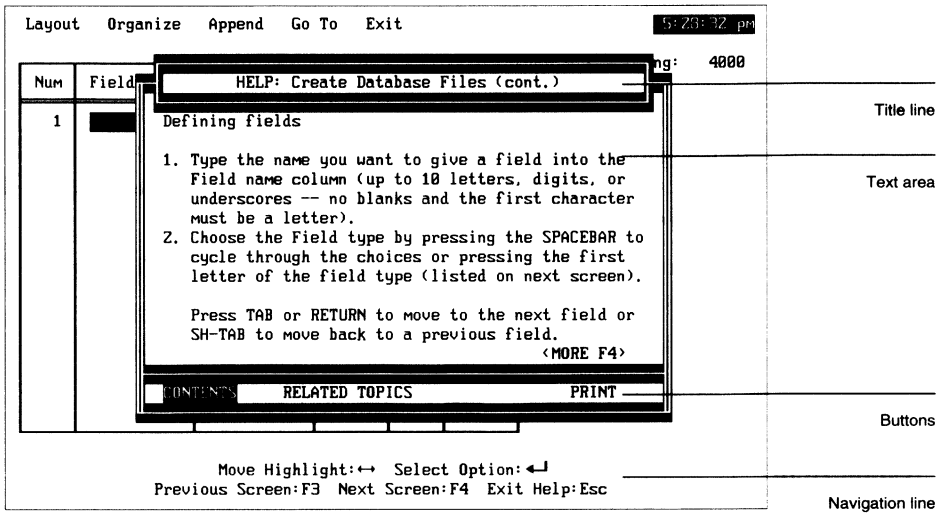


Figure 8-3 Help box

A Help box consists of four elements: title line, text area, buttons, and navigation line.

Title Line

The title line identifies the topic covered by the Help box text.

Text Area

The text area contains an explanation of the topic. If there is more than one page of text, you can view the next page by pressing **F4 Next Screen**. You can return to the previous page by pressing **F3 Previous Screen**.

Buttons

The buttons at the bottom of the Help box let you select Help options. You activate a button by highlighting it and pressing ↵, or by typing the first letter of its name. The buttons used on Help boxes are:

- **CONTENTS** — displays a table of contents for the current Help area. For more general information on a listed topic, press the **F3 More General** key; for more specific information, press **F4 More Specific**.
- **RELATED TOPICS** — lets you choose a topic that relates to the current topic. For example, a Help box about sorting database files would have indexing as a related topic. When you select a related topic, information about that topic appears.

- **BACKUP** — allows you to retrace your steps through the Help boxes that you have viewed on your way to the current Help box. Accordingly, this button does not appear on the first Help screen that you display.
- **PRINT** — automatically prints the current page of Help information. This is a useful feature if you find yourself returning often to the same Help boxes.

Navigation Line

The navigation line lists the keys that you use to control a Help display. In Figure 8-3 these are: **Move Highlight:← →, Select Option:↵, Previous Screen:F3**, **Next Screen:F4**, and **Exit Help:Esc**.

While viewing a table of contents screen, pressing F3 and F4 provides the following functions:

- **More general:F3** — display a more general level of contents
- **More specific:F4** — change to more specific topics

Using Help

Now take some time to explore Help on your own.

1. While displaying the Help box shown in Figure 8-3, press **F4 Next** to display the next page of information about creating a database file.

   Notice the **<MORE F4>** message in the lower right of the Help box. You'll see this message as long as there is more information to be displayed about the current topic.

2. Press → and ↵ to choose the **RELATED TOPICS** button.

   A box appears showing topics that are related to the topic of database files.

3. Press ↵ to choose **Help on Help**.

   Help on Help is included in all Related Topics boxes.

4. Read through each screen about using Help, pressing **F4 Next** to display the next screen.

   On the third screen, the message **<MORE F4>** is no longer displayed, indicating that there is no more information for **Help on Help.**

5. Press → and ↵ to select the **BACKUP** button (or type B).

   The previous **Help on Help** screen is displayed.

6. Press B until you return to the first **Help: Create Database Files** screen.

   Notice that the highlight jumps to the **CONTENTS** button.

7. Press ↵.

   A table of contents appears for topics related to files, the current area of Help.

8. Press **F4 Next** to redisplay the first **Help: Create Database Files** screen.
9. If the printer connected to your computer is turned on, select the **PRINT** button to print the currently displayed Help text.
10. Press **Esc** to remove the Help box from the screen.

Quitting dBASE IV

To leave dBASE IV and display the operating system prompt, choose **Quit to DOS** from the Control Center **Exit** menu, shown in Figure 8-4.

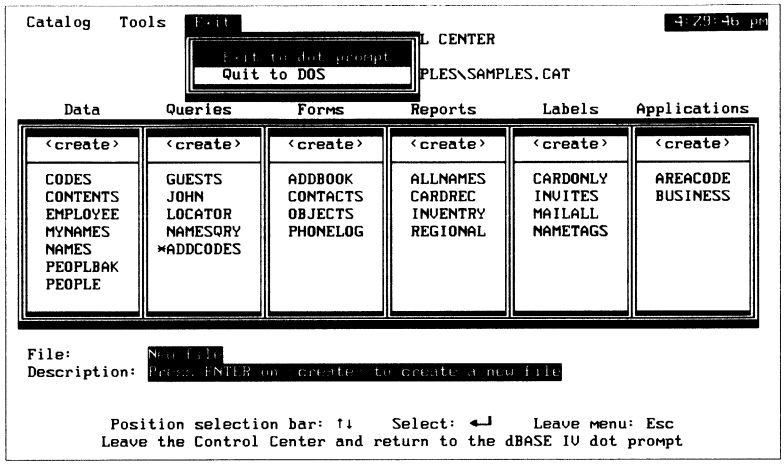


Figure 8-4 **Exit** menu

Summary

This chapter introduced you to the dBASE IV menu system. The skills you've just learned are summarized below. In the following chapters, you will be using these skills to work with the menu system.

To choose menus and menu options:

Press **F10 Menus** to activate the menu bar and open a menu. Use the ← or → key to display the desired menu.

Or, press **Alt** plus the first letter of the menu name.

Use the ↑ or ↓ key to highlight the desired menu option and press ↵ to select the option.

Or, type the first letter of the option name.

To remove the menu from the screen, press **Esc**.

To get Help in the menu system:

Press **F1 Help**.

Choose the **CONTENTS** button in the Help box to display the table of contents for the topic you've selected. While viewing a table of contents screen, press **F3** to display a more general level of contents, **F4** to view more specific topics.

Choose the **RELATED TOPICS** button to display Help on topics related to the current topic.

Choose the **BACKUP** button to display the previous Help boxes that you have viewed during a search.

Choose the **PRINT** button to print the current Help text.

On a topic screen, press **F3** to view the previous screen, **F4** to view the next screen.

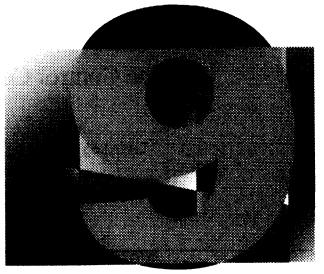
Press **Esc** to remove the Help box from the screen.

To quit dBASE IV:

Press **F10 Menus** and open the **Exit** menu.

Choose **Quit to DOS**.

A Tour of the Control Center



In Chapter 7, you used the **Data** panel of the Control Center to create a database file and edit data. You also used the **Queries** panel to display selected data in a database file.

This chapter lets you perform simple tasks using the **Queries** panel and other Control Center panels. You will learn how to:

- Navigate between Control Center panels and select files for use
- Create a query using the **Queries** panel and use the query to display data in a database file
- Display database records using a data entry form selected from the **Forms** panel
- Choose a report from the **Reports** panel and print it
- Select a label from the **Labels** panel and use it to print labels
- Run an application from the **Applications** panel

After performing the exercises in this chapter, you can refer to *Using dBASE IV* for more detailed information about any of these tasks.

Preparing for This Chapter

If the Control Center with the sample files is not already displayed, type dbsample at the operating system prompt and press ↵.

If you have a problem displaying the Control Center and the sample files, refer to the instructions in the Displaying the Control Center section of Chapter 7.

Control Center Screen

The Control Center screen is shown in Figure 9-1.

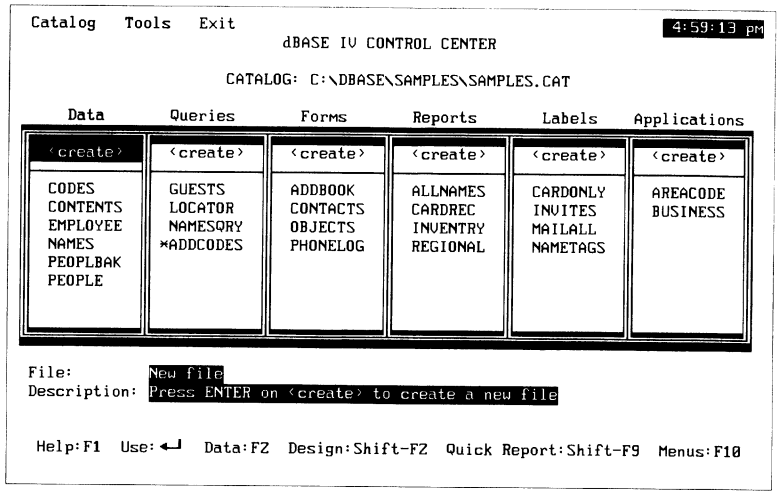


Figure 9-1 Control Center screen

Menus

The Control Center menu bar contains the following menus:

- **Catalog** — provides options for managing catalogs.
- **Tools** — provides utilities for accessing the operating system, making and storing keyboard macros, importing and exporting files, implementing data protection, and changing system settings.
- **Exit** — used to leave the Control Center and go to the dot prompt or the operating system prompt.

Catalog Name

In Figure 9-1, the name of the active catalog, Samples.cat, appears above the panels in the center of the Control Center screen.

A catalog is a file that you can create to group information about all of the files that you use for a particular purpose, or *application*, so that they can be conveniently accessed when needed. The Samples catalog file, for example, contains information about the files that you are using for the exercises in this section of the manual.

Catalog file information includes filename, file type (.dbf, .qbe, and so on), alias name (if any), path (if the catalog does not reside on the current, or *default*, drive and directory), and description (if any). If the same database file is used for more than one application, information about the file can be included in more than one catalog file.

When, at the beginning of this chapter, you entered dbsample at the operating system prompt, the directory was automatically changed to \DBASE\SAMPLES and dBASE IV was started. Because Samples is the only catalog file in the Samples directory, it was automatically activated and its files displayed in the Control Center.

NOTE
If you enter dbase *instead of* dbsample *at the operating system prompt, dBASE IV is started from the DBASE, rather than the SAMPLES, subdirectory, and the Control Center appears with Untitled.cat as the active catalog file. Because this catalog file contains no files information, the Control Center panels contain no filenames.*

Refer to Chapter 3 of *Using dBASE IV* for complete information about working with catalog files.

Panels

Each of the six Control Center panels contains files used for performing a different type of dBASE IV activity. The types of files available in Control Center panels are described in Chapter 7.

The files in different panels are often directly related. For example, a database file can have a number of query files for filtering its data, a forms file for data entry, report files for displaying and printing its data, and label files for printing labels containing its data.

While the highlight is within a panel, you can create new files to be added to the current catalog using the **<create>** marker, or select from the existing files that appear in the panel.

When you select a database file for use, its name appears at the top of the **Data** panel, separated from the other database filenames by a horizontal line. Simultaneously, the names of all of its related files (except for query files) appear at the top of their panels, separated from other filenames by a horizontal line.

The Working with Files section, later in this chapter, discusses how to work with each type of file displayed in the panels.

Navigating in Panels

When the Control Center is first displayed, the highlight is on the **<create>** marker of the **Data** panel. Use the following keys to move the highlight within the panels:

- ← or → moves the highlight between panels
- ↑ or ↓ highlights the **<create>** marker or individual filenames within a panel

- **Home** and **End** move the highlight between the top and bottom of a panel
- **PgUp** and **PgDn** move the cursor up and down a long list of filenames one page at a time

If more filenames are available in a panel than can be shown on the screen, you can scroll the filenames using the last two types of keys.

Choosing Files

You select a file by moving the highlight to the filename and pressing ↵. A prompt box appears giving you choices of action.

The choices of action vary, depending on the type of file. You choose an action by moving the highlight to the action name and pressing ↵, or by typing the first letter of the action.

Choose the Names database file and display the prompt box, as follows:

1. Move the highlight to **NAMES** in the **Data** panel.
2. Press ↵.

   The prompt box appears.
3. Press ↵ to choose **Use file**.

   The **NAMES** filename appears at the top of the **Data** panel, separated from the other filenames by a horizontal line. Notice that its related forms file, **CONTACTS**, also appears at the top of the **Forms** panel.

Once you have highlighted a filename, as in step 1, you can use the following shortcuts to work with the file:

- Press ↵ twice to open a closed file or close an open one.
- Press **Shift-F2 Design** to display the data design screen so that you can modify the file's structure.
- Press **F2 Data** to display file data.

Working with Files

In the chapters that follow, you will be working extensively with database files. For now, take a look at the other types of dBASE IV files and what you can do with them.

Queries

Queries are a powerful tool for organizing and managing data. For example, a large database file is easier to work with if you define a view query that includes only the information that you need most frequently. Or, you can define an update query that automatically updates fields that contain the same information.

In Chapter 7, you created a simple query that located the Names records whose Firstname fields contained "John." Now, you'll create a simple view query of Names to list only Lastname, Firstname, State, and Telephone number information for business acquaintances.

1. Move the highlight to the **<create>** marker in the **Queries** panel and press ↵.

   The queries design screen appears. Because this time you selected Names before displaying queries design, a view skeleton containing all of the fields in Names.dbf is already displayed at the bottom of the screen.

2. Press **Tab** until the ADDRESS field of the file skeleton is highlighted.

3. Select **Remove field from view** from the **Fields** menu.

   The ADDRESS field disappears from the view skeleton at the bottom of the screen.

4. Repeat the process in steps 2 and 3 to remove the BUSINESS field from the view skeleton (or press **F5 Field** when the field is highlighted).

5. While the BUSINESS field is still highlighted, type .T. (be sure to include the periods).

   dBASE IV will display only records that contain .T. (True) in the Business field of the Names file.

6. Press **Shift-Tab** to highlight the STATE field and type ASC.

   dBASE IV will arrange records alphabetically (in ascending order) by State values.

7. Select **Edit description of query** from the **Layout** menu, type Business telephone numbers arranged by State in the prompt box, and press ↵.

8. Press **F2 Data**.

   A Browse screen displays the selected records.

9. Press **Esc**.

   You are prompted to save the query design.

10. Type Y to save the query. In the **Save as** prompt box, type BUSPHONE and press ↵ to return to the Control Center.

Notice that the name of the new query file is displayed in the **Queries** panel.

In addition to saving this view query as a query file, you could use the **Layout** menu option **Write view as database file** to save the query and its data as a database file.

You can use a view query to display data from more than one file. Refer to *Using dBASE IV*, Chapter 6.

Forms

As you learned in Chapter 7, you can display data in two different ways: as a table using the Browse screen, or as a form using the Edit screen. Pressing **F2 Data** enables you to toggle back and forth between each form of display.

The Edit form associated with a database file is simply a vertical display of the fields in each database record. The size of any field, as in the Browse table, is dictated by the size defined in the file structure.

Using the **Forms** panel of the Control Center, you can design a customized form for entering and displaying the data in a database file. To get an idea of the kind of form you can design, follow these steps:

1. Choose the **CONTENTS** file from the **Data** panel.
2. Select **Use file** from the prompt box.

   Notice that **CONTENTS** appears above the horizontal line in the **Data** panel and that **OBJECTS** appears above the horizontal line in the **Forms** panel.
3. Highlight the **OBJECTS** form file in the **Forms** panel and press **F2 Data**.

   The first record of the Contents database file is displayed using the custom form defined in the Objects form file, as shown in Figure 9-2. Notice how boxes and lines are used to draw attention to different fields of the record.
4. To return to the Control Center, press **Esc**.

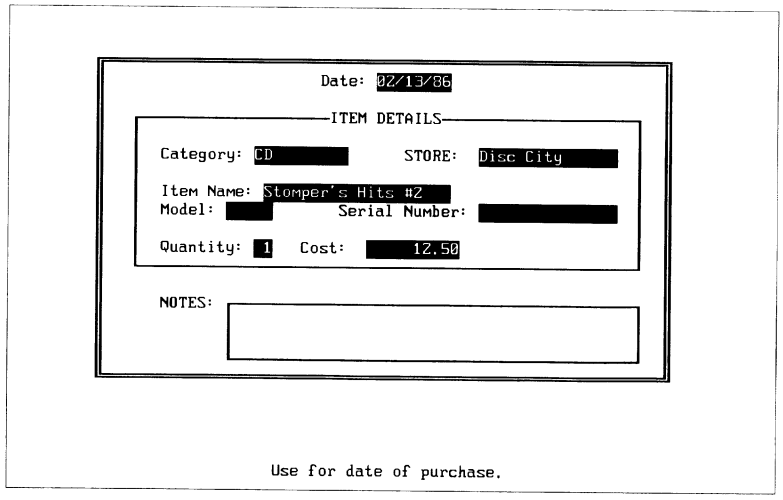


Figure 9-2 Form display for Contents database file

Reports

You can design a report to organize and print database records in your own special format, rather than using the Quick Report format that dBASE IV provides.

Using a report, you can organize your data into subgroups, print totals for each subgroup, and then print a grand total. You can limit the fields that appear in the report and print only certain categories of records.

For information about creating and printing reports, refer to Chapter 12. In the meantime, follow the steps below to print an existing report:

1. Select **GUESTS** in the **Queries** panel.
2. Choose **Use view** from the prompt box.

   Notice that **GUESTS** appears above the horizontal line in the **Queries** panel and that **REGIONAL** appears above the horizontal line in the **Reports** panel.
3. Highlight the **REGIONAL** report file in the **Reports** panel and press ↵. Choose **Print report** from the prompt box.

   The **Print** menu for reports appears with **Begin printing** highlighted, as shown in Figure 9-3.

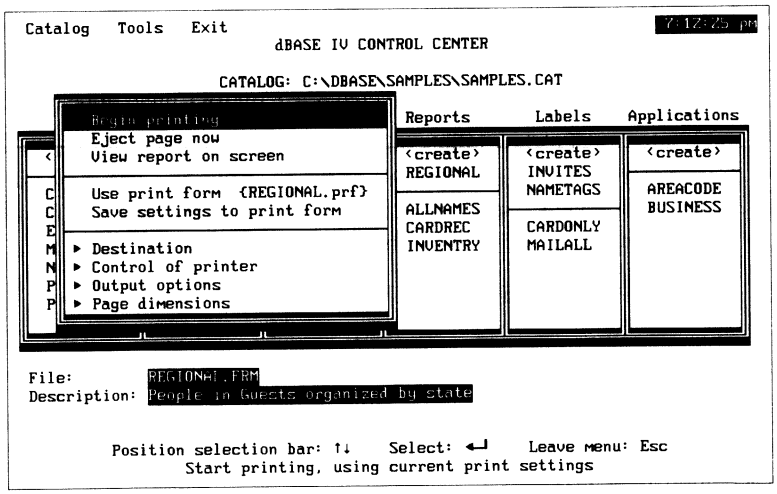


Figure 9-3 **Print** menu for reports

NOTE
To print the Regional report, you must have a printer connected to your computer. If you have not installed your printer for dBASE IV, install it using the DBSETUP program described in Chapter 2 of this manual.

4. If your printer is on and you would like to print the report, select **Begin printing**. Otherwise, select **View report on screen** to display the report on the screen.

   The Regional report uses the Guests view query, which lists the records in the Names database file in alphabetical (ascending) order first by State, and within state by City. The report groups name records by state, prints the number of names for each state, and prints the total number of records in the file.

5. To return to the Control Center without printing or viewing the report, press **Esc**.

Labels

You create a label file to instruct dBASE IV to print labels in a customized format. As with a report, you can design a label to print only the database fields that you want to use, in any order. dBASE IV provides predetermined sizes for labels, or you can specify a custom size.

To work with an example label file, follow the steps below.

1. Select the **INVITES** file from the **Labels** panel.
2. Select the **Print label** option from the prompt box.

   The **Print** menu for labels appears. This menu is similar to that for reports, but has slightly different options.

3. If your printer is on and you would like to print the Invites labels, select **Begin printing**. When printing is completed, the Control Center appears.

   Otherwise, select **View labels on screen** to display the labels on the screen. Notice that the labels are arranged on the page two to a line, in the order dictated by the Guests view query (in ascending order first by State and then by City). Press **Spacebar** to view all labels and go to the Control Center.

   Or, press **Esc** to redisplay the Control Center.

Applications

An application is a group of commands that you use to accomplish a dBASE IV task. An application *program* is usually composed of several simple programs that interact to accomplish the task.

To see how an application works, follow the steps below.

1. Select the **AREACODE** file from the **Applications** panel.

   The Areacode application displays the area code for a given city in the United States.

2. Select **Run application** from the prompt box and then choose the **Yes** option from the confirmation box.

Messages are displayed while the application is being compiled for the first time. The main screen of the application, shown in Figure 9-4, appears.

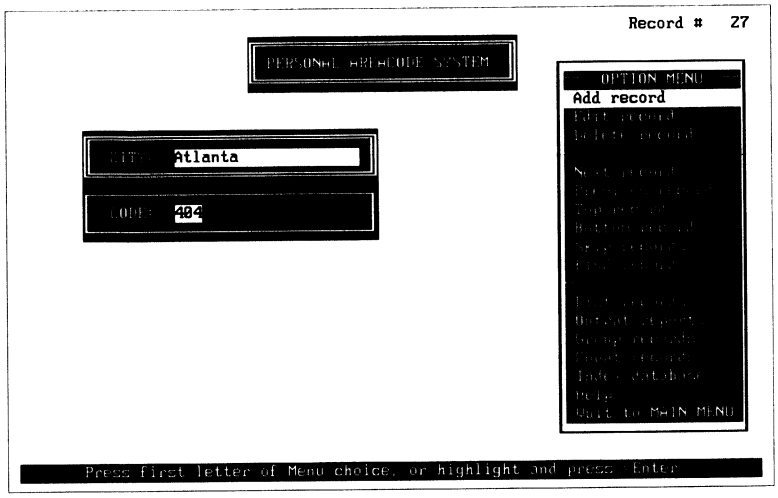


Figure 9-4 Areacode application main screen

3. Type F to select **Find record**.
4. Type Nashville and press ↵.

   The area code for Nashville, 615, appears.
5. If you like, display area codes for other cities. Notice that you must enter a city name with an initial capital, and that the application does not provide information for all cities.
6. When you are ready to return to the Control Center, type q.

   The application terminates and you are back at the Control Center.

For more information about running applications from the Control Center, refer to *Using dBASE IV*, Chapter 15. For information about creating your own applications through an easy-to-use menu system, refer to the Using the Applications Generator section of *Using dBASE IV*.

Before going on to the next chapter, take a break and leave dBASE IV. Select **Quit to DOS** from the **Exit** menu.

Summary

In this chapter, you explored the Control Center and the **Queries**, **Forms**, **Reports**, **Labels**, and **Applications** panels. In later chapters, you'll learn more about using dBASE IV to manipulate the data in your files.

Some of the points covered in this chapter are summarized below.

To navigate between panels:

Press ← or → to move the highlight.

To create and use files in a panel:

To create a new file, highlight the **<create>** marker and press ↵.

To select an existing file, press ↓ or ↑ to move the highlight to the filename and press ↵.

To choose an action from the prompt box:

Press ← or → to move the highlight to the action title and press ↵. Or, type the first initial of the action title.

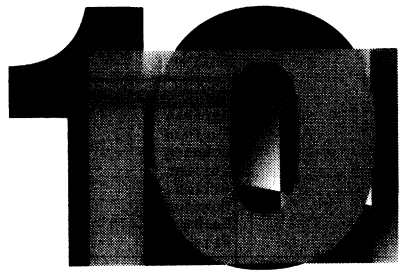
To bypass the prompt box:

Highlight the filename and press:

F2 Data to display data using the file.

Shift-F2 Design to display the design screen for the file.

Entering and Editing Data



In Chapter 7, you were introduced to entering, editing, and adding data in dBASE IV database files. This chapter expands your knowledge of entering and editing data. You will:

- Display data in a database file
- Add new records to the file
- Update file data and undo changes
- Enter and edit text in memo fields
- Remove records from a database file
- Copy data from one database file to another
- Add data from one database file to another

Preparing for This Chapter

If the Control Center with the sample files is not already displayed, type dbsample at the operating system prompt.

If you have a problem displaying the Control Center and the sample files, refer to the instructions in the Displaying the Control Center section of Chapter 7.

Displaying Data in a Database File

Before you can add or edit database records from the Control Center, you must display the file data. To view the records in the Names database file:

1. Highlight **NAMES** in the **Data** panel.
2. Press **F2 Data**.

The Names records appear in a Browse table, as shown in Figure 10-1. Use the **Tab** key to move forward from field to field. Press **Shift-Tab** to move backward from field to field.

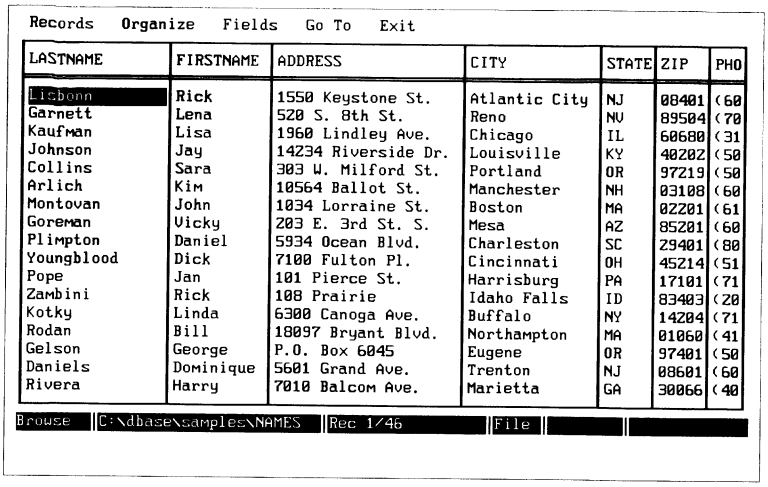


Figure 10-1 Browse table for the Names file

Adding New Records to a File

You can enter new records using either a Browse or Edit screen. Because the Edit screen allows you to concentrate on one record at a time, use it now to enter records.

1. Press **F2 Data**.

   The first record in the Names file is displayed in an Edit screen form, as shown in Figure 10-2.

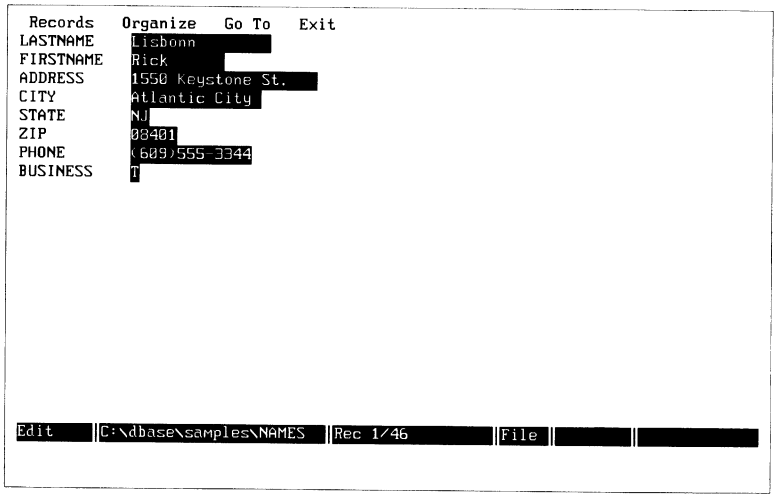


Figure 10-2 Edit screen for the Names file

Because the Browse and Edit screens are just different ways of viewing the same data, they share the same menus, except for the **Fields** menu of the Browse screen. The **Fields** menu contains viewing options that pertain only to Browse.

2. From the **Records** menu, select the **Add new records** option.

   This option presents a blank record, ready to be filled in. The cursor is situated in the first field.

3. Type in the following information:

   Mellon
   Catherine
   1305 Sherman Ave.
   Pittsburgh
   PA
   15212
   (412) 322-4282
   T

   Press ↵ to progress from one field to the next. If you make a mistake while typing, use the arrow keys or **Backspace** to edit the error. If the data you enter completely fills a field, dBASE IV beeps and automatically moves the cursor to the next field; after you've entered data for the last field, dBASE IV displays a new blank record form.

4. Press **PgUp** to view the completed record shown in Figure 10-3.

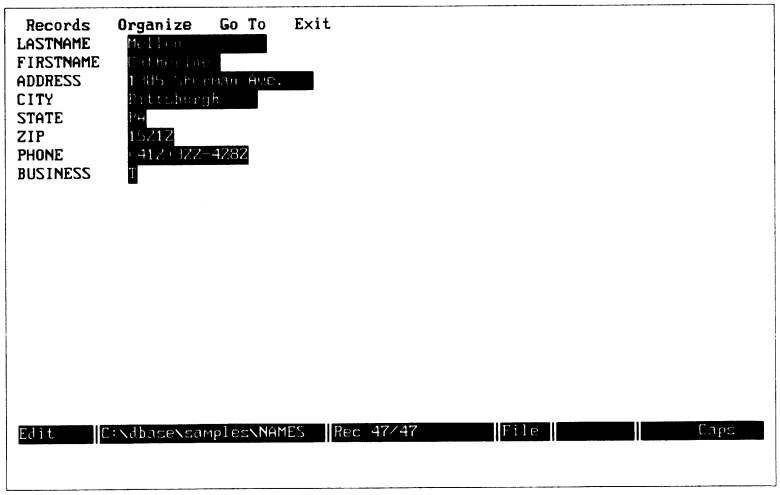


Figure 10-3 Catherine Mellon's record

You do not need to enter data in every field. When entering a new record in the Names file, for example, you may not always have all of the information that you need, such as a telephone number. In other database files, not every field may be relevant to a given record, for example, a telephone extension. You can leave any field blank or add information later.

If database records contain identical information, you can save time by copying field information from one record to another.

5. Press **PgDn** and type Y in response to the prompt at the bottom of the screen to display a blank record form.
6. In the blank record that is displayed, type Mellon in the Lastname field and George in the Firstname field.
7. With the cursor in the Address field, press **Shift-F8 Ditto**.

   dBASE IV copies the data from the same field in the previous record to this one.
8. Use this technique to enter information for the remaining fields.
9. When you have completed the record, press **Ctrl-End**.

   Ctrl-End performs the same function as the **Exit** option of the **Exit** menu. It saves all records entered up to that point and redisplays the Control Center.

Editing Existing Data

Because life involves change, you will periodically be updating many of your database files. You can change outdated data using either the Browse or Edit screen.

Updating a Database

Use the Browse screen to update records in the Names database file.

1. Select **NAMES** from the **Data** panel and press **F2 Data**.

   The last record that you entered appears in an Edit screen form, because that is how you last displayed data.
2. Press **F2 Data** to view data on the Browse screen.
3. Select **Top record** from the **Go To** menu.

   The cursor moves to the top record.

   Sara Collins has moved across town. Her new address is *456 Oak St.* To update Sara's address:
4. Press ↓ to move the cursor to Sara's record.

5. Press **Tab** to move the highlight to Sara's old street address in the Address field and type 456 Oak St.

   The address field now looks like this:

   456 Oak St.ord St.

6. Press **Del** to remove the remains of the old address.

Using the Ins Key

In the last exercise, you were able to type Sara's new address right over the old one. The new characters that you typed replaced the existing characters at the cursor position. This is because Insert mode, by default, is turned off in dBASE IV.

You can also enter field information with Insert mode on. With Insert on, the characters that you type push existing characters at the cursor position to the right. This is useful if you want to type one or more characters between two existing characters.

For example, suppose Sara's new address were 4536 Oak St. rather than 456 Oak St.

1. Press **Ins**.

   Notice that Ins appears on the status bar at the far right.

2. Press ← to move the cursor under the *6* of *456*.

3. Type *3*.

   The *6* moves to the right and the *3* is inserted.

4. Press **Ins** again to set Insert off.

   Ins disappears from the status bar.

Blanking a Field

You can also clear the information in a field to make way for new information. Suppose Sara moved out of Portland altogether, to Eugene:

1. Press **Tab** to highlight the City field in Sara's record.

2. Select **Blank field** from the **Fields** menu.

   Portland disappears.

3. Type Eugene.

TIP
To blank a field quickly, press **Ctrl**-Y *to delete from the cursor position to the end of the field. To blank all of the fields in a record, select* **Blank record** *from the* **Records** *menu.*

Undoing a Change to Data

If you edit a record by mistake, you can restore the original data as long as you discover your mistake before leaving the record.

For example, suppose that you're editing Sara's record when she telephones and tells you that she's not moving anywhere — not to Oak Street in Portland, and certainly not to Eugene:

1. With the cursor on any field in Sara's record, select **Undo change to record** from the **Records** menu.

   Sara is back on West Milford Street in Portland. The undo feature works only for the record that you're currently editing. Once you move on to another record, you can no longer undo changes to a previous record.

2. Select **Exit** from the **Exit** menu to return to the Control Center.

Entering and Editing Memo Text

dBASE IV provides a field type, the *memo field*, for use in entering blocks of text that would not conveniently fit in an ordinary character type field. You can use a memo field to enter an explanatory note about a record or for any other purpose.

Memo fields are not stored with the database file to which they belong. They are stored in a separate file (with a .dbt extension) that is opened automatically along with the database file.

Opening and Closing Memo Fields

To discover how a memo field works, display the Contents database file:

1. Highlight **CONTENTS** in the **Data** panel and press **F2 Data**.

   The Contents file catalogs household possessions.

2. Press **End** to display the Descript field.

   This field contains the word *memo*. If this *memo marker* appears in all capitals, the field contains text. If the memo marker appears in lowercase, the field is empty.

3. Press ↓ to highlight the Descript field of record 8 (on the status bar at the bottom of the screen, the **Rec** indicator reads **8/20**).

   Here the memo marker is in capitals (**MEMO**).

4. Press **F9 Zoom**.

   Text describing the painting titled *Jumped* appears.

5. Press **F9 Zoom** to close the memo field.

Editing Text in a Memo Field

Use the following keys to edit text in a memo field:

- Arrow keys — to move the cursor from character to character or from line to line. **Ctrl-→** moves the cursor to the next word to the right, **Ctrl-←** to the next word on the left.
- **Backspace** — with Insert on, erases the character to the left of the cursor and closes up the space; with Insert off, erases the same character without closing up the space.
- **Del** — erases the character at the cursor position and closes up the space.

Now, edit a memo field:

1. Press ↓ to highlight the memo field for record 9.
2. Press **Ctrl-Home** to open the memo field.

   The memo field shown in Figure 10-4 appears. This field describes the painting *Hummer's Father.*

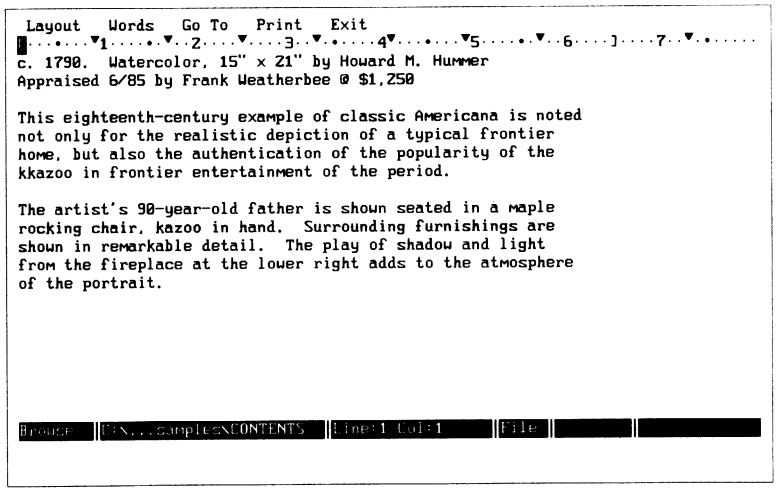


Figure 10-4 Open memo field

3. Make sure that Insert is off. Move the cursor to one of the *K*'s in the word kkazoo in the last sentence of the second paragraph.
4. Press **Del** to delete the character.

   Notice that the space closes up automatically.
5. On the fourth line of the third paragraph, move the cursor to the *l* in **lower**.
6. Type upp.

   Because Insert is off, these characters replace the existing characters.

7. Press **Ins** to turn Insert on.
8. Move the cursor to the word portrait in the last line of text.
9. Type the word rugged and add a space after it.

   With Insert on, the existing characters are pushed to the right.
10. Press **Ctrl-End** to save your changes and close the memo field.

    The memo field closes and the cursor appears at the memo marker.

Adding Text to a Memo Field

You can use the same keystrokes that you used in editing to enter text in an empty memo field.

1. Highlight the first Descript field in Contents (**Rec 1/20**) and press **Ctrl-Home**.

   An empty memo field screen appears for Stomper's Hits #2.
2. Type the following text:

   The title of this CD speaks for itself. This is what you play when you're dead tired and bored stiff, and wish everybody would go home. They go.
   NOTE: There was no Stomper's Hits #1.
3. Press **Ctrl-End** to save your changes and close the memo field.

NOTE
To close a memo field without saving your changes, press **Esc**.

4. Select **Exit** from the **Exit** menu and return to the Control Center.

Removing Records from a Database File

Database records are continually becoming obsolete as employees leave the company, clients move out of town, and product lines are discontinued. You remove an obsolete record from a database file in two stages:

- Mark the record for deletion.
- Instruct dBASE IV to delete it.

Marking Records for Deletion

You can mark records for deletion using either the Browse or Edit screen.

1. Highlight in the **Data** panel and press **F2 Data**. (If the Edit screen appears, press **F2 Data** to display the Browse screen.)
2. Highlight record 7 (**Rec 7/48** on the status bar).
3. Select **Mark record for deletion** from the **Records** menu.

   Del appears on the status bar to indicate that this record is marked for deletion.
4. Highlight record 29 by selecting **Record number** from the **Go To** menu and typing 29 at the prompt. (Make sure that Insert mode is turned off, press ←, and type over the current record number, 7.)
5. Mark this record for deletion as in step 3.
6. Highlight record 19 as in step 4 and mark it for deletion.

   Again, **Del** appears in the status bar, as shown in Figure 10-5.

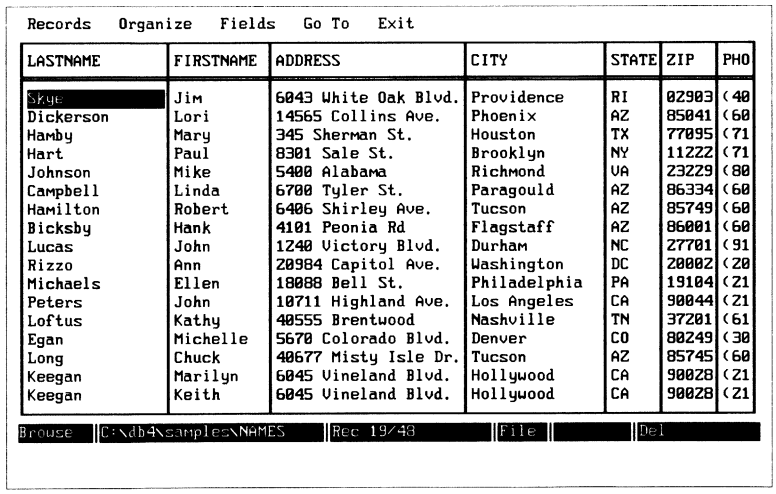


Figure 10-5 Record marked for deletion

Unmarking Marked Records

To remove a mark from a record that you have marked for deletion:

1. Highlight record 29, which you marked for deletion (look for the **Del** indicator in the status bar).
2. Select **Clear deletion mark** from the **Records** menu.

   Del disappears from the status bar, indicating that record 29 is no longer marked for deletion.

NOTE
The **Clear deletion mark** *option replaces* **Mark record for deletion** *when you have highlighted a record that is marked for deletion.*

TIP
To quickly mark and unmark a highlighted record, press **Ctrl**-U.

3. Select **Exit** from the **Exit** menu to preserve the deletion marks on the two records that you have marked (7 and 19) and return to the Control Center.

Erasing Marked Records

Use the **Organize** menu on the Browse, Edit, or database design screen to erase the records that you marked in the preceding exercise. Erasing marked records allows you to *pack* the database so that it takes less storage space.

WARNING
Although you can change your mind after marking a record for deletion, once you've erased marked records, they cannot be recovered.

---

1. Highlight **NAMES** in the **Data** panel and press **Shift-F2 Design**.

   The database design screen appears with the **Organize** menu open. Notice that you can reverse your decision to erase marked records by selecting **Unmark all records** from the menu.

2. Select **Erase marked records** from the **Organize** menu.

   dBASE IV prompts you to confirm that you really do want to erase the marked records.

3. Select **YES**.

   The marked records are permanently removed from the database file. Messages tell you the number of records that were retained in the file and that dBASE IV has rebuilt all of the indexes associated with the file.

4. Select **Save changes and exit** from the **Exit** menu. Press ↵ once more to confirm your deletion and return to the Control Center.

5. Highlight **Names** in the **Data** panel, press ↵, and select **Close file** to close the Names file.

Copying Data

You can use the copy capabilities of the dBASE IV menu system in two different ways:

- To copy data from one file to another
- To add data to a database file from another database file

Copying a dBASE IV File

Being able to copy database files is important for creating *backups* — copies of original files that you can use if the data in the original files is accidentally lost. When backing up a database file, you can also back up associated index (.mdx) and memo field (.dbt) files.

To copy the Contents database file, its index (.mdx) file, and its memo (.dbt) file to a backup disk:

1. Insert the formatted backup disk in one of your computer's disk drives.
2. Highlight **CONTENTS** in the **Data** panel and press **Shift-F2 Design**.
3. From the **Layout** menu, select **Save this database file structure**.

   A **Save as**: prompt box appears, containing the current path information for Contents.
4. Edit this information for the drive that contains the backup disk (for example, **A:\Contents.dbf**) and press ↵.

   dBASE IV briefly displays statistics as it backs up the files.
5. Press **Ctrl-End** to return to the Control Center.

NOTE
Besides this method, you can copy a dBASE IV file using the:

- *COPY TO command at the dot prompt (refer to Chapter 14 of this manual).*
- *DOS COPY command (refer to your DOS manual).*
- **Copy** *option of the* **Operations** *menu (under the* **DOS utilities** *option of the* **Tools** *menu of the Control Center; refer to Chapter 14 of* Using dBASE IV*).*

Adding Data from Another dBASE IV File

Adding (*appending*) data to a dBASE IV database file from another database file is particularly useful for consolidating data from a number of sources. For example, suppose that you want to inventory all of the office equipment owned by your company.

To do this, you could:

- Design and create a database file containing the necessary fields.
- Copy the file to floppy disks and distribute a disk to each of your department managers.
- Have each manager enter information for department equipment in the database file and return the floppy disk to you.
- Append all of the files to create a master database of office equipment.

NOTE
When appending data from one database file to another, both files must have similar structures, that is, fields with the same names, sizes, and data types. Only data from fields whose names match those in the active database file is copied.

To append records from the database file you created in Chapter 7, Mynames, to the Names database file, do the following:

1. Highlight **MYNAMES** in the **Data** panel and press **Shift-F2 Design**.
2. Select **Append records from dBASE file** from the **Append** menu.
3. From the list that appears, select Names.dbf.

   The records in the Names file are appended to the Mynames file.
4. Press **F2 Data** to display the data in Mynames.

   The last record copied from Names is displayed.
5. Press **PgUp** to view the other Names records and the original Mynames records.
6. Press **Ctrl-End** to exit from Browse and to save the appended records.
7. Highlight **MYNAMES** in the **Data** panel, press ↵, and select **Close file** to close the Mynames file.

Summary

This chapter explored some of the methods that you can use to change data in a database file. These methods are summarized below.

To display the data in a database file:

Highlight the filename in the **Data** panel and press **F2 Data**.

To add a new record to a file:

Display the data in the file and select **Add new records** from the **Records** menu. Enter information in the blank record that appears.

To copy information from a field to the same field in the next record:

Move the cursor to the new field and press **Shift-F8 Ditto**.

To change existing data:

Move the cursor to the record to be edited and type new field information over the old. Press **Del** to remove the remains of the old data. Press **Ins** to add one or more characters between existing characters.

To remove information from a highlighted field to make room for new information, select **Blank field** from the **Fields** menu or press **Ctrl**-Y. To blank all fields in the current record, select **Blank record** from the **Records** menu.

To restore the original data in the current record:

Before moving to another record, select **Undo change to record** from the **Records** menu.

To open and close a memo field:

With the cursor on the memo marker, press **F9 Zoom** to open the memo field. Press **F9 Zoom** to close the memo field. In the memo marker, MEMO means that the memo field contains data; **memo** means that the field is empty.

To enter and edit memo field data:

Use the arrow keys, **Backspace**, and **Del** to enter and edit data. Use **Ins** to push existing data to the right rather than type over it.

To mark records for deletion:

Move the cursor to each record that you want to mark and select Mark **record for deletion** from the **Records** menu, or press **Ctrl**-U.

To unmark marked records:

Move the cursor to the record that you want to unmark and select **Clear deletion mark** from the **Records** menu, or press **Ctrl**-U. To unmark all records that are marked for deletion, select **Unmark all records** from the **Organize** menu.

To erase marked records:

Select **Erase marked records** from the **Organize** menu. Select **YES** from the confirmation prompt box.

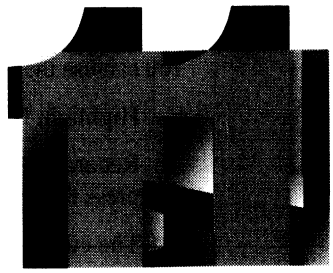
To make a backup copy of a database file:

Highlight the filename in the **Data** panel and press **Shift-F2 Design**. Select **Save this database file structure** from the **Layout** menu, enter path information for the drive and directory to which you want to back up, and press ↵.

To append data from another dBASE IV file:

Highlight the name of the file to which you want to append data in the **Data** panel of the Control Center, press **Shift-F2 Design**, and select **Append records from dBASE file** from the **Append** menu. From the list that appears, select the name of the file from which you want to copy records.

Finding and Arranging Data



dBASE IV database files let you store large amounts of data. As your files get larger, you will need more efficient ways of organizing and locating their data. In this chapter, you will learn how to:

- Find records
- Create indexes for organizing database records in various ways
- Create a sorted version of a database file

Preparing for This Chapter

If the Control Center with the sample files is not already displayed, type dbsample at the operating system prompt.

If you have a problem displaying the Control Center and the sample files, refer to the instructions in the Displaying the Control Center section of Chapter 7.

Finding Records

When a database file is small, you can easily use the arrow keys to move to specific records. As a file grows larger, however, you need more efficient ways of locating data.

dBASE IV provides the following methods to help you find file data quickly:

- Paging up and down
- Locating the first and last records
- Locating a record by number
- Locating a record by its field data

Paging Up and Down

You can use the **PgUp** and **PgDn** keys to scan quickly through a file, one screen at a time.

1. Highlight **NAMES** in the **Data** panel and press **F2 Data**.

   Records in the Names database file are displayed on the Browse screen. (If not, press **F2 Data** once again.)

   The cursor is in the first record of the file. You know this because the *record pointer*, **Rec 1/46** on the status bar, tells you so.

   The record pointer consists of two numbers separated by a slash (/). The first number indicates the current record's position in the file (in this case, 1/) and the second number shows the total number of records in the file.

2. Press **PgDn**.

   The next screen of records is displayed. If you have a standard-size screen, the cursor appears in the record for John Orlando (**Rec 17/46**), which was the last record on the first screen.

3. Press **PgDn** once more.

   The record for Keith Keegan is displayed. Again, if you have a standard-size screen, this record was the last record on the previous screen.

4. Press **PgDn** until the cursor is in the last record of the file.

Locating the First and Last Records

The **Go To** menu, on the Browse or Edit screen, has a number of options for finding specific records. The first two on the menu, shown in Figure 11-1, are **Top record** and **Last record**.

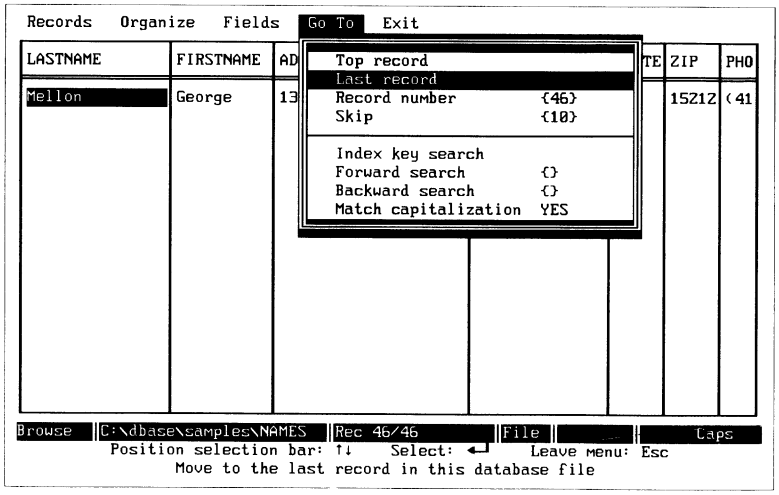


Figure 11-1 **Go To** menu

1. With the cursor in the last record of Names, select **Top record** from the **Go To** menu.

   The current record is now the first record of the file (1/46).

2. Select **Last record** from the **Go To** menu to move the cursor to the last record in Names.

A quicker alternative to selecting options from the **Go To** menu is to press **Ctrl-PgUp** to move to the first record of a database, and **Ctrl-PgDn** to move to the last record.

Locating a Record by Number

The **Go To** menu also lets you locate a record by record number. Every database record has a number that indicates its location.

1. Select **Record number** from the **Go To** menu.
2. Press **Backspace** to erase the number of the current record (46) in the prompt box, type 18, and press ↵.

   Record 18 (**Rec 18/46**) is now the current record, where the cursor and the highlight are located.

3. Select **Record number** from the **Go To** menu, type 1 in the prompt box, and press ↵.

   The current record is now the first record of the file.

NOTE
A record number indicates the order in which a record was originally entered in a file. As other records are added and deleted and as the file is indexed, a record's numeric position in the file may change.

Locating a Record by Field Data

Because you probably won't remember the record number of every record in a file, dBASE IV provides other options on the **Go To** menu. These options let you locate a record by specifying the data in any of its fields:

- **Forward search** — looks for the specified data in subsequent records
- **Backward search** — looks for the specified data in preceding records
- **Match capitalization** — if **YES**, looks for the specified character pattern with the same combination of upper- and lowercase letters; if **NO**, merely looks for the specified pattern in any combination of upper- and lowercase letters.

NOTE
The **Index key search** *option is shown on the menu in gray to indicate that it is not available, because the Names file is not currently ordered by any of its indexes. This option lets you search for a record using the value of an index key, as discussed later in this chapter.*

1. Position the cursor in the Lastname field of the first record.
2. Select **Forward search** from the **Go To** menu, type johnson in the prompt box, and press ↵.

   dBASE IV beeps and displays a ****Not Found**** box.
3. Press any key and again select **Forward search.**

   This time, type Johnson with a capital J.

   dBASE IV finds the first record whose Lastname field contains "Johnson" (for Jay Johnson), because **Match capitalization** is set to **YES.** Now, try a shortcut.
4. Press **Shift-F4 Find Next.**

   Mike Johnson's record is highlighted in a forward search.
5. Press **Shift-F3 Find Previous.**

   Jay Johnson's record is again highlighted in a backward search.

   When doing forward and backward searches, you can use the *wildcard* characters *?* and *. These work the same way as in DOS. The question mark is used to represent a single character and the asterisk can represent any number of characters.

   A search for *cou?t*, for example, would find any record whose current field contained *court* or *count*. Searching for *cou*t* would find records containing *court*, *couplet*, *couchant*, or even *counterpart*.
6. If you like, practice searching for Names records using the values in fields other than Lastname. Try using the *?* and * wildcards. When you are finished, return to the Control Center by selecting **Exit** from the **Exit** menu.

Organizing Data

The search techniques that you've just practiced work well for small databases. However, when a database gets so large that searches become slower, an index key search is faster and more efficient.

This section introduces you to indexing, which lets you organize large amounts of data in many different ways, depending on your needs. It also discusses the use of sorting to reorganize data. However, sorting has no effect on the speed of your searches.

Indexing

An *index* reorganizes the records in a database file by the values in one or more fields. An index is a simple file that dBASE IV refers to internally when locating and displaying the records in a database file.

Once you organize database records using an index, the records appear in a different order. However, the records in the database file remain stored in the same order in which they were originally entered. Figure 11-2 illustrates the process.

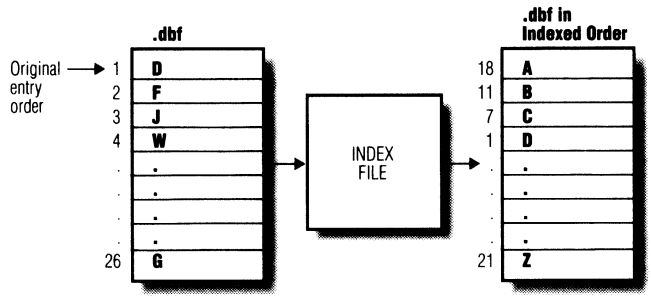


Figure 11-2 Indexing a database file

An index file contains two columns:

- Index *key* — the field values that are used to organize database file records.
- Unique record numbers — pointers to the actual storage location of each record in the database file.

An index for a database file is similar to the index of a book, which lists subject keys and page numbers that point to where the actual information is located in the book.

You can use the values of almost any field or combination of fields in a database as the index *key* or *expression*. For example, you can use the values of Lastname, State, or Zip fields as the key for listing records in the Names database file. You can also use the expression Lastname + Firstname as the key to index records first by Lastname values and, when these values are identical, by Firstname values.

However, you cannot use a logical field such as Business as the index key.

Figure 11-3 illustrates an index file that organizes records in the Names database file by the values in the Lastname field.

Index Key Data	Record Number
Arlich	6
Beman	39
Bicksby	26
Brendon	36
Campbell	24
Cohen	42
Collins	5
Daniels	16
DeBello	43

Figure 11-3 Index file on Names

Creating and using an index has the following advantages over sorting:

- Because an index file contains only the values of one or more fields of a database file rather than entire records, an index takes up less disk space than a sorted database file.
- dBASE IV can use an index to organize a database file faster than it can create a new, sorted file.
- You can create a number of indexes for organizing a database file without storing a lot of redundant data.
- Maintaining data in one indexed file rather than in several sorted files promotes data integrity because there is only one version of the data, which doesn't need to be updated in more than one file.

In Chapter 7, you used an existing index on the City field to display the Names file. Now, you'll learn how to create a new index.

Creating an Index

You can index a file on any type of database field except logical or memo fields. In this section, you will index the Names file by the values in the State field and exclude records for acquaintances in California.

1. Highlight **NAMES** in the **Data** panel and press **F2 Data.**

   The Browse screen appears.

2. Select **Create new index** from the **Organize** menu.

   The submenu shown in Figure 11-4 is displayed.

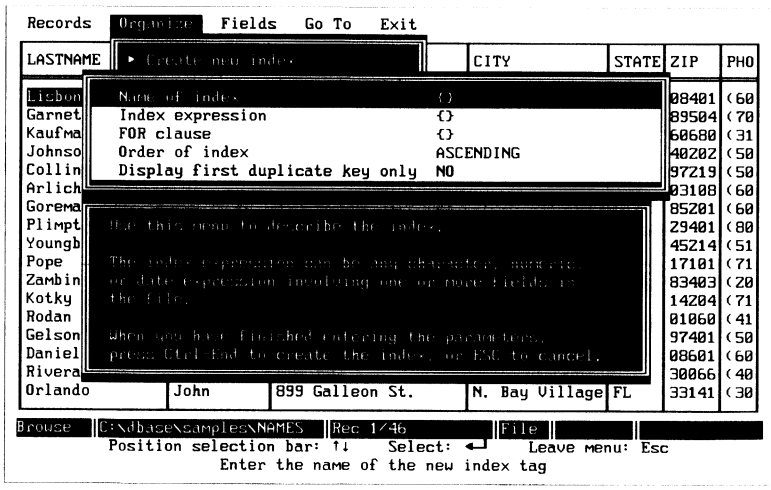


Figure 11-4 Index creation submenu

3. Select the **Name of index** option from the submenu.

   It is a good idea to give your index the same name as its index expression.

4. Type state and press ↵.

   The cursor moves to the next option, **Index expression**.

5. Press ↵ to select **Index expression**.

6. Press **Shift-F1 Pick**.

   A three-column list appears:

   - **Fieldname** lets you choose the name of the field whose values are to compose the index expression.
   - **Operator** lists the arithmetic and logical operators that you can use to build an index expression on more than one field.
   - **Function** lists functions that you can use to create a complex index.

7. Select **STATE** in the **Fieldname** column.

   The list disappears and **STATE** is entered in the **Index expression** option. (If you know the field's name, you can type it directly in step 6 instead of picking from a list.)

8. Press ↵ to move the cursor to the **FOR clause** option. This option allows you to specify the records that are to be included in the index. Pressing ↓ to bypass this option and move the cursor to the next menu option would include all records in the new index.

9. Press ↵ to select **FOR clause** and press **Shift-F1 Pick**. The same list displayed in step 6 appears.

10. Select **STATE** in the **Fieldname** column; press **Shift-F1 Pick** to redisplay the list and select <> (not equal to) in the **Operator** column (press ↓ to display the <> operator); type "CA" (use capital letters and include quotation marks); and press ↵ to highlight the next menu option.

    The next option asks you to specify the order of the index. **ASCENDING** (lowest to highest) is the default. Pressing **Spacebar** or ↵ toggles between **ASCENDING** and **DESCENDING** (highest to lowest) order.

11. Specify **ASCENDING**.

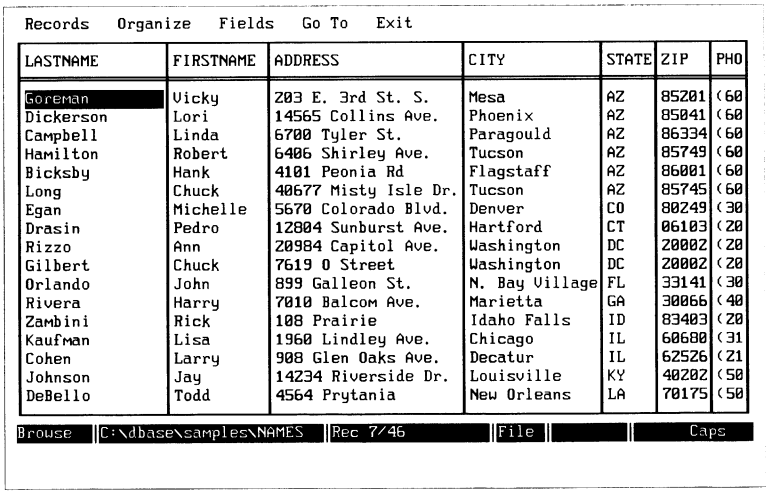
12. Ignore the next option, **Display first duplicate key only**, which is explained in the next section. Press **Ctrl-End** to create the new index.

    Messages are displayed as the new index is created and the Names records are reorganized in alphabetical order by state. Records that contain *CA* in the State column are not displayed.

Unique Indexes

A *unique index* includes only the first occurrence of a record that contains a particular index expression. Therefore, you use a unique index to remove duplicate records from your file.

For example, six of the people listed in the Names database file live in Arizona. The State index that you just created includes them all, as shown in Figure 11-5.



Records Organize Fields Go To Exit

LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	PHO
Foreman	Vicky	203 E. 3rd St. S.	Mesa	AZ	85201	(60
Dickerson	Lori	14565 Collins Ave.	Phoenix	AZ	85041	(60
Campbell	Linda	6700 Tyler St.	Paragould	AZ	86334	(60
Hamilton	Robert	6406 Shirley Ave.	Tucson	AZ	85749	(60
Bicksby	Hank	4101 Peonia Rd	Flagstaff	AZ	86001	(60
Long	Chuck	40677 Misty Isle Dr.	Tucson	AZ	85745	(60
Egan	Michelle	5670 Colorado Blvd.	Denver	CO	80249	(30
Drasin	Pedro	12804 Sunburst Ave.	Hartford	CT	06103	(20
Rizzo	Ann	20984 Capitol Ave.	Washington	DC	20002	(20
Gilbert	Chuck	7619 0 Street	Washington	DC	20002	(20
Orlando	John	899 Galleon St.	N. Bay Village	FL	33141	(30
Rivera	Harry	7010 Balcom Ave.	Marietta	GA	30066	(40
Zambini	Rick	108 Prairie	Idaho Falls	ID	83403	(20
Kaufman	Lisa	1960 Lindley Ave.	Chicago	IL	60680	(31
Cohen	Larry	908 Glen Oaks Ave.	Decatur	IL	62526	(21
Johnson	Jay	14234 Riverside Dr.	Louisville	KY	40202	(50
DeBello	Todd	4564 Prytania	New Orleans	LA	70175	(50

Browse | C:\dbase\samples\NAMES | Rec 7/46 | File | Caps

Figure 11-5 Nonunique State index

To see only the first record for each state represented by Names records (excluding California), change the State index to a unique index.

1. Select **Modify existing index** from the **Organize** menu.

   A list of indexes is displayed. The index expression of the highlighted index appears to the left of the list.

2. Select **STATE**.

   The same submenu that was displayed for **Create new index** appears with the options that you defined for the State index.

3. Highlight **Display first duplicate key only** and press **Spacebar** to change the default to **YES**.

4. Press **Ctrl-End** to reorganize Names records according to the unique State index.

   Notice that only one record is displayed for each state, as shown in Figure 11-6. Once dBASE IV locates the first record whose State field contains a given abbreviation, it ignores all subsequent records that contain the same abbreviation.

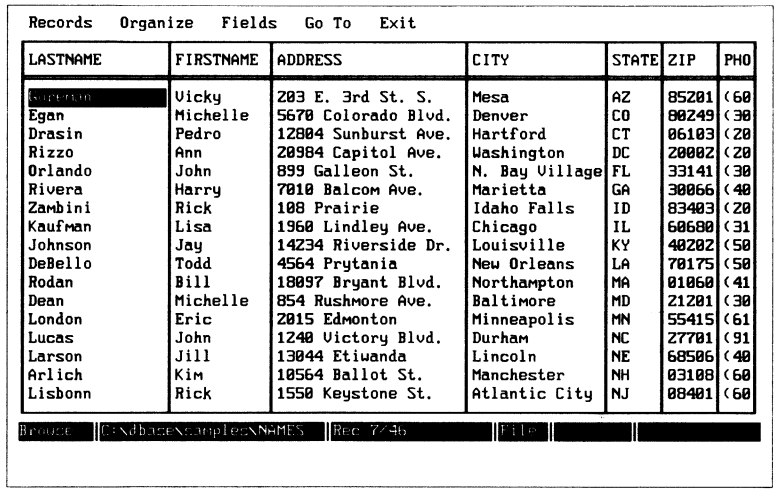


Figure 11-6 Unique State index

Indexing on More than One Field

When you use more than one field in an index, records are ordered by values in the first field. Then, generally, for each identical value in the first field, records are ordered by the values in the second field. This process is repeated for additional fields.

For example, if Names were indexed using the index expression *Lastname + Firstname*, records would be ordered by Lastname values. Records containing duplicate values for Lastname would then be ordered by Firstname values.

1. Select **Create new index** from the **Organize** menu.

2. Press ↵ on the **Name of index** option, type mail, and press ↵.

3. Press ↵ on the **Index expression** option, type state + lastname + firstname, and press ↵.

   The other options are as you want them, so you can leave them alone.

4. Press **Ctrl-End** to reorganize Names using the new index.

   dBASE IV reorders the records alphabetically, first by State, within State by Lastname, and within Lastname by Firstname (see the entries for Keith and Marilyn Keegan).

Sorting

When you *sort* a database file, you create a new file with the same records, physically rearranged according to the values in one or more fields. The original file is left unchanged.

Indexing is generally faster and more efficient than sorting. However, sorting enables you to arrange records in ascending order by the values in one field and in descending order by the values in another, as shown in the next exercise. Indexing can be used to order records in either ascending or descending order by the values in one or more fields, but not in both.

To create a new file from the data in Names, with records in descending order by State and in ascending order by Lastname, follow the steps below:

1. Select **Sort database on field list** from the **Organize** menu.

   The box shown in Figure 11-7 is displayed. You enter the names of the fields that you want to use for sorting in the **Field order** box and the sort order for each in the **Type of sort** box.

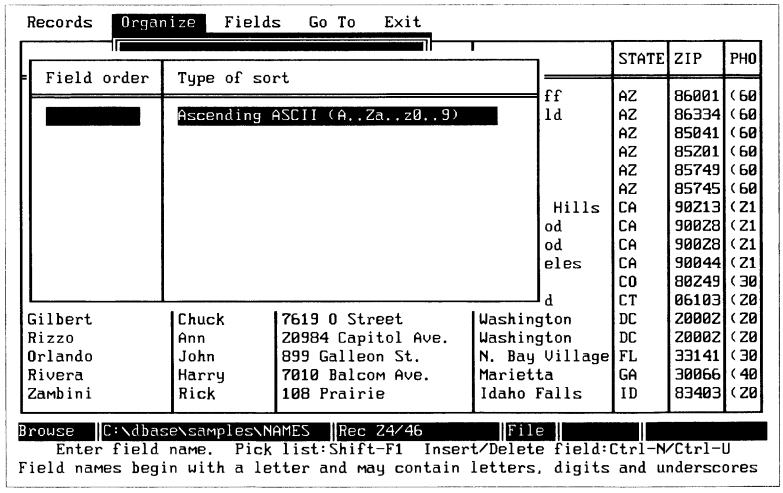


Figure 11-7 Entering sort information

2. Type state in the **Field order** box and press ↵.

   The cursor moves to the **Type of sort** box, with **Ascending ASCII** as the default order.

3. Press **Spacebar** to change the order to **Descending ASCII**.

4. Press ↵ to move the cursor back to the **Field order** box of the next line and press **Shift-F1 Pick**.

   A list of the available sorting fields appears.

5. Highlight **LASTNAME** and press ↵.

   Lastname is displayed in the **Field order** box.

6. Press **Ctrl-End** to conclude your sort instructions.

   dBASE IV asks you to name the new, rewritten file.

7. Type namesrt and press ↵.

   A box prompting you to edit the description of the new database file appears.

8. Type Names sorted by STATE (DESC) and LASTNAME (ASC) and press ↵.

9. Select **Exit** from the **Exit** menu to return to the Control Center.

10. Highlight **NAMESRT** in the **Data** panel and press **F2 Data**.

    The Names records are displayed as sorted in Namesrt.

11. Select **Exit** from the **Exit** menu to return to the Control Center.

Summary

This chapter showed you how to locate, index, and sort data in a database file. The basic procedures are summarized below.

To move up or down through a file, one page at a time:

Press **PgUp** or **PgDn**. The record pointer on the status bar lets you know the cursor position.

To display the first or last record in a database file:

Select **Top record** or **Last record** from the **Go To** menu, or press **Ctrl-PgUp** or **Ctrl-PgDn**.

To move to a specific record:

Select **Record number** from the **Go To** menu, type the number of the record that you want over the current record number, and press ↵.

To search for a record by its field data:

Position the cursor on the field that you want to search. Choose **Forward search** or **Backward search** from the **Go To** menu and type the data for which you want to search. (If **Match capitalization** is **YES**, make sure that you type the correct combination of uppercase and lowercase letters.) Press ↵.

Press **Shift-F4 Find Previous** or **Shift-F3 Find Next** to find other records that contain the same data.

To create an index:

Select the file to be indexed from the Control Center **Data** panel and press **F2 Data** or **Shift-F2 Design**. Select **Create new index** from the **Organize** menu.

Press ↵ to select each menu option, in turn. Specify information for each desired option and press ↵. (To specify **Index expression** and **FOR clause** information, you can press **Shift-F1 Pick** to list field names and operators.) Press **Ctrl-End** to create the index.

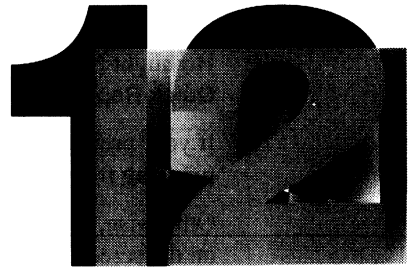
To sort a file:

Select the database file to be sorted from the Control Center **Data** panel and press **F2 Data** or **Shift-F2 Design**. Select **Sort database on field list** from the **Organize** menu.

Type the names of the fields on which you wish to sort in the **Field order** column (if you like, choose each from a list displayed by pressing **Shift-F1 Pick**). For each field, use **Spacebar** to specify sort order in the **Type of sort** column.

Press **Ctrl-End** to conclude your sort instructions, and name and describe the new, sorted file.

Printing in dBASE IV



dBASE IV gives you a wide choice of options for printing your data. For example, you can print:

- Selected fields and records in a database file.
- Labels, letters, and reports using preset formats or formats that you have designed yourself.

In this chapter, you will learn how to:

- Print a Quick Report of the data in a database file
- Create your own customized report
- Use **Print** menu options to print your report
- Save your print settings to a print form file

You will be working with options on the **Print** menu for reports. With the exception of a single option, this menu provides the same options as the **Print** menu for labels. Thus, the principles for printing reports apply equally well to labels.

Preparing for This Chapter

If the Control Center with the sample files is not already displayed, type dbsample at the operating system prompt.

If you have a problem displaying the Control Center and the sample files, refer to the instructions in the Displaying the Control Center section of Chapter 7.

Printing Data Quickly

The Quick Report feature of dBASE IV, which you used in Chapter 7, lets you organize and print data using a view or a database file without having to design a report format. A Quick Report prints data in column format, much as you see it on the screen.

To print a Quick Report of the Names database file:

1. Highlight **NAMES** in the **Data** panel of the Control Center and press **Shift-F9 Quick Report**.
2. If your printer is ready to print and you want to print the report, select **Begin printing** from the **Print** menu.

   Otherwise, select **View report on screen** and press **Spacebar** to scroll the report on the screen.
3. Once the Control Center is redisplayed, select **NAMES** in the **Data** panel and select **Close file** from the prompt box.

NOTE
At any time, you can pause printing temporarily by pressing **Ctrl**-S, *or stop printing entirely by pressing* **Esc**. *If your printer has a large print buffer, turn the printer off to stop printing immediately.*

Creating a Report

In Chapter 9, you used the **Print** menu to print the existing Regional report. Now you will create and print your own report based on the Busphone query that you created in Chapter 9. You will:

- Define the right margin for the report
- Create a report title
- Modify column headings
- Delete the ZIP column from the layout
- Move the PHONE column to the left to close up the space left by ZIP
- Add a page footer

1. Select **BUSPHONE** from the **Queries** panel and select **Use view** from the prompt box.

   BUSPHONE appears above the horizontal line in the **Queries** panel.
2. Highlight the **<create>** marker in the **Reports** panel and press ↵.

   The **Report** design screen appears.
3. Select **Quick layouts** from the **Layout** menu and then **Column layout** from the submenu that appears.

The initial report design screen for the Busphone file appears, as shown in Figure 12-1. The screen contains the following *bands*, which divide the report into horizontal areas:

- Page header band — Lets you enter a report heading. Contains place markers for page number and current date information, and the column headings for the current database or query file.
- Report intro band — Lets you enter a description of the report.
- Detail band — Lets you control the fields that are displayed for each file record and their positions in the report. Each field position is represented by a *template*, a number of X characters.
- Report summary band — Lets you display totals for numeric fields.
- Page footer band — Lets you enter information that is printed at the bottom of each report page, such as time and date of printing, page number, or report title.

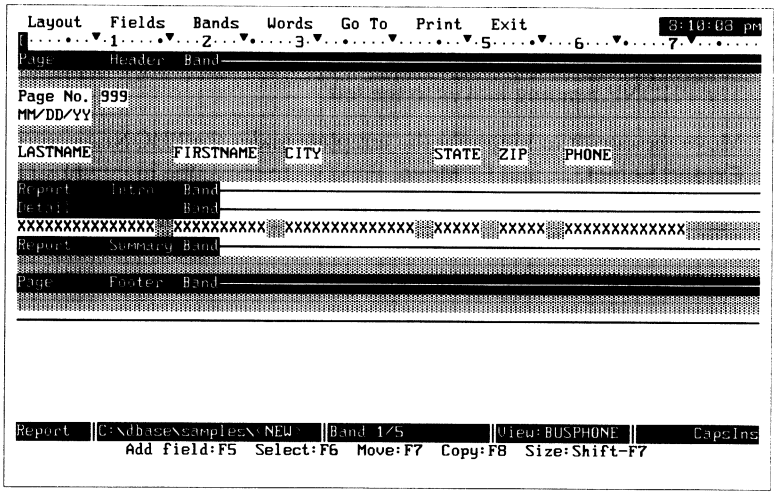


Figure 12-1 Report design screen for Busphone

4. Press **Alt-W** to open the **Words** menu and select **Modify ruler**.

   The cursor moves to the ruler line at the top of the screen.

5. Press **Tab** to move the cursor to column 72 (just below the right-most arrowhead marker), type] to specify the right margin, and press **Ctrl-End** to save the setting.

   The cursor moves back to the page header band.

6. Press ↓ twice to move the cursor to the page number place marker inside the page header band and press **Ctrl-Y** twice to delete this and the date place marker.

7. Press ↑ to move the cursor to the first line of the page header band and type the report title: Phone Numbers for Business Contacts (by State).

8. Press **Alt-W** to open the **Words** menu, select **Position**, and then select **Center** from the submenu.

   The title is centered on the screen.

9. Move the cursor to the N in the LASTNAME column heading, press the **Ins** key (if **Ins** is not displayed at the far right side of the status bar), and press **Spacebar** to add a space between LAST and NAME. Repeat the procedure to add a space between FIRST and NAME in the FIRSTNAME column heading.

10. Press ↓ to move the cursor to the templates in the detail band. Press **Tab** to highlight the template under the ZIP column heading and press **Del** to remove the template.

11. Move the cursor to the Z in the ZIP column heading, press **F6 Extend Select**, press → to highlight all three letters of the heading, and press **Del** to remove the heading.

12. Move the cursor to the P in the PHONE heading. Press **F6 Extend Select**, → to highlight the heading, ↵, and **F7 Move**. Press ← to move the highlight four spaces to the left of the PHONE heading (use the vertical lines as a guide), and press ↵.

    The PHONE heading moves to the left.

13. Move the cursor to highlight the template under the PHONE heading. Press **F6 Extend Select**, ↵, and **F7 Move**. Press ← to move the highlight so that its left edge aligns with the P in the PHONE heading, press ↵, and type Y to respond affirmatively to the prompt at the bottom of the screen.

    The PHONE template moves to the left.

14. Move the cursor to the page footer band and press ↓ to move inside the band.

15. Press **F5 Field**, select **Date** from the **PREDEFINED** column on the menu that appears, and press **Ctrl-End** to accept the default (NO) for the **Suppress repeated values** option on the submenu that appears.

    A place marker for current date appears on the footer line.

16. Move the cursor to the date place marker to highlight it, select **Position** from the **Words** menu, and select the **Left** option from the submenu.

    The current date place marker moves to the left of the footer line.

    The new report format for Busphone is shown in Figure 12-2.

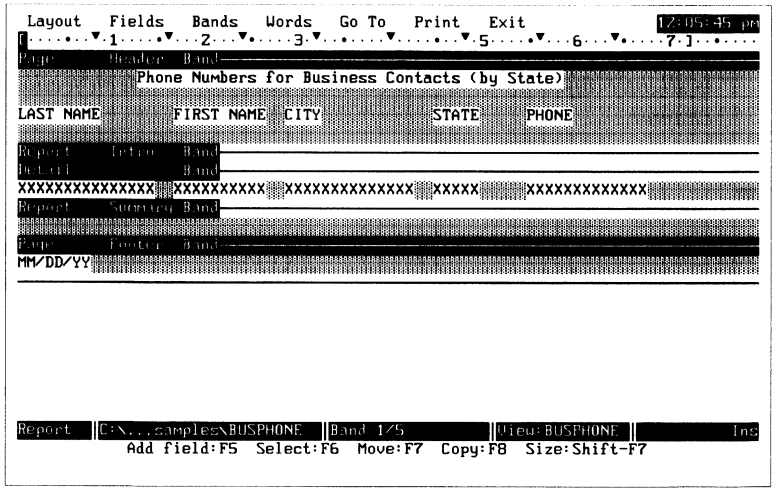


Figure 12-2 Completed report design for Busphone

17. Press **Ctrl-End** to save the report format, type BUSPHONE in the **Save as:** prompt box to name the new format file, and press ↵.

    dBASE IV processes the new report file and redisplays the Control Center, with **BUSPHONE** displayed above the horizontal line in the **Reports** panel.

Printing a Report

In Chapter 9, you used the **Begin printing** option of the **Print** menu, pictured in Figure 12-3, to print a report and labels. Now, take a brief look at the other **Print** menu options.

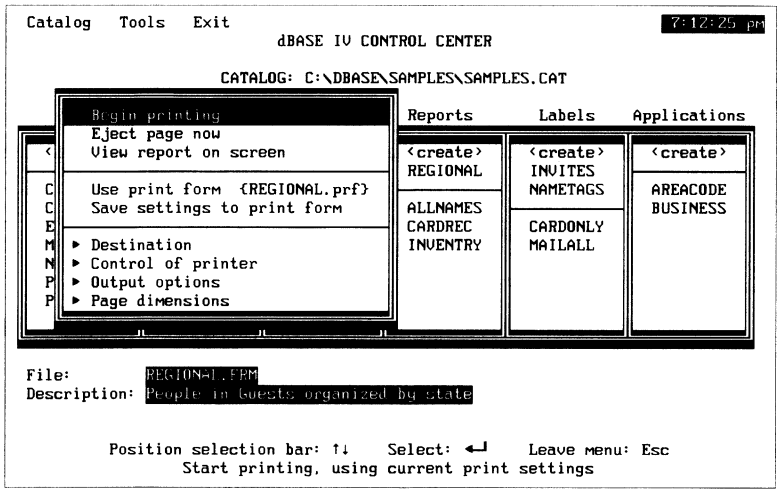


Figure 12-3 **Print** menu

NOTE
Figure 12-3 illustrates the **Print** *menu for reports. The* **Print** *menu for labels is slightly different, as described in the Labels Print Menu section later in this chapter.*

Print Submenus

Each of the last four options on the **Print** menu is preceded by an arrowhead symbol, indicating that it includes submenu options. **Print** submenu options control the following print functions:

- **Destination** — where the print output is sent: either to a printer or to an operating system file for printing later. You can also view the report on the screen as it is printed or saved to a file.
- **Control of printer** — how the printer prints: number of characters per inch, paper feeding and page advances, and starting and ending control codes.
- **Output options** — what is printed: starting and ending pages, page number of first page, and number of copies to be printed. (The ability to change the number of your first page is useful when printing a report that is to be included in another report or in a document.)
- **Page dimensions** — format options: page length, offset from left, and line spacing.

NOTE
To perform the following exercise, you must have a printer connected to your computer. If you have not installed your printer for dBASE IV, install it before performing the next exercise. Refer to Chapter 2 of this manual.

Modifying Print Settings

In this section, you'll modify some of the print settings for a report and save the new settings to a file so that you can use them again. For complete information about **Print** menu options, refer to Chapter 13 of *Using dBASE IV.*

1. Select **BUSPHONE** from the **Reports** panel and then **Print report** from the prompt box.

   The **Print** menu appears.

2. Select the **Destination** option.

   A submenu opens. Notice that the **Write to** option is set to **PRINTER**.

3. Make sure that the **Printer model** option is set to the printer that you installed with dBASE IV. If it is not, highlight the option and press **Spacebar** until you display the correct model name. To view the report as it is being printed, highlight **Echo to screen** and press **Spacebar** to change the setting to **YES**.

4. Press **Esc** to return to the **Print** menu and select the **Control of printer** option. With the **Text pitch** option highlighted, press **Spacebar** until the **ELITE** setting is displayed (the report will print at 12 characters per inch).
5. Highlight the **New page** option and press **Spacebar** to change the setting to **NONE** (no new page is ejected either before or after a printed report page).
6. Press **Esc** to return to the **Print** menu and select the **Output options** option. Press **Esc** to retain the default settings for these options and return to the **Print** menu.
7. Select the **Page dimensions** option. Select **Length of page** and type 60.

   The **Length of page** option is reset to 60.
8. Select the **Offset from left** option, type 10 (to specify a page indent of 10 characters), and press ↵.
9. Press **Esc** to return to the **Print** menu.

Saving Print Settings

Now, save your settings in a *print form* (.prf) file so that you can reuse them whenever you wish.

1. Select **Save settings to print form** from the **Print** menu.

   A prompt box appears asking you to name the print form file. The current print form file, Busphone.prf, is displayed as the default choice. You can press ↵ to choose the default or use **Backspace** to erase the default, type another name, and press ↵.
2. Press ↵ to accept the default.

   dBASE IV saves the settings in the Busphone.prf print form file. Notice that the default for the **Use print form** option is displayed as Busphone.prf.
3. To print the Busphone report, select **Begin printing**.

   Notice that, as the report is printing, the report is also scrolling by on your screen.
4. If you don't want to print the entire report, press **Esc** to cancel printing and redisplay the Control Center. If the report continues to print from your printer's print buffer, turn the printer off.

Labels Print Menu

You display and use the **Print** menu for labels, pictured in Figure 12-4, in exactly the same way as the **Print** menu for reports.

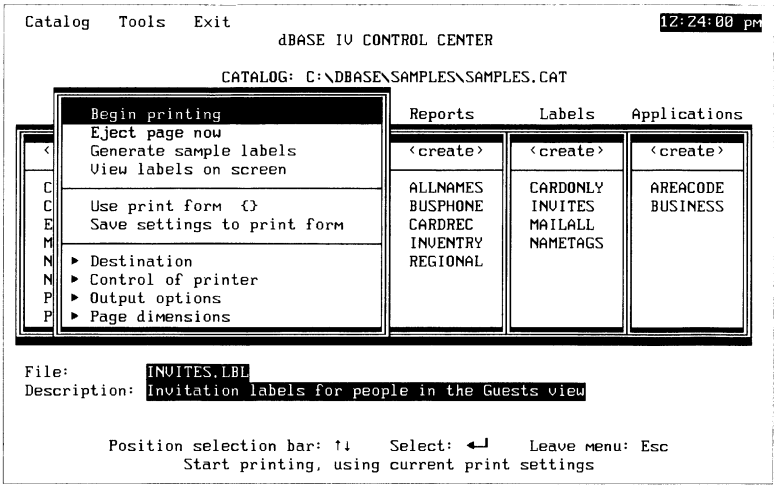


Figure 12-4 **Print** menu for labels

Notice, however, that the **Print** menu for labels contains one extra option, **Generate sample labels**. You use this option to print a test row of labels, so that you can verify that the labels will print as expected and check the printing alignment on the labels paper.

1. Select **INVITES** from the **Labels** panel and then **Print labels** from the prompt box.
2. In the next prompt box that is displayed, select **GUESTS.qbe** (the current view, Busphone, is not the correct one for printing Invites labels).

   dBASE IV makes Guests the current view and displays the **Print** menu for labels.
3. Select **Generate sample labels** from the **Print** menu.

   dBASE IV prints a series of X characters to indicate where the actual labels will print on the page and prompts you to print more samples. If you accept the default (Y), more samples are printed.
4. To begin printing the actual Invites labels, type N. Otherwise, press **Esc** to cancel printing.

Previewing

Before printing, you can see your report or labels on the screen using the **View report on screen** or **View labels on screen** option on either **Print** menu.

When you select this option, the first page of the report or labels is displayed on the screen. Press **Spacebar** to view each page or press **Esc** to stop the display.

Summary

This chapter introduced you to using a print menu to print data in dBASE IV. Printing guidelines, which apply to labels as well as reports, are summarized below.

To print a Quick Report of a file:

Highlight the name of the file in the **Data** panel, press **Shift-F9 Quick Report**, and select **Begin printing** from the **Print** menu.

To modify print settings:

On a **Print** menu, select one of the options preceded by an arrowhead symbol. On the submenu that is displayed, select the desired option. You can press **Spacebar** to change a setting, or type setting information within the curly braces.

Press **Esc** to remove the submenu from the screen and return to the **Print** menu.

To save print settings to a print form:

After modifying print settings, select **Save settings to print form** from the **Print** menu. Accept the default print form filename or select another filename from the box that appears.

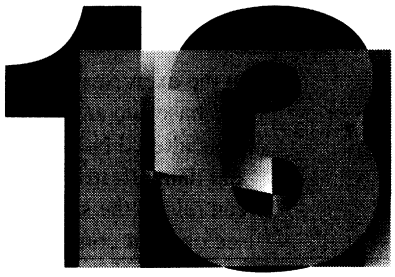
To print using a saved print form:

While the name of the print form file is displayed as the **Use print form** option, select **Begin printing**.

To preview a report or labels on the screen:

Select **View report on screen** or **View labels on screen**.

Dot Prompt Basics



In previous chapters, you performed a number of data management tasks using the Control Center and menu system. In the next five chapters, you will learn to perform some of the same tasks and a number of new ones using another, complementary way of communicating with dBASE IV: typing dBASE IV *commands* at the *dot prompt*.

The dBASE language consists of English-like commands that have a logical syntax for conveying complex instructions to dBASE IV. The commands that you enter at the dot prompt are the same as those used by professional programmers to build sophisticated database applications.

Even if you are not planning to write applications, learning how to use dBASE commands interactively has certain advantages. Entering a command directly at the dot prompt is often quicker than navigating through the menu system to accomplish the same task.

Chapters 13 through 17 let you practice entering dBASE IV commands at the dot prompt. In this chapter, you will learn how to:

- Activate a database file
- Display Help system information
- Redisplay commands that you have entered
- Create memory variables
- Use dBASE functions to display system information
- Enter SET commands to customize your working environment
- Use the internal dBASE IV editor to change command settings in your Config.db file
- Quit dBASE IV

For complete information about any command used in this manual, refer to *Language Reference*.

Preparing for Dot Prompt Chapters

In the exercises in this chapter and the next four chapters, you will be using the Names database file and other sample files that you used in Chapters 7 through 12. If you have altered Names by updating and adding records, some of the screen illustrations and data referred to in this chapter may differ from what you see on your screen. You can reload the sample files from your dBASE IV product disk so that the results you see on your computer screen match those in the exercises:

1. At the DOS prompt, type INSTALL and press ↵.
2. Select **Optional Software Installation**.
3. Select **Sample files** from the Configuration option and follow the prompts to reinstall the sample files.

Preparing for This Chapter

Display the Control Center by typing dbase at the operating system prompt. Notice that the sample files are not displayed in the Control Center panels, and that the catalog displayed is **UNTITLED.CAT**.

Accessing the Dot Prompt

From the Control Center, go to the dot prompt by selecting **Exit to dot prompt** from the **Exit** menu.

The dot (.) prompt appears on the left side of your screen just above the status bar, with the cursor opposite, as shown in Figure 13-1.

Figure 13-1 Dot prompt

Using Commands

dBASE IV commands follow a generalized *syntax*:

VERB [<scope>] [<expression list>] [FOR <condition list>]
[WHILE <condition list>]

Some of the components of the dBASE command syntax are:

- VERB — the command keyword (for example, *USE* or *DISPLAY*).
- Scope — an option that indicates which records the command will affect, based on the current record order (for example, *RECORD 3*, *NEXT 20*, or *REST*).
- Expression list — a list of fields and other command parameters (for example, *Lastname*, *Firstname*, or *TAG Names*).
- Condition list — an expression that limits the records that are affected by the command (for example, *FOR Lastname = "Smith"*, or *WHILE Salary > 20000*). You can specify more than one condition, and can combine conditions with a scope.

The scope, expression list, and condition list enable you to focus a command's effect, and are not required by most commands.

In dBASE IV manuals, the following conventions are used in specifying command syntax:

- Command verbs are shown in capital letters.
- An item for which you supply a value is indicated by angle brackets (< >). Enter the value, but not the brackets themselves.
- An optional item is indicated by square brackets ([ ]). Enter the item, but not the brackets themselves.
- An item followed by an ellipsis (...) can be repeated as many times as needed.
- Items separated by a slash (/) are mutually exclusive.

Entering Commands

To enter a command, type the command at the dot prompt and press ↵. For example, to access the sample files and open the Names database file:

1. At the dot prompt, enter the following command:

```
. SET DIRECTORY TO SAMPLES
C:\DBASE\SAMPLES
```

2. Open the Names database file by typing:

```
. USE Names
```

   Notice that the path information and the name of the file is displayed on the status bar along with the record pointer, positioned at the first record of Names (**Rec 1/46**).

The first command makes the Samples directory (in which you installed the sample files) the *default* or current directory from which you will run dBASE IV. When you enter the USE command for one of the sample files, dBASE IV will be able to find it in this directory. Also, any file that you create will be stored in this directory.

The default directory is normally the directory from which you started dBASE IV — for example, C:\DBASE, the home directory. If you tried to USE a sample file while the default directory was C:\DBASE, dBASE IV would search in vain for the file in the \DBASE directory and in the root directory (C:\).

By entering SET DIRECTORY TO SAMPLES, dBASE IV automatically searches the Samples directory. While you are working with dBASE IV, the files that you create are automatically stored in the Samples directory.

NOTE
If you include the dBASE home directory in the PATH statement of your Autoexec.bat file, you can start dBASE IV from a directory other than the dBASE home directory (for example, C:\ or C:\WP). If you do this, enter SET DIRECTORY TO \DBASE\SAMPLES. Entering SET DIRECTORY is the same as using the **Set default drive:directory** *option of the* **DOS** *submenu of the* **DOS Utilities** *submenu of the* **Tools** *menu of the Control Center.*

TIP
You need type only the first four letters of a command verb or modifier, for example, DISP STRU for DISPLAY STRUCTURE. You can type a command in uppercase or lowercase. In this manual, all command verbs are capitalized and spelled out in full.

If you make an error while typing a command, before pressing ↵ you can edit the command using the edit keys: ←, →, **Ins**, and **Del**. Or, you can press **Esc** to cancel the command and erase what you typed.

If you enter a command that dBASE IV doesn't recognize, or that contains a syntax error, dBASE IV displays an error box.

1. Enter US Names at the dot prompt.

   An *** **Unrecognized command verb** error box appears with the following options:

   - **Cancel** — cancel execution and erase the command line.
   - **Edit** — return the cursor to the dot prompt so that you can edit what you typed.
   - **Help** — display a Help screen.

2. Press → and then ↵ (or type E) to select the **Edit** option.

   The cursor is redisplayed on the command line.

3. Press ← to move the cursor to the space after *US* and press **Ins**.
4. Type E to change US to USE.
5. Press ↵ to enter the command.

NOTE
Entering the USE command is the same as selecting the filename in the **Data** *panel of the Control Center and then selecting* **Use file** *from the prompt box.*

Getting Help

dBASE IV provides a full on-line Help system. For example, to get help on the SET FIELDS command, do the following:

1. Enter help at the dot prompt, or press **F1 Help** while the command line is empty.

   The Help screen shown in Figure 13-2 is displayed, with the **Commands** topic highlighted.

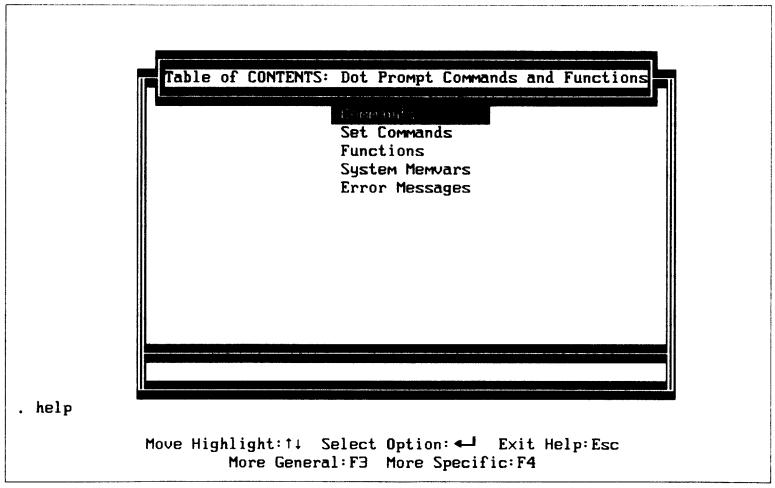


Figure 13-2 Help screen

2. Use the ↓ key to highlight the Set Commands topic, or type SE.
3. Press ↵.

   A list of SET commands appears.
4. Press the **PgDn** key until SET FIELDS is visible. Use the ↑ key to highlight it and press ↵.

   The syntax of the two variants of the SET FIELDS command appears.

5. Press **F4 Next Screen** to see the next page of SET commands information. Press **F3 Previous Screen** to redisplay the information for SET FIELDS.
6. Press ↵ to select the **CONTENTS** button at the bottom of the display.

   The list of SET commands reappears.
7. Press **F3 More General** to redisplay the original table of contents.
8. Press **Esc** to leave the Help system and redisplay the dot prompt.

Here's a quicker way to display the same information for SET FIELDS:

1. At the dot prompt, type HELP SET FIELDS and press ↵.
2. Press **Esc** to move the cursor to the beginning of the command line so that you can type your command while viewing the Help information.

   When you press ↵ to enter your command, the Help box is removed from the screen.

Redisplaying Commands

dBASE IV stores the commands that you enter at the dot prompt in a *history buffer.* By pressing ↑ and ↓ keys, you can display a command from the history buffer so that you can re-enter it without retyping. Before re-entering, you can use the edit keys to modify the command to perform a different task.

For example, to re-enter the USE command that you entered earlier:

1. Press ↑ until *USE Names* appears on the command line.
2. Press ↵ to re-enter it.

   By default, the history buffer stores the last 20 commands that you've entered. Table 13-1 lists the commands that you use to control the history buffer.

Table 13-1 dBASE history buffer commands

Command	Function
DISPLAY HISTORY	Displays the commands stored in the history buffer and pauses at each full screen of commands (if SET HISTORY is set to greater than 20).
LIST HISTORY	Lists the commands stored in the history buffer without pausing.
SET HISTORY ON/OFF	Stops or starts the storing of commands in the history buffer.
SET HISTORY TO	Specifies the number of commands that you want the history buffer to store.

Command Building Blocks

You build a dBASE IV command using the following elements:

- Expressions
- Fields
- Memory variables
- Functions
- Operators

Expressions

You use expressions to create conditions and expression lists for your commands. An expression can be a field, variable, or function, alone or combined with another expression by means of an operator.

Examples of expressions are *Lastname*, *Firstname*, *Phone*, *(State = "GA") .AND. (Business)*.

Fields

A field is a component of a record in a database file. For example, the name *Phone* identifies the field in a Names record that contains a telephone number.

To use field names in the <expression list> of the LIST command, enter:

```
. LIST Lastname, Firstname, Phone
```

dBASE IV scrolls through the data in the Lastname, Firstname, and Phone fields for all records of the Names database file.

TIP
To halt a scrolling display so that you can read information before it scrolls off the screen, press ←. Press ← again to resume scrolling.

NOTE
Entering the LIST command is similar to using the Browse screen to display database information.

Memory Variables

You use a memory variable to store information in memory temporarily. Memory variables are used primarily for storing data that is used in dBASE programs.

To create a memory variable, name it and assign it a value.

For example, enter the following command (an alternate syntax for the STORE command) to create memory variable x and assign it the character value *in*:

```
. x = "in"
in
```

Notice that dBASE IV displays the memory variable value after it is created.

You can use the STORE <exp> TO <memvar list> command to store an expression to more than one memory variable at a time. For example, enter:

```
. STORE 45.6 TO a, b, c
        45.60
```

To display the value assigned a memory variable, use the ? command. For example, to display the value assigned to memory variable c, enter:

```
. ? c
        45.60
```

Functions

dBASE functions perform many different types of operation. For example:

- LOWER( ) converts uppercase letters to lowercase
- EOF( ) indicates the end of a file
- SQRT( ) returns the square root of a positive number

A function name is suffixed by parentheses, which can be used to contain a parameter. Even when no parameter is required, the parentheses are retained to distinguish the function name from the name of a command, variable, or field.

You can return the value of a function using the ? command. For example, to return the current time maintained by your computer, use the TIME() function. Enter:

```
 ? TIME()
14:47:36
```

To display the length of the Lastname field of the Names file, enter:

```
. ? LEN(Lastname)
        15
```

For information about individual functions, refer to Chapter 4 of *Language Reference*.

Operators

You use operators in command expressions to tell dBASE IV to calculate or compare information. The basic kinds of operators are:

- Mathematical (+, –, *, /, **, ^, =) — for performing calculations and returning numeric values.
- Relational (>, <, =, #, < >, >=, <=, $) — for comparing expressions and returning a logical (true or false) value.
- Logical (.AND., .OR., .NOT.) — for comparing two logical expressions and returning a logical (true or false) value.
- String (+, –) — for concatenating character strings.

Operators are evaluated in a specific order of precedence:

1. Mathematical operators: unary + and –, **, ^, * and /, + and –
2. String operators: left to right
3. Relational operators: left to right
4. Logical operators: .NOT., .AND., .OR.

Use parentheses to group expressions when you want to change the order of evaluation. dBASE IV evaluates the inner set of parentheses first.

Customizing Your Working Environment

You can customize your dBASE IV working environment to suit your needs by entering:

- SET commands at the dot prompt, to alter the environment for the current session.
- Configuration commands in your Config.db file, to set the environment every time you start dBASE IV.

SET Commands

You can enter SET commands at the dot prompt to change almost any environment variable for your current dBASE IV session, including default drive, status display, and function key assignments. You can also change settings using a menu listing.

1. Enter SET at the dot prompt.

   A menu bar appears containing five menus. The **Options** menu, listing all SET command settings, is open.

2. Press ↓ until **History** is highlighted.
3. Press ↵ twice to set **History** off and then on again.
4. Press **Esc** to leave the menu and return to the dot prompt.

For complete information about each of the SET commands, refer to Chapter 3 of *Language Reference*.

NOTE
Entering SET commands is similar to changing system options using the **Settings** *option of the* **Tools** *menu of the Control Center.*

Config.db File

During start-up, dBASE IV executes the configuration commands stored in the Config.db file to establish your working environment. For example, it executes the command **COMMAND = ASSIST** so that the Control Center is initially displayed. You can use the internal dBASE IV editor to add, remove, or change configuration commands.

Because you will be using the dot prompt for the remaining chapters of this manual, change Config.db so that the dot prompt is displayed during start-up instead of the Control Center.

1. At the dot prompt, enter:

```
. SET PATH TO \DBASE
. MODIFY FILE CONFIG.DB
```

   Your Config.db file is displayed.

2. Move the cursor to the line containing **COMMAND = ASSIST**.
3. Select **Remove line** from the **Words** menu.

   The **COMMAND = ASSIST** line is removed.

4. Select **Save changes and exit** from the **Exit** menu to redisplay the dot prompt.

NOTE
To display the Control Center while viewing the dot prompt, press **F2 Assist**. *If desired, you can later use the internal editor to replace the* ***COMMAND = ASSIST*** *line in Config.db so that the Control Center is displayed at start-up.*

For complete information about configuration commands, refer to Chapter 2 of this manual.

Quitting dBASE IV

To exit from dBASE IV and return to the operating system, enter:

```
. QUIT
```

NOTE
Entering the QUIT command is the same as selecting **Quit to DOS** *from the* **Exit** *menu of the Control Center.*

Summary

In this chapter, you learned how to enter dBASE IV commands interactively at the dot prompt. The basic principles for using commands are as follows:

To access the dot prompt:

From the Control Center, select **Exit to dot prompt** from the **Exit** menu.

To enter a command:

At the dot prompt, type the command using its proper syntax.

To get Help:

At the dot prompt, type help, or press **F1 Help** when the command line is empty.

To re-enter a command:

Press ↑ to redisplay the command and press ↵ to re-enter it.

To set the environment for the current session:

Enter SET at the dot prompt and select options from the **Options** menu. Press **Esc** to return to the dot prompt.

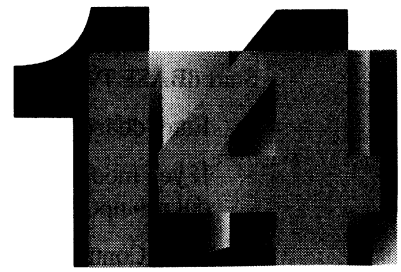
To set the environment when starting dBASE IV:

At the dot prompt, enter SET PATH TO \DBASE and then enter MODIFY FILE CONFIG.DB. Use the cursor keys and **Words** menu options to enter new configuration commands and change existing commands. To save your changes, select **Save changes and exit** from the **Exit** menu.

To exit from dBASE IV:

At the dot prompt, enter QUIT.

Displaying and Modifying Data



The next three chapters show you how to use dBASE IV commands at the dot prompt to perform some of the tasks that you performed earlier using the menu system. This chapter covers entering and editing data in a database file.

You will learn how to:

- Choose the directory in which to create files
- Display the structure of a database file
- Move the record pointer back and forth over a specified number of records, to the first or last record, and to a specified record
- Display the current record in the database file, records that satisfy a specified condition, and all records; restrict the fields that are displayed
- Print displayed records or save them in a text file
- Cause every command that you enter and all dBASE IV responses to be echoed to the printer
- Display the database design screen so that you can create a new database file
- Add records to a database file
- Edit the current record, a specified number of records, and records that satisfy a specified condition; restrict the fields that are edited
- Display the Browse screen for a database file, specify fields that are always displayed regardless of cursor position, and restrict access to a specified field
- Replace values in a specified field, globally or conditionally
- Mark records for deletion, prevent marked records from being displayed, unmark marked records, and delete marked records
- Close all open files, or just database files and their index files; close all open files in all dBASE IV work areas

For complete information about any command used in this manual, refer to *Language Reference*.

Preparing for This Chapter

Start dBASE IV:

1. Enter dbase at the operating system prompt.

   If you modified your Config.db file as discussed in Chapter 13 and the dBASE IV dot prompt is displayed, skip step 2.

2. If the Control Center is displayed, choose **Exit to dot prompt** from the **Exit** menu to access the dot prompt.

3. To make Samples the default directory, enter:

```
. SET DIRECTORY TO SAMPLES
C:\DBASE\SAMPLES
```

Opening a File (USE Command)

The USE command opens a database file so that you can view and manipulate its data by entering commands at the dot prompt.

1. Open Names.dbf by typing the following command:

```
. USE Names
```

2. To review the structure of the Names database file, enter the command:

```
. DISPLAY STRUCTURE
```

   You see the display shown in Figure 14-1. Notice that the record pointer display on the status bar, **Rec 1/46**, indicates that the pointer is positioned to the first record of the Names file.

```
. use names
. display structure
Structure for database: C:\DBASE\SAMPLES\NAMES.DBF
Number of data records:     46
Date of last update    : 01/18/90
Field  Field Name  Type       Width    Dec    Index
    1  LASTNAME    Character     15             N
    2  FIRSTNAME   Character     10             N
    3  ADDRESS     Character     20             N
    4  CITY        Character     14             Y
    5  STATE       Character      2             Y
    6  ZIP         Character      5             Y
    7  PHONE       Character     13             N
    8  BUSINESS    Logical        1             N
** Total **                     81

.
Command | C:\dbase\samples\NAMES | Rec 1/46 | File |
```

Figure 14-1 Names database structure

NOTE
Entering the DISPLAY STRUCTURE command is similar to selecting the filename in the **Data** *panel and then selecting* **Modify structure/order** *from the prompt box.*

Listing and Printing Data

Dot prompt commands let you view and print data in any of the ways offered by dBASE IV — from a simple screen display of the records in the current database file to a complex printed report of data from a number of different files.

Record Pointer

Each time that you add a new record to a database file, dBASE IV assigns it a unique record number. As you work with the file, dBASE IV uses the record pointer to display the record number of the current record.

When you first open a file, dBASE IV sets the record pointer to the first record. You then use commands to move to other records.

1. Enter the following command at the dot prompt:

```
. SKIP 3
NAMES: Record No   4
```

The record pointer skips over the first three records of Names and points to record 4.

The SKIP [<number>] command moves the record pointer forward (or backward, if you enter a negative number) over the specified number of records. If you enter SKIP without a number, the pointer advances to the next record. Notice that this command only moves the record pointer; it doesn't display the record.

NOTE
Entering the SKIP command is the same as using the **Skip** *option of the* **Go To** *menu of the Browse or Edit screen.*

2. Enter:

```
. GO 15
NAMES: Record No   15
```

The record pointer is positioned to record 15.

The GO <record number> command moves the record pointer to the specified record number. (As a shortcut, type <record number> without "GO".)

GO BOTTOM moves the pointer to the last file record, GO TOP to the first record.

Besides checking the record pointer display on the status bar, you can query the RECNO() function to determine the current record.

NOTE
Entering this form of the GO command is the same as using the **Record number** *option of the* **Go To** *menu of the Browse or Edit screen.*

3. Enter:

```
. ? RECNO()
      15
```

DISPLAY Command

You use the DISPLAY command to list field information. If you do not specify which records, DISPLAY uses only the current record.

1. Enter:

```
. DISPLAY
```

dBASE IV displays record 15.

To specify the fields and records to be displayed, include a scope or field list and a search condition. dBASE IV lists the records that satisfy the condition.

2. Enter:

```
. DISPLAY Firstname, Lastname FOR Lastname > "B"
```

dBASE IV displays the first screen of Firstname and Lastname field information for Names records whose Lastname values begin with a letter that comes after "B" in the alphabet. This is shown in Figure 14-2.

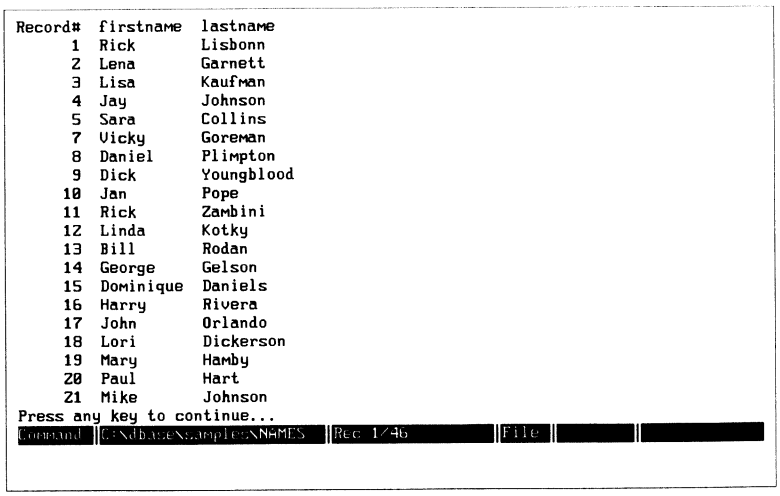


Figure 14-2 DISPLAY result

NOTE
Entering the DISPLAY command is similar to creating a query on the queries design screen.

3. Press any key to see successive result pages.

   You can interrupt a DISPLAY and redisplay the dot prompt by pressing **Esc**.

LIST Command

Use the LIST command as you would use DISPLAY. However,

- A LIST display does not pause for each result screen, but scrolls immediately through the result to the last record.
- If you enter LIST without conditions, dBASE IV lists all records in the database file instead of just the current record.

1. Enter:

```
. LIST Firstname, Lastname FOR Lastname > "B"
```

   You see the same result as that produced by the DISPLAY command above, but there is no pause for each successive page.

NOTE
Entering the LIST command is similar to creating a query on the queries design screen.

2. Enter:

```
. LIST
```

All Names records and fields are displayed, not just a single record.

Pause the scrolling of records by pressing ←. To continue scrolling, press any key.

Saving Lists of Records

You can use LIST to save listed records to a text file.

1. Enter:

```
. LIST Lastname, Firstname, Phone FOR Lastname = "B" TO FILE Namelist
```

dBASE IV creates the file Namelist.txt, containing Lastname, Firstname, and Phone information for the three records whose Lastname fields begin with "B", and displays this information on the screen. You can use the internal dBASE IV editor to open this file and edit its data.

2. Enter:

```
. MODIFY FILE Namelist.txt
```

dBASE IV displays the contents of Namelist.txt.

3. Select **Abandon changes and exit** from the **Exit** menu to redisplay the dot prompt.

Printing Lists of Records

You can use the LIST command to print the records as they are listed on the screen.

If a printer is connected to your system, turn it on and enter:

```
. LIST Lastname, Firstname, Phone TO PRINTER
```

As the list of names and phone numbers scrolls on your screen, your printer begins printing the same list.

NOTE
Entering this form of the LIST command is similar to using **Shift-F9 Quick Report** *from the Control Center.*

Here's an alternative to LISTing TO PRINTER. Enter:

```
. SET PRINTER ON
. LIST Lastname, Firstname, Phone
```

After you enter SET PRINTER ON, every command that you enter and all dBASE IV responses are echoed to the printer.

To stop printer echoing, enter:

```
. SET PRINTER OFF
```

TIP

Print echoing is useful for keeping a log of a dBASE IV session. Use the SET ALTERNATE TO and SET ALTERNATE ON commands to keep a file record instead of a print record.

Function Key Shortcuts

You can use certain function keys at the dot prompt as shortcuts for displaying and maintaining database files. The function keys are listed in Table 14-1.

Table 14-1 Function Keys

Function Key	Operation
F2	Displays the Control Center.
F3	Displays records in the current database, as if you had entered the LIST command.
F4	Lists the database files available in the current directory, as if you had entered the DIR command. (Refer to *Language Reference*.)
F5	Displays the field definitions for the current database, as if you had entered the DISPLAY STRUCTURE command.
F6	Displays information about your dBASE session, as if you had entered the DISPLAY STATUS command. (Refer to *Language Reference*.)
F7	Displays information about how dBASE IV is using your computer's memory, as if you had entered the DISPLAY MEMORY command. (Refer to *Language Reference*.)
F8	Displays the current database record, as if you had entered the DISPLAY command.
F9	Displays an edit screen for adding a record to the current database. (Refer to the Entering Data section later in this chapter.)
F10	Displays an edit screen for the current database record.

Custom Reports

You can do more than simply LIST database records to your printer. Using the report or label design features of dBASE IV, you can design and print your own custom reports and labels containing database information.

Use the CREATE REPORT or CREATE LABEL command to access the design screen for reports or labels. Then, design your report or labels as described in Chapter 12.

Once you've created a report or labels, you can print them using the REPORT FORM or LABEL FORM command, described in Chapter 2 of *Language Reference*.

Creating a Database File

You can create a database file at the dot prompt in much the same way as you created the Mynames file using the menu system in Chapter 7.

1. Enter the following command at the dot prompt:

```
. CREATE Hernames
```

   The database design screen appears. The design process is the same as described in the Creating a Database File section of Chapter 7.

NOTE
Entering the CREATE command is the same as selecting the **<create>** *marker in the* **Data** *panel of the Control Center.*

2. Press **Esc** and select **Yes** from the prompt box to abandon the design screen and redisplay the dot prompt.

Entering Data

Once you have created a database file, use the APPEND command at the dot prompt to add records to the file.

1. Enter the following commands:

```
. USE Names
. APPEND
```

NOTE

- *When you entered the CREATE command in the preceding exercise, you closed the Names file. You must reopen Names before you can append records to it.*
- *Entering the APPEND command is the same as using the* **Add new records** *option of the* **Records** *menu of the Browse or Edit screen.*

The screen shown in Figure 14-3 appears. Notice that the record pointer on the status bar is pointing to a new, blank record at the end of the Names file. At this point, the process of adding new records is the same as described in the Adding a New Record section of Chapter 7.

2. Select **Exit** from the **Exit** menu or press **Esc** to redisplay the dot prompt.

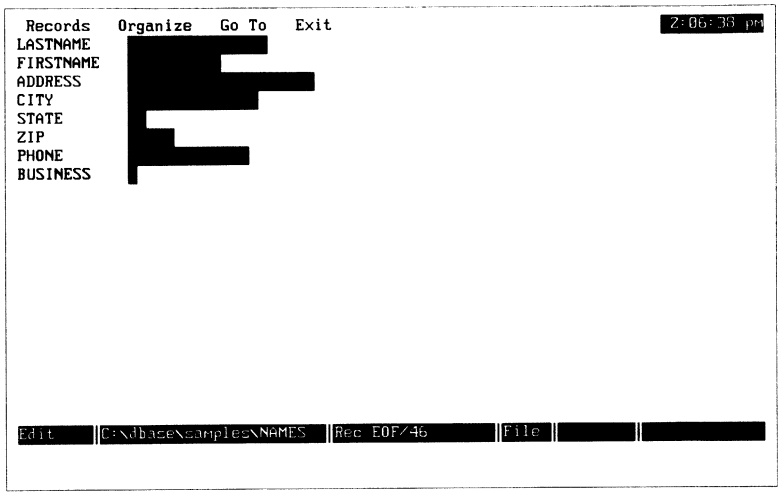


Figure 14-3 Append record screen

Modifying Data

You can display and edit data using the EDIT and BROWSE commands at the dot prompt. You can also update the value of a field in all records using the REPLACE command.

EDIT Command

Entering the EDIT command (or pressing **F10**) lets you see and modify one record at a time on the Edit screen. Enter:

```
. GO TOP
NAMES: Record No     1
. EDIT
```

NOTE
Entering GO TOP is the same as selecting the **Top record** *option of the* **Go To** *menu of the Browse or Edit screen.*

The Edit screen, shown in Figure 14-4, appears containing the first record of Names. Notice that this is the same Edit screen that you used in Chapter 7 to modify database records through the menu system. You can use the same menu options to display records for modification.

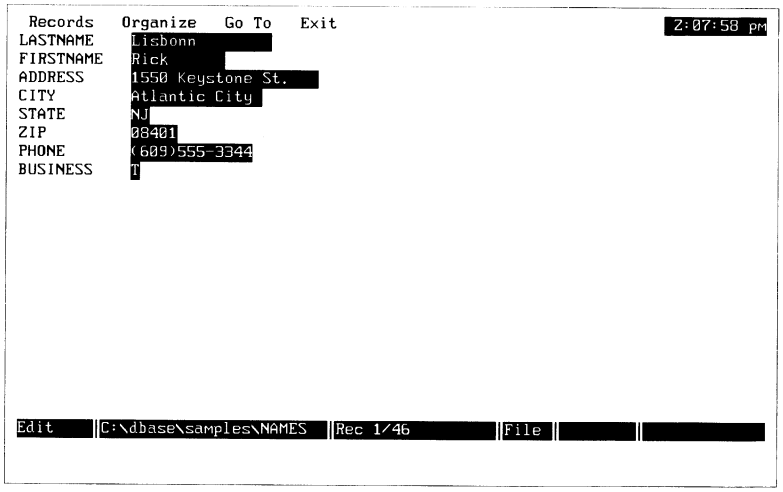


Figure 14-4 Edit screen

Once the Edit screen is displayed for a database file, the process of modifying data in the file is the same as described in the Editing Data in the File section of Chapter 7, summarized as follows:

- Press ↵, ↑, ↓, or **Tab** to move the cursor between fields.
- Use the character keys, **Ins**, and **Del** to make changes to field data.
- Use the **PgUp** and **PgDn** keys to save changes to the current record and move to the previous or next record.
- Press **Esc** to abandon changes to the current record and exit to the dot prompt.
- Press **Ctrl-End** to save changes to the current record and exit to the dot prompt.

However, the EDIT command itself has options that simplify the editing task.

1. Press **Esc** to go to the dot prompt.
2. Enter:

```
. EDIT NEXT 3 FIELDS Lastname, Firstname
```

   An abbreviated Edit screen displays just the Lastname and Firstname fields for the first Names record.

3. Press **PgDn** twice to display the next two records, then press **PgDn** once more to go to the dot prompt.
4. Enter:

```
. EDIT FOR Lastname = "Daniels"
```

   The record for Dominique Daniels is displayed for edit.

5. Press **Esc** to go to the dot prompt.

BROWSE Command

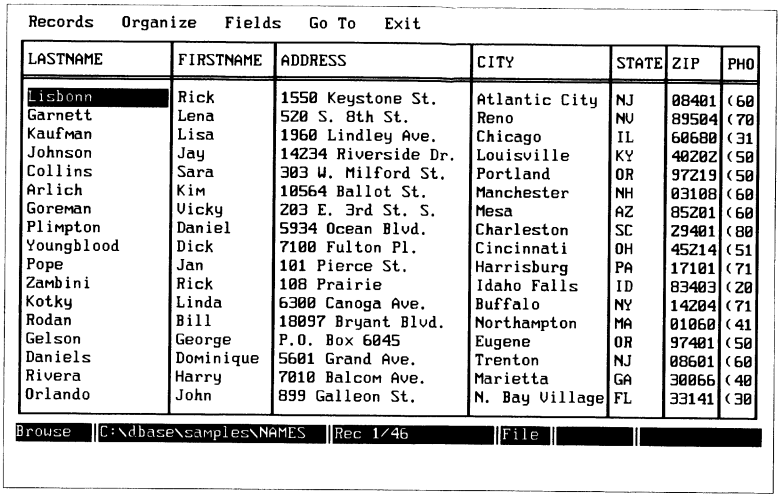
The BROWSE command is used to see and modify a full screen of records at the same time.

1. Enter:

```
. GO TOP
NAMES: Record No     1
. BROWSE
```

   The Browse screen, shown in Figure 14-5, appears containing a number of Names records. This is the same Browse screen that you used in Chapter 7 to modify database records through the menu system. You can use the same menu options to display records for modification.

NOTE
Entering the BROWSE command at the dot prompt is the same as highlighting a filename in the **Data** *panel of the Control Center and pressing* **F2 Data.**



Records Organize Fields Go To Exit

LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	PHO
Lisbonn	Rick	1550 Keystone St.	Atlantic City	NJ	08401	(60
Garnett	Lena	520 S. 8th St.	Reno	NV	89504	(70
Kaufman	Lisa	1960 Lindley Ave.	Chicago	IL	60680	(31
Johnson	Jay	14234 Riverside Dr.	Louisville	KY	40202	(50
Collins	Sara	303 W. Milford St.	Portland	OR	97219	(50
Arlich	Kim	10564 Ballot St.	Manchester	NH	03108	(60
Goreman	Vicky	203 E. 3rd St. S.	Mesa	AZ	85201	(60
Plimpton	Daniel	5934 Ocean Blvd.	Charleston	SC	29401	(80
Youngblood	Dick	7100 Fulton Pl.	Cincinnati	OH	45214	(51
Pope	Jan	101 Pierce St.	Harrisburg	PA	17101	(71
Zambini	Rick	108 Prairie	Idaho Falls	ID	83403	(20
Kotky	Linda	6300 Canoga Ave.	Buffalo	NY	14204	(71
Rodan	Bill	18097 Bryant Blvd.	Northampton	MA	01060	(41
Gelson	George	P.O. Box 6045	Eugene	OR	97401	(50
Daniels	Dominique	5601 Grand Ave.	Trenton	NJ	08601	(60
Rivera	Harry	7010 Balcom Ave.	Marietta	GA	30066	(40
Orlando	John	899 Galleon St.	N. Bay Village	FL	33141	(30

Browse | C:\dbase\samples\NAMES | Rec 1/46 | File

Figure 14-5 Browse screen

2. Press **Tab** or ↵ to move the highlight to the Phone field of the first record.

   Notice that the display moves to the left to reveal fields that normally extend beyond the right edge of the screen, and that the Lastname field disappears to the left to accommodate this information.

Once the Browse screen is displayed for a database file, the process of modifying data in the file is the same as described in the Editing Data in the File section of Chapter 7, summarized as follows:

- Use the character keys, **Ins**, and **Del** to make changes to field data.
- Move to a different record to save changes to the current record.
- Press **Esc** to abandon changes to the current record and exit to the dot prompt.
- Press **Ctrl-End** to save changes to the current record and exit to the dot prompt.

A number of BROWSE options, including LOCK and FREEZE, simplify your editing task.

LOCK Option

The LOCK option of the BROWSE command prevents a specified number of fields on the left side of the screen from moving out of view as you move the cursor to the right in a record.

1. Press **Esc** to go to the dot prompt.
2. Enter:

```
. BROWSE LOCK 2
```

   The Browse screen for Names records appears.

3. Press **End** to move the cursor to the Business field of the first record.

   Notice that this time the first two fields of every record, Lastname and Firstname, remain displayed (locked), while the Address field disappears instead. This enables you to identify the Names record that you are editing even when the highlight is in the last field.

4. Press **Esc** to exit to the dot prompt.

NOTE
Entering this form of the BROWSE command is the same as using the **Lock fields on left** *option of the* **Fields** *menu of the Browse screen.*

FREEZE Option

The FREEZE option of BROWSE limits changes to a single field.

1. Enter:

```
. BROWSE FREEZE Phone
```

   The resulting Browse display lets you alter only the Phone field of each record.

2. Press **Esc** to exit to the dot prompt.

NOTE
Entering this form of the BROWSE command is the same as using the **Freeze field** *option of the* **Fields** *menu of the Browse screen.*

REPLACE Command

The REPLACE command updates the value of a field in all records that satisfy a condition that you specify. Enter:

```
. REPLACE ALL Business WITH .T. FOR State = "CA"
          4 records replaced
```

dBASE IV changes Business field values to .T. for each record whose State field contains "CA" and tells you how many records were affected by the command.

NOTE
Entering this form of the REPLACE command is similar to creating and executing an update query.

Deleting Records

You use the DELETE, SET DELETED, RECALL, and PACK commands to control the deletion of database records that you no longer need.

DELETE Command

The DELETE command marks records for deletion without removing them permanently from the database file.

1. Enter:

```
. DELETE RECORD 12
      1 record deleted
```

   Record 12 is marked for deletion.

2. Enter:

```
. DISPLAY ALL Lastname
```

   The first page of Lastname records appears. Note that record 12 is marked with an asterisk. This indicates that the record is marked for deletion.

3. Press **Esc** to go to the dot prompt.

NOTE
Entering the DELETE command is similar to using the **Mark record for deletion** *option of the* **Records** *menu of the Browse or Edit screen.*

SET DELETED Command

Once you mark a record for deletion, you can use the SET DELETED ON command to exclude the record from being processed by subsequent commands. Normally, records marked for deletion are included in subsequent processing because SET DELETED defaults to OFF.

1. Enter the following commands:

```
. SET DELETED ON
. DISPLAY ALL Lastname
```

   The first page of Lastname records appears. Record 12 is no longer listed. Note, however, that record 12 is still available to commands such as GO, which access records by record number.

2. Press **Esc** and enter:

```
. SET DELETED OFF
. DISPLAY ALL Lastname
```

   Notice that record 12 is again listed, with an asterisk to show that it is still marked for deletion.

3. Press **Esc**.

RECALL Command

Before you actually delete marked records, you can use the RECALL command to unmark records that were marked for deletion. Enter:

```
. RECALL ALL
     1 record recalled
```

Record number 12 is no longer marked for deletion.

You can also unmark selected records using the **Clear deletion mark** option of the **Records** menu on the Browse or Edit screen.

NOTE
Entering this form of the RECALL command is the same as using the **Unmark all records** *option of the* **Organize** *menu of the Browse or Edit screen.*

PACK Command

You use the PACK command to remove permanently any record marked for deletion in the current database file. Once you enter this command, the deleted records cannot be recovered. Enter:

```
. DELETE RECORD 46
        1 record deleted
. PACK
        45 records copied
```

When you enter PACK, dBASE IV deletes marked records and then updates each of the indexes for the database file. After all indexes are updated, dBASE IV redisplays the dot prompt.

NOTE
Entering the PACK command is the same as using the **Erase marked records** *option of the* **Organize** *menu of the Browse or Edit screen.*

Closing Files

To close the Names database file, enter:

```
. USE
```

You can also use the CLOSE command to close a database file. For example, entering:

```
. CLOSE ALL
```

closes all open files. Entering:

```
. CLOSE DATABASES
```

closes all database files and associated index (.ndx and .mdx), memo (.dbt), and format (.fmt) files. Entering:

```
. CLEAR ALL
```

closes all open files in all work areas. (Work areas are explained in Chapter 17.)

Quitting dBASE IV

To exit from dBASE IV and return to the operating system, enter:

```
. QUIT
```

Summary

In this chapter you used dot prompt commands to display, print, modify, and delete data in a database file. The fundamentals of performing these operations are summarized below.

To open a database file:

At the dot prompt, enter the USE <filename> command.

To move the record pointer in a file:

Use the SKIP <number> command to move the pointer ahead (or backward, with a negative <number>) over the specified number of records. Use the GO <record number> command to move the pointer to the specified record. Use the ? RECNO( ) command or the record pointer display on the status bar to determine the current record number.

To display records:

Use the DISPLAY command without parameters to display the record to which the pointer is positioned. Use the DISPLAY <field name,...> FOR <search condition> form of the command to display subsets of file records.

To save or print records:

Use the LIST...TO FILE form of the LIST command to save record data in a text file. Use the LIST...TO PRINTER form of the command to print records on your printer.

To create a database file:

Enter the CREATE <filename> command.

To enter data to a new file:

Open the new database file and then enter APPEND. Use the Edit screen to enter record data or press **F2 Data** to enter data on a Browse screen.

To display and modify one record at a time:

With a database file in USE, enter the EDIT command and use the record pointer commands, **Go To** menu options, **PgUp**, or **PgDn** to move to each record to be modified. Press **Ctrl-End** to conclude an edit session and redisplay the dot prompt.

To display a number of records for modification:

With a database file in USE, enter the BROWSE command and use the **Go To** menu options, **PgUp** or **PgDn**, or the cursor movement keys to display records for modification. Press **Ctrl-End** to conclude an edit session and go to the dot prompt.

To update a number of field values at a time:

Enter the REPLACE command to change all values in a field whose records satisfy a condition.

To mark records for deletion:

Enter the DELETE command to mark records for deletion without permanently removing them from the database file.

To exclude marked records from processing:

Enter SET DELETED ON to prevent records marked for deletion from being processed by commands.

To unmark records marked for deletion:

Enter the RECALL command.

To erase records marked for deletion:

Enter the PACK command.

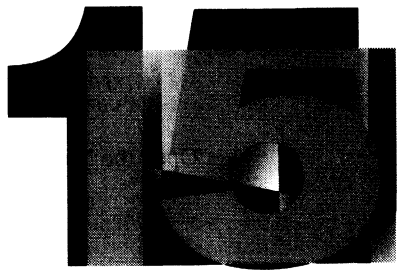
To close the current database file:

Enter the USE command.

To close a number of open files:

Enter CLOSE ALL to close all open files. Enter CLOSE DATABASES to close all open database files and associated index, memo, and format files. Enter CLEAR ALL to close all open files in all work areas.

Organizing Data



This chapter shows you how to use dBASE commands at the dot prompt to sort and index database records for easy retrieval and display. You will learn how to:

- Sort records in a database file by creating a new database file that contains the sorted records
- Create a number of different indexes (tags) within a multiple index file
- Select one of the index tags as the master index for displaying database records
- Open another multiple index file for the same database file and use one of its index tags for displaying database records
- Create a single index file which contains only one index
- Use dBASE IV functions to query index information
- Update single index and multiple index files that were closed when a database file was modified
- Reinstate the natural order of the current database file
- Close all open index files
- Delete single index files and tags from multiple index files

For complete information about any command used in this manual, refer to *Language Reference*.

Preparing for This Chapter

Start dBASE IV:

1. Enter dbase at the operating system prompt.

   If you modified your Config.db file as discussed in Chapter 13 and the dBASE dot prompt is displayed, skip step 2.

2. If the Control Center is displayed, choose **Exit to dot prompt** from the **Exit** menu to access the dot prompt.

3. To make Samples the default directory, enter:

```
. SET DIRECTORY TO SAMPLES
C:\DBASE\SAMPLES
```

4. Open the Names database file by entering:

```
. USE Names
```

Sorting

The SORT command creates a copy of a database file and organizes records in the new file in the order that you specify. Because the SORT command uses temporary files to sort a database file, it requires up to three times the free disk space needed by the original file.

Indexing, discussed later in this chapter, is more efficient than sorting, which creates a whole new version of a database file. Further, as you will see in the next chapter, indexing speeds up your search for records.

Use a sort when you need to:

- Organize a database by the values in several fields using both ascending and descending order
- Create an ordered subset of a database for a particular use

Sorting by a Single Field

Now, sort a database file on a single field.

1. Enter:

```
. SORT TO First ON Firstname
  100% Sorted          45 Records sorted
```

dBASE IV sorts Names records to the new First database file in ascending (alphabetical) order (the default) by data in the Firstname field.

2. To view the new order in the First file, enter:

```
. USE First
. LIST Firstname, Lastname
```

dBASE IV lists the Firstname and Lastname fields for all records in First.

3. Reopen the Names file by entering:

```
. USE Names
```

Sorting by Multiple Fields

You also can sort database records using several fields.

1. Enter:

```
. SORT TO CLF_Name ON City/D, Lastname, Firstname
```

   dBASE IV sorts Names records to the CLF_Name database file. Notice that you can specify a descending sort, rather than the default ascending sort, by appending /D to a field name or by adding the modifier DESCENDING (for example, *City DESCENDING*).

2. To view the sorted order in the CLF_Name file, enter:

```
. USE CLF_Name
. DISPLAY ALL City, Lastname, Firstname
```

   The file is reorganized in the following order:

   a. In descending order by City name

   b. Within City, in ascending order by Lastname

   c. Within Lastname, in ascending order by Firstname

3. Reopen the Names file by entering:

```
. USE Names
```

Sorting a Subset of Records

You can use the SORT command to create a sorted file containing only a subset of records from the original database file.

1. Enter:

```
. SORT TO CA_Names ON Lastname FOR State = "CA"
. USE CA_Names
. LIST State, Lastname
```

   The new CA_Names file contains only Names records whose State field contains CA. Records are sorted by Lastname field values.

2. Reopen the Names file by entering:

```
. USE Names
```

NOTE
Entering the SORT command is similar to using the **Sort database on field list** *option of the* **Organize** *menu of the Browse, Edit, or database design screen.*

Indexing

Like sorting, indexing enables you to reorganize a database file in a particular order. However, instead of creating a new, sorted file, indexing creates an internal file of index key values and record numbers.

For a discussion of how indexing works, refer to the Indexing section in Chapter 11.

Types of Index Files

dBASE IV supports two types of index files, each directly related to the database file for which it is created.

- Single index (.ndx) file — this file contains a single index, and is used in dBASE III PLUS and earlier versions of the dBASE product.
- Multiple index (.mdx) file — this enhanced index file can contain up to 47 different indexes.

Each index in a multiple index file is identified by a *tag*. For example, the Names.mdx file contains an index on the City field of the Names.dbf file with the tag *City*. Another index in Names.mdx, on the expression TRIM(Lastname) + Firstname, has the tag *Trimname*.

If any multiple index tag is defined for a database, a default (or *production*) multiple index file is opened every time you USE the database file. The multiple index file has the same name as the database file, but an .mdx extension.

As you modify the database file, the index tags in the default multiple index file are modified as needed. Therefore, this index file always reflects the current state of your data.

In addition to the default multiple index file, another multiple index or single index file can be open simultaneously. However, a non-production .mdx file or an .ndx file is not opened automatically along with the database file, but must be specifically opened.

Regardless of the number of available indexes, only one single index file or tag at a time can serve as the *master index*, used to determine the order of the database file.

Creating Multiple Index Tags

You use the INDEX command to create an index tag for the database file currently in use. Now, create a new tag for the Names database file.

1. Enter:

```
. INDEX ON City + Zip TAG CitZip OF Names2
100% indexed        45 Records indexed
```

   dBASE IV creates an index tag named CitZip on the City and Zip fields of Names, and stores it in a newly created multiple index file, Names2.

NOTE
City and Zip are both character fields. Normally, you can't create an index key that combines fields of different data types. However, you can use dBASE functions such as STR( ), DTOC( ), and CTOD( ) to combine numeric or date fields with character fields in an index key. Refer to the INDEX command in Chapter 2 of Language Reference.

If you don't specify Names2 as the multiple index file, the new tag is added to the default multiple index file, Names.mdx. Because Names2 did not exist before you entered this command, dBASE IV automatically creates it before adding the new tag.

2. Enter:

```
. INDEX ON State TAG UState OF Names2 UNIQUE DESCENDING
100% indexed        29 Records indexed
```

dBASE IV creates a unique index, which eliminates records containing duplicate State values. The new index reorganizes Names in descending order by State instead of in the default ascending order.

3. Enter:

```
. INDEX ON State + Zip TAG StateZip OF Names2 FOR State >= "LA"
100% indexed      25 Records indexed
```

dBASE IV reorganizes Names records using the new StateZip index: alphabetically by State values, and for each State value, numerically by the Zip values for that State. The FOR clause eliminates records whose State values occur before LA (Louisiana) in the alphabet.

4. To display Names records ordered by the StateZip index, enter:

```
. DISPLAY ALL Lastname, State, Zip
```

NOTE
Entering the INDEX command to create an index tag is the same as using the **Create new index** *option of the* **Organize** *menu of the Browse, Edit, or database design screen. However, you cannot use this menu option to create a new multiple index or single index file. To do either, you must enter the INDEX command at the dot prompt.*

Establishing a Master Index

When you open a database file, no tag in the file's default multiple index file is activated as the master index file. Database records are displayed in natural record order, the order in which they were originally entered.

To display a file in indexed order, enter the SET ORDER command to establish a master index. Enter:

```
. SET ORDER TO TAG Name
Master index: NAME
. DISPLAY ALL Lastname, Firstname
```

dBASE IV uses the Name tag of the default multiple index to display Names records in order by Lastname + Firstname. Records remain in this order until you:

- Enter SET ORDER TO TAG <filename> to name a different master index.
- Enter SET ORDER TO with no parameters to reinstate natural record order.

NOTE
Entering the SET ORDER command is the same as using the **Order records by index** *option of the* **Organize** *menu of the Browse, Edit, or database design screen. The menu option lets you use only index tags in the production multiple index file, however.*

Opening Other Multiple Index Files

You use the SET INDEX command to open other multiple index files in addition to the default multiple index file.

1. Enter:

```
. SET INDEX TO Names2 ORDER TAG CitZip
Master index: CITZIP
. DISPLAY ALL Lastname, City, Zip
```

   dBASE IV opens the Names2 multiple index file and orders Names database records by City and Zip values, according to your CitZip index.

2. Enter:

```
. SET ORDER TO TAG UState
Master index: USTATE
. DISPLAY ALL Lastname, City, State
```

   dBASE IV uses your unique UState index in the Names2 multiple index file to order Names records in descending order by State values and eliminate subsequent records with duplicate state values.

TIP
When you enter the USE command to open a database file, you can specify the multiple index files that you will be using during your session, as well as the index tag by which you initially want database records ordered. Refer to Chapter 2 of Language Reference *for details.*

Using Single Index Files

You create and use a single index (.ndx) file in much the same way as you do a multiple index file. Simply think of an .ndx file as an index file containing a single tag.

1. Enter:

```
. INDEX ON Phone TO Phone
  100% indexed          45 Records indexed
. DISPLAY ALL Lastname, Phone
```

   dBASE IV creates an index on the Phone field of the Names database file and stores it to a single index file named Phone.ndx.

2. Now, close Names and then reactivate it with the new Phone single index as the master index. Enter:

```
. USE
. USE Names INDEX Phone
Master index: PHONE
```

   You could accomplish this same task by entering the following commands:

```
. USE Names
. SET INDEX TO Phone
Master index: PHONE
```

NOTE
Entering the SET INDEX command to order records by a single index file is the same as using the **Order records by index** *option of the* **Organize** *menu of the Browse, Edit, or database design screen. The single index file must be in the current catalog.*

Querying Index Information

If you're not sure about which multiple index files are currently active, query the MDX( ) function. Enter:

```
. ? MDX(1)
C:NAMES.MDX
. SET INDEX TO Names2
Database is in natural order
. ? MDX(2)
C:NAMES2.MDX
. ? MDX(3)
.
```

The message following the first query tells you that the Names multiple index file is active. After you activate the Names2 multiple index file, the message following the second query tells you that Names2 also is active. The absence of a message following the third entry tells you that no other multiple index file is active.

To determine which single index files are active, query the NDX( ) function. Enter:

```
. SET INDEX TO Phone
Master index: PHONE
. ? NDX(1)
C:PHONE.NDX
. ? NDX(2)
.
```

The message following the first query tells you that Phone is active. The absence of a message following the second query tells you that no other single index file is active.

To determine the order of the active database file, query the ORDER( ) function. Enter:

```
. ? ORDER( )
PHONE
```

The message following this query tells you that the Names database file is currently ordered by the Phone index.

To determine which indexes are currently available for the active database file, query the TAG( ) function. Enter:

```
. ? TAG(1)
PHONE
. ? TAG(2)
NAME
.
.
.
. ? TAG(8)
MAIL
. ? TAG(9)
.
```

The message after the first query tells you that the first available index for the Names database file is Phone, by which the file is currently ordered. Subsequent query messages identify as available the tags in the Names multiple index file. The absence of a message following the last entry indicates that there are no more available indexes.

You could more easily determine the number of active indexes by querying the TAGCOUNT function. Enter:

```
. ? TAGCOUNT( )
        8
```

NOTE
Querying the TAG( ) function is similar to using the **Order records by index** *option of the* **Organize** *menu of the Browse, Edit, or database design screen to display the box that lists available indexes. However, the menu option lets you display only tags in the production multiple index file and any active single index file.*

Now, check the tags that are available in Names2. Enter:

```
. SET INDEX TO Names2
Database is in natural order
. ? TAG(1)
NAME
. ? TAG(8)
USTATE
. ? TAG(9)
STATEZIP
```

Notice that, now that the Names2 multiple index file is activated, the Phone index is not available. To make both Phone and Names2 available with Phone as the master index, enter:

```
. SET INDEX TO Phone, Names2
Master index: PHONE
```

To make UState in Names2 the master index instead of Phone, enter:

```
. SET INDEX TO Names2, Phone ORDER UState
Master index: UState
```

You can query functions to learn the attributes of an index tag. For example, for information about the UState index tag, enter:

```
. ? TAGNO("UState")
        1
```

The TAGNO( ) function gives you UState's tag number within its multiple index file, Names2. Enter:

```
. ? DESCENDING("Names2",1)
.T.
. ? UNIQUE("Names2",1)
.T.
```

DESCENDING( ) tells you that UState's index order is descending (that is, not ascending); UNIQUE( ) identifies the index as unique. Entering:

```
. ? KEY( )
STATE
```

displays the key of the currently active index, UState. Entering:

```
. ? KEY("Names2",2)
State + Zip
. ? FOR("Names2",2)
State>="LA"
```

gives you the index key and FOR clause expressions for StateZip (index number 2 in the Names2 file).

Refer to Chapter 4 of *Language Reference* for more information about these functions.

TIP
To list active index files, along with other system information, enter DISPLAY STATUS or press **F6 Select** *at the dot prompt.*

Keeping Indexes Current

As you modify records in a database file, all tags in the default multiple index file are updated automatically. This is because the default multiple index file is always opened with the database file. However, other multiple index files and single index files associated with the database file are not updated unless they are open during database modification.

To update index files that were closed during database modification, open each file and enter the following command:

```
. REINDEX
```

dBASE IV displays messages as indexes are updated.

Reinstating Natural Order

Use the SET ORDER command to reinstate natural record ordering in the database file. Enter:

```
. SET ORDER TO
Database is in natural order
. DISPLAY ALL Lastname, Firstname
```

dBASE IV reorganizes Names in natural record order. All open index files remain open.

NOTE
Entering the SET ORDER command to reinstate natural order is the same as selecting the **Order records by index** *option of the* **Organize** *menu of the Browse, Edit, or database design screen and then selecting* **Natural Order**.

Closing Indexes

Use the CLOSE INDEX command or the SET INDEX command without parameters to close all open index files except the default multiple index. Enter:

```
. CLOSE INDEX
```

dBASE IV closes the Names2 and Phone index files. The default Names multiple index file remains open.

Deleting Index Tags

Use the DELETE TAG command to delete a tag from a multiple index file. If you delete all of the tags in a multiple index file, the file itself is also deleted.

Now, delete the UState tag from the Names2 multiple index file.

1. Enter:

```
. SET INDEX TO Names2
Database is in natural order
```

   dBASE IV reopens the Names2 multiple index file. Before you can delete a tag from a multiple index file, the file must be open.

2. Enter:

```
. DELETE TAG UState OF Names2
```

   dBASE IV permanently deletes the UState tag.

   In this command, specifying the Names2 multiple index file is redundant because no UState tag exists in the only other open multiple index file, Names. However, if there were another open multiple index file that also contained a tag named UState, you would need to specify Names2.

NOTE
Entering the DELETE TAG command is similar to using the **Remove unwanted index tag** *option of the* **Organize** *menu of the Browse, Edit, or database design screen. However, the menu option lets you delete only tags in the production multiple index file.*

Deleting Single Index Files

Use the ERASE or DELETE FILE command to remove a single index file. Before you can delete an index file, it must be closed. Enter:

```
. DELETE FILE Phone.ndx
File has been deleted
```

Quitting dBASE IV

To exit from dBASE IV and return to the operating system, enter:

```
. QUIT
```

Summary

In this chapter you used dot prompt commands to sort and index a database file. These tasks are summarized as follows:

To sort a database file:

Open the database file and use the SORT TO <filename> ON <field name,...> command to sort the file on one or more fields to a new database file. Enter the USE command to open the new file and then enter the LIST or DISPLAY ALL command to display its contents.

To sort a portion of a database file:

Enter the SORT TO <filename> ON <field name,...> FOR <condition> command. Enter the USE command to open the new file and then enter the LIST or DISPLAY ALL command to display its contents.

To create multiple index tags:

Open the database file. Enter the INDEX ON <key expression> TAG <tag name> OF <.mdx filename> command to create a new multiple index file and its first tag. If you don't specify the OF clause, the tag is added to the default multiple index file for the database file.

To set a master index for the current file:

Enter the SET ORDER TO TAG <tag name> command to use a tag in the default multiple index file. To use a tag in another multiple index file, enter the SET INDEX TO <.mdx filename> ORDER TAG <tag name> command.

Once another multiple index file is open, enter the SET ORDER TO TAG <tag name> commmand to use one of its tags. If <tag name> is identical to a tag name in the default multiple index file, enter SET ORDER TO TAG <tag name> OF <.mdx filename>.

To create a single index file:

Open the database file and enter the INDEX ON <key expression> TO <.ndx filename> command. To establish a single index as the master index, enter the SET INDEX TO <.ndx filename>.

To determine the names of active .mdx and .ndx files:

For multiple index files, query the MDX( ) function (for example, ? MDX(1), ? MDX(2), and so on); for single index files, query the NDX( ) function.

To determine the order of the current file:

Query the ORDER( ) function; for example, ? ORDER( ).

To get information about active index tags or .ndx files:

Query the TAG( ) function to determine the names of active .mdx tags or .ndx files; for example, ? TAG(1), ? TAG(2), and so on. Query: NUMTAG( ) to determine the number of active indexes; TAGSLOT( ) to determine an .mdx file's tag number within its multiple index file; DESCENDING( ) to verify index order as descending; UNIQUE( ) to determine whether an index is unique; KEY( ) to learn the index key expression; and FOR( ) to learn the FOR clause expression.

To update non-default indexes for a database file:

The default multiple index file for a database file is updated automatically as the file is changed. To update other multiple index files and single index files, open each file and enter the REINDEX command.

To reinstate natural file ordering:

Enter SET ORDER TO without parameters to change the order of the active database file to that in which records were originally entered.

To close indexes:

Enter CLOSE INDEX or SET INDEX without parameters to close all index files except the default multiple index.

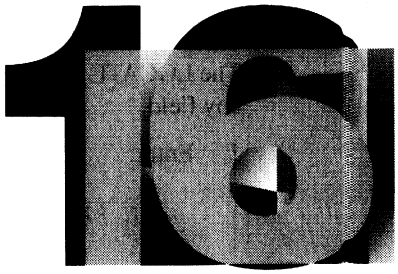
To delete a multiple index tag:

Open the multiple index file. Enter the DELETE TAG <tag name> OF <.mdx filename> command.

To delete a single index file:

Enter the ERASE <filename> or DELETE FILE <filename> command.

Locating Records



This chapter shows you how to use dBASE IV commands at the dot prompt to locate records in your database files. You will learn how to:

- Locate records by searching for a specified field value
- Locate indexed records by searching for a key value that matches a specified expression, either exactly or partially
- Control where the record pointer moves following an unsuccessful search
- Define a search condition to limit the records that are affected by commands
- Limit the fields that are affected by commands
- Create a query, or use an existing query to display data

For complete information about any command used in this manual, refer to *Language Reference.*

Preparing for This Chapter

Start dBASE IV:

1. Enter dbase at the operating system prompt.

   If you modified your Config.db file as discussed in Chapter 13 and the dBASE dot prompt is displayed, skip step 2.

2. If the Control Center is displayed, choose **Exit to dot prompt** from the **Exit** menu to access the dot prompt.

3. To make Samples the default directory, enter:

```
. SET DIRECTORY TO SAMPLES
C:\DBASE\SAMPLES
```

4. Open the Names database file by entering:

```
. USE Names
```

Searching a Database File

The LOCATE command lets you find records in a database file using the data in any field.

1. Enter:

```
. LOCATE FOR Lastname = "Zambini"
Record =      11
```

   The record pointer is now at the first Names record that meets the condition *Lastname = "Zambini"*. Notice that LOCATE does not display the record.

NOTE
Entering the LOCATE command is the same as using the **Forward search** *option of the* **Go To** *menu of the Browse or Edit screen.*

2. To display record 11, enter:

```
. DISPLAY
```

   dBASE IV displays the record for Rick Zambini.

   LOCATE always begins its search at the top of the file, regardless of the position of the record pointer when you enter the command.

3. Enter the following command to continue your search from the current record for records whose Lastname field contains "Zambini":

```
. CONTINUE
End of LOCATE scope
```

   dBASE IV informs you that there are no more records that meet the search condition.

Searching in an Indexed File

Use the SEEK command to search a database file for the first record that contains a specified expression, literal character string, or number. You can use this command only to search an indexed file. LOCATE and CONTINUE can be used to search both indexed and non-indexed files.

SEEK operates faster than LOCATE and CONTINUE because it uses the master index for the file. However, SEEK cannot be used to locate subsequent records that contain the same expression, string, or number, as LOCATE and CONTINUE can.

SEEK Command

The SEEK command searches an indexed file for the first record with an index key that matches a specified expression.

1. Enter:

```
. SET ORDER TO TAG Trimname
Master index: TRIMNAME
```

The master index for Names is set to the Trimname tag in the Names default multiple index file.

NOTE
Trimname defines the expression TRIM(Lastname)+Firstname as the index key. In this expression, the TRIM() function removes any blank spaces at the end of a Lastname character string and the + concatenation operator appends the Firstname character string.

2. Enter:

```
. SEEK "OrlandoJohn"
. DISPLAY
```

dBASE IV displays the record for John Orlando. Like LOCATE, SEEK sets the record pointer to the record location, but it doesn't display the record number.

NOTE

- *The SEEK expression must have the same data type as the index key. If the expression is a character string, as above, enclose it in quotes.*
- *Entering the SEEK command is the same as using the* **Index key search** *option of the* **Go To** *menu of the Browse or Edit screen. (To use this menu option, you must first have used the* **Order records by index** *option of the* **Organize** *menu.)*

Exact or Partial Matching

The SET EXACT command lets you specify whether character searches will look for exact matches.

1. Enter:

```
. SET EXACT ON
. SEEK "S"
Find not successful
```

Because no entry in the Trimname index contains the letter *S* by itself, dBASE IV does not find a match.

2. Enter:

```
. SET EXACT OFF
. SEEK "S"
. ? Lastname, Firstname
Sanders         Kathy
```

Notice that dBASE IV finds the first record with a Trimname index key that begins with *S*. SET EXACT OFF is the default.

Controlling the Record Pointer

Use the SET NEAR command to control where the record pointer moves following an unsuccessful SEEK.

1. Enter:

```
. SET EXACT ON
. SEEK "S"
Find not successful
```

Notice that the record pointer is positioned at the last file record.

2. Enter:

```
. SET NEAR ON
. SEEK "S"
Find not successful
. DISPLAY
```

Notice that the record pointer is set to the record immediately following the record position where it would expect to find a record containing the index key *S*. SET NEAR OFF is the default.

Defining Query Conditions

When you want to work with a subset of records and fields, dBASE IV lets you define scope and condition modifiers that restrict all commands to the subset of records that you've chosen. This frees you from having to specify these conditions each time that you enter a command.

Specifying a Subset of Records

You can use the SET KEY, SET FILTER, or INDEX...FOR command to limit all operations to a subset of records.

If you use SET FILTER or INDEX...FOR, dBASE IV examines every record to see if it matches the filter condition. If you use either command with SET KEY, dBASE IV examines only the records in the range specified by SET KEY. When you are searching a large database, this can save time.

1. Enter:

```
. SET EXACT OFF
. SET KEY TO RANGE "L", "M"
. SET FILTER TO Lastname >="L" .AND. Lastname <= "M"
. GO TOP
NAMES: Record No            38
```

   Notice that after entering SET FILTER, you must move the record pointer for the filter to take effect. Any command that you enter will now affect only records whose Lastname values begin with L or M.

2. Enter:

```
. DISPLAY ALL Lastname
```

   dBASE IV displays Lastname values for filtered records, from L through M.

3. Enter:

```
. SET KEY TO
. SET FILTER TO
```

   The filter is turned off, and no longer applies to commands.

Specifying a Subset of Fields

Use the SET FIELDS command to limit the action of subsequent commands to a subset of fields.

1. Enter:

```
. SET FIELDS TO Lastname, Phone
. DISPLAY ALL
```

   Commands now affect only the Lastname and Phone fields of the Names database file.

2. Enter:

```
. SET FIELDS OFF
. DISPLAY ALL
```

   Commands now affect all Names fields.

3. Enter:

```
. SET FIELDS ON
. DISPLAY ALL
```

   The SET FIELDS specification is reactivated.

4. Enter:

```
. SET FIELDS TO
```

The SET FIELDS specification is cancelled. If you now enter SET FIELDS ON without first specifying new fields using SET FIELDS TO, an error box appears when you enter the LIST or DISPLAY ALL command.

Creating a Query File

If you often use the same subset of records, you can avoid having to enter and re-enter the same SET FILTER and SET FIELDS commands by creating a *query file*.

1. Enter:

```
. CREATE QUERY
Enter filename:
```

2. Enter the following filename: Fonelist

The queries design screen shown in Figure 16-1 appears. Now, create a list of phone numbers for business and social acquaintances who live in California.

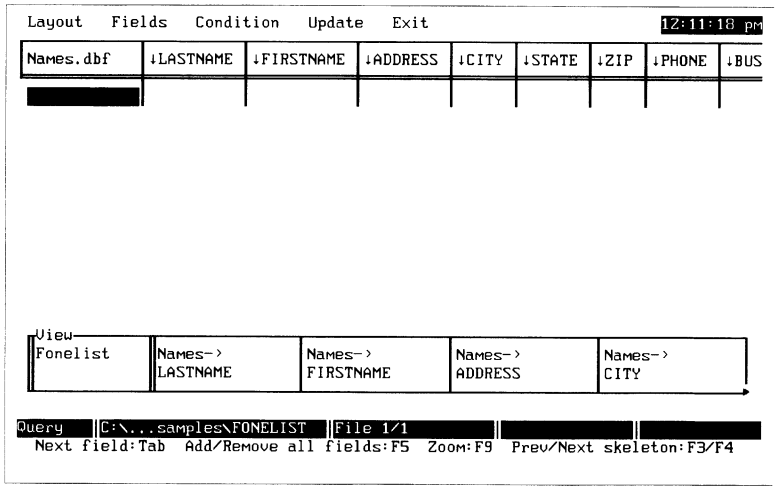


Figure 16-1 Queries design screen

3. Press **F5 Field**.

The view skeleton at the bottom of the screen disappears.

4. Press **Tab** to highlight the **Lastname** field and press **F5 Field**.

The skeleton for the Fonelist view is created at the bottom of the screen, with the Lastname field of the Names file as its first field.

5. Press **Tab** to highlight the **Phone** field and press **F5 Field**.

   The Phone field of Names is added to the view skeleton.

6. Press **Shift-Tab** to highlight the **State** field and type "CA" (include quotation marks).

7. Select **Save changes and exit** from the **Exit** menu.

   The dot prompt appears. Your new query file, Fonelist.qbe, is the file currently in use.

8. Enter:

```
. DISPLAY ALL
```

   The phone list appears.

To learn how to design more complex queries, refer to Chapter 7 of *Using dBASE IV.*

NOTE
Entering the CREATE QUERY command is the same as displaying the queries design screen by selecting the **<create>** *marker in the* **Queries** *panel of the Control Center, or by selecting the* **Transfer to Query Design** *option of the* **Exit** *menu of the Browse or Edit screen.*

Using an Existing Query File

To display data using an existing query, use the SET VIEW command.

1. Enter:

```
. SET VIEW TO Namesqry
. DISPLAY ALL
```

   The search conditions contained in the Namesqry query are activated just as if you had specified them using a SET FILTER command. The Namesqry view query displays Names data for associates who live in Washington, D.C. Now, redisplay data using the Fonelist query.

2. Enter:

```
. SET VIEW TO Fonelist
. DISPLAY ALL
```

NOTE
Entering the SET VIEW command is the same as selecting the name of the query file from the **Queries** *panel of the Control Center.*

Quitting dBASE IV

To exit from dBASE IV and return to the operating system, enter:

```
. QUIT
```

Summary

In this chapter you used dot prompt commands to search for and query data in a database file. Following are the key points of this chapter:

To locate a record using its data:

Open the database file and enter the LOCATE FOR <search expression> command to position the record pointer at the first record that satisfies the condition. Enter the CONTINUE command to locate subsequent records satisfying the same search condition.

To search an indexed file using record data:

Set the master index for the database file. Enter the SEEK <search expression> command. SET EXACT ON to look for an exact match; SET EXACT OFF (the default) to look for an approximation.

To display a located record:

Enter the DISPLAY command without parameters.

To control the record pointer during an indexed search:

Enter SET NEAR ON to keep the record pointer from moving to the end of the file following an unsuccessful SEEK. The pointer is positioned at the record following the one at which dBASE IV expected to find the match.

To restrict operations to a record subset:

Enter the SET KEY and SET FILTER commands and move the record pointer (for example, enter GO TOP). To turn the filter off, enter SET KEY TO and SET FILTER TO without a condition.

To restrict operations to a subset of fields:

Enter the SET FIELDS TO <field name,...> command. To turn the restriction off, enter SET FIELDS OFF; to turn it on again, enter SET FIELDS ON. To cancel the restriction altogether, enter SET FIELDS TO.

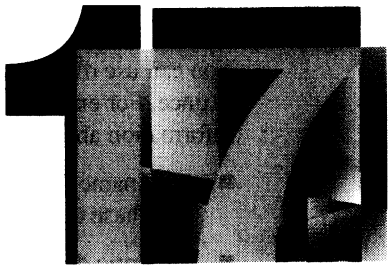
To create a query file:

Enter the CREATE QUERY command. At the prompt, enter a name for the query file. The queries design screen appears for you to create the query. After saving the query and redisplaying the dot prompt, enter the DISPLAY ALL command to display the query result.

To use an existing query:

Enter the SET VIEW TO <query filename> command. Enter the DISPLAY ALL command to display the query result.

Relating Files



This chapter shows you how to relate a number of database files in different work areas. You will learn how to:

- Open database files in different work areas
- Assign alias names to database files in work areas
- Relate one file to another using the values in a shared field
- Display information from the related files
- Save information about the relationship between files so that the relationship can later be restored

For complete information about any command used in this manual, refer to *Language Reference*.

Preparing for This Chapter

Start dBASE IV:

1. Enter dbase at the operating system prompt.

   If you modified your Config.db file as discussed in Chapter 13 and the dBASE dot prompt is displayed, skip step 2.

2. If the Control Center is displayed, choose **Exit to dot prompt** from the **Exit** menu to access the dot prompt.

3. To make Samples the default directory, enter:

```
. SET DIRECTORY TO SAMPLES
C:\DBASE\SAMPLES
```

Relating Database Files

You can use dBASE IV to relate and use the information from several database files at once. For example, say that you maintain the following files to monitor day-to-day information about each stock that you own:

- Stokname.dbf — contains general information about each stock: symbol, name, purchase date, total purchase price, number of shares purchased, and comments.
- Stokpric.dbf — contains day-to-day prices for each stock: symbol, date, closing price per share, high price for the day, low price for the day, and volume traded.

You store each type of information, general and daily, in a separate file for the following reasons:

- To avoid storing redundant information. If you stored both types of information in a single file, you would have to repeat the general information about each stock in each daily price record. This would waste disk space.
- To avoid data entry errors. In re-entering general information each day for each stock, you might make mistakes.
- To avoid extra work. Retyping general stock information each day would be time-consuming.

Figure 17-1 illustrates your stock tracking database.

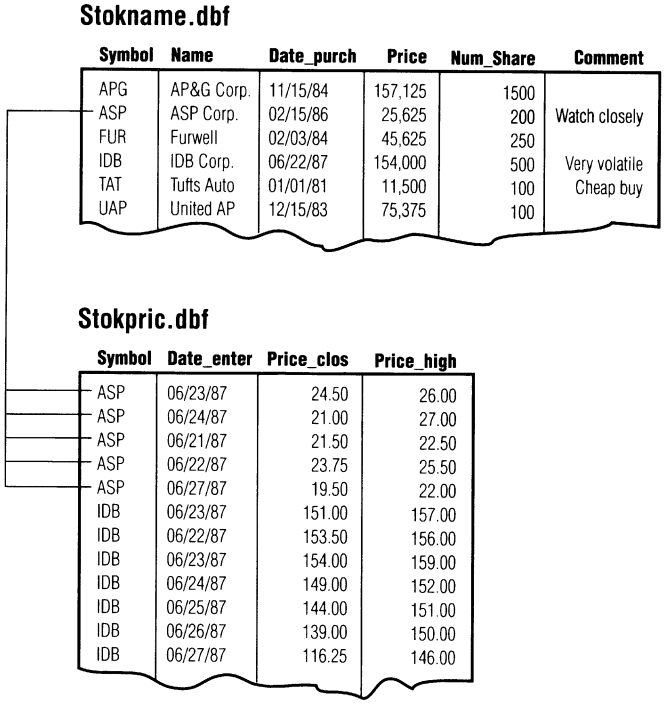


Figure 17-1 Database for tracking stocks

Notice that both files share a common field, Symbol, which is used to relate their information. When you look up a stock in Stokname, dBASE IV uses the Symbol field to find all related records in Stokpric.

Thus, Stokname is considered the *parent* file, containing the antecedent information for each stock. Stokpric is the *child* file, containing information related to each stock.

Each parent file can have active links with several child files. Each child file, in turn, can be parent to other child files, and so on. For example, in an inventory control system, a file that contains an information line for each vendor can be parent to a child file that contains a line for each order, which is parent to a file containing a line for each inventory item.

Using Multiple Files

Besides the ability to relate database files, there are other reasons for having a number of files open simultaneously:

- To compare information in unrelated database files
- To access data in a number of files quickly and efficiently

Work Areas and Aliases

You open each database file in a *work area*. dBASE IV provides 40 work areas, each of which can contain an open database file.

Only one work area at a time can be the current work area, in which you type commands that affect the open database file. However, open files in the other work areas also are readily available.

In a dBASE command, you refer to a work area using its work area number (1 through 40) or an *alias* name. You can assign an alias name using the ALIAS modifier of the USE command, discussed below.

NOTE
If you don't assign an alias name, the USE command assigns a default alias name — normally the name of the database file.

Now, open the Stokname file.

1. Enter:

```
. USE Stokname ORDER Symbol
Master index: SYMBOL
```

   By default, dBASE IV opens the Stokname file in work area 1.

2. Enter:

```
. ?ALIAS( )
STOKNAME
```

The ALIAS( ) function returns the alias of the current work area, in this case, the name of the open file in work area 1.

SELECT Command

dBASE IV opens work areas beginning with work area 1. Each time that you open a database file with the USE command, dBASE IV opens the next available work area for the file.

To choose the next available work area, use the SELECT command.

1. Enter:

```
. SELECT SELECT( )
```

dBASE IV switches to the next available work area, work area number 2. The command that you enter next affects that work area.

2. Enter:

```
. USE Stokpric
```

dBASE IV opens the Stokpric database file in work area 2.

3. Enter:

```
. DISPLAY STRUCTURE
```

The file structure of the Stokpric file (preceded by the commands that you've typed since beginning this session) appears, as shown in Figure 17-2.

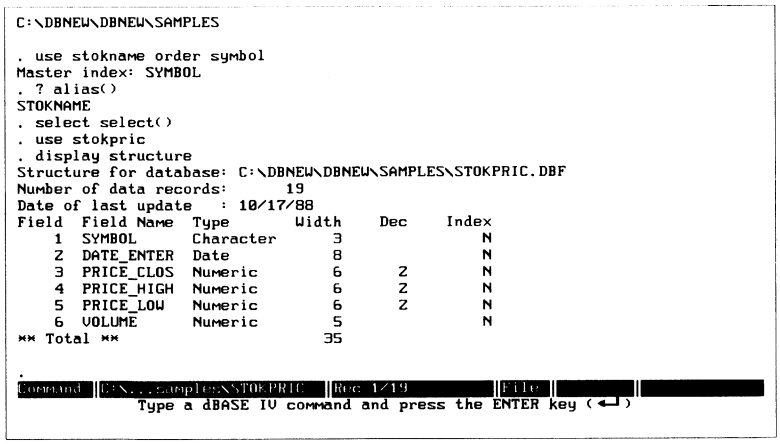


Figure 17-2 Structure of Stokpric.dbf

Now, display the structure of the Stokname file in work area 1.

4. Enter:

```
. SELECT Stokname
. DISPLAY STRUCTURE
```

Both Stokname and Stokpric files are open. You can work with either one by SELECTing its work area.

USE Command

You can select a work area at the same time that you open a database file by entering the USE command. To open Stokpric in work area 2, for example, you could have entered the following command:

```
. USE Stokpric IN SELECT( )
```

You can also employ USE to change the alias of a work area.

Enter:

```
. USE Stokpric IN 2 ALIAS Price
```

The alias for work area 2 is now Price, not Stokpric.

NOTE
When you enter the USE command with the IN option to open a file in a work area that already contains a file, dBASE IV closes the existing file and opens the one that you've specified in its place.

Relating Files

The following requirements apply to files that are to be related:

- The files must share a common field.
- The child file must have a master index on the common field.

Stokname and Stokpric share a common field, Symbol. Activate Symbol as the master index for the child file, Stokpric.

1. Enter:

```
. SELECT Price
```

   dBASE IV selects work area 2 by its new alias name, Price.

2. Enter:

```
. SET ORDER TO TAG Symbol
Master index: SYMBOL
```

SET RELATION Command

Use the SET RELATION command to create a relationship between a parent file in the current work area and a child file in a work area that you specify.

1. Enter:

```
. SELECT Stokname
```

   dBASE IV selects work area 1 by its alias name, Stokname.

2. Enter:

```
. SET RELATION TO Symbol INTO Price
```

   Records in the parent file, Stokname, are related to records in the child file, Stokpric, by the common field, Symbol. Now, when you access a record in Stokname, dBASE IV uses the value in the Symbol field to find the first related record in Stokpric (that is, the first record that contains a matching Symbol value).

Displaying Child Records

Once parent and child files are related, you can refer to fields in related child records using the syntax <alias>-><field name>. Enter the pointer symbol by typing a hyphen (-) and a greater-than sign (>).

1. Enter:

```
. LIST Symbol, Name, Price->Date_enter, Price->Price_clos
```

dBASE IV displays seven lines of data, one for each stock record in Stokname. Each line contains Symbol and Name data for the record, plus Date_enter and Price_close data from the first child record in Stokpric (alias Price, in work area 2) that is related to the Stokname record.

However, a given parent record in Stokname can be related to several child records in Stokpric. To locate all available child records for each parent record, use the SET SKIP command.

2. Enter:

```
. SET SKIP TO Price
```

dBASE IV locates all Stokpric child records that are related to each parent Stokname record.

3. Again, enter:

```
. LIST Symbol, Name, Price->Date_enter, Price->Price_clos
```

The resulting display is shown in Figure 17-3. Notice that dBASE IV displays the same field information as before, but for all child records related to each parent record.

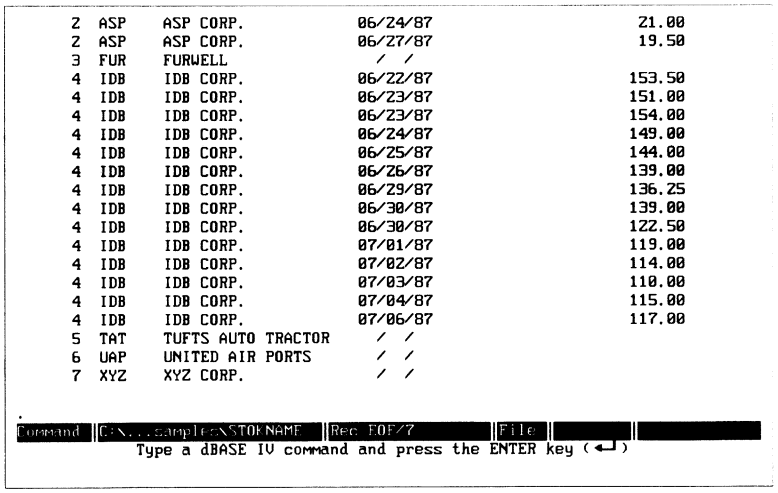


Figure 17-3 Combined parent and child records

4. To turn the skip feature off, enter:

```
. SET SKIP TO
```

Saving Relation Conditions

To preserve the relationships that you create between parent and child files, use the CREATE VIEW FROM ENVIRONMENT command.

1. Enter:

```
. CREATE VIEW Stoktrac FROM ENVIRONMENT
```

   dBASE IV creates a view (.vue) file that records all currently open files, relations, filters, and field lists.

2. Enter:

```
. CLOSE ALL
. LIST Symbol, Name, Price->Date_enter, Price->Price_clos
```

   The CLOSE ALL command closes Stokname and Stokpric files and reselects work area 1. Notice that when you enter the second command, dBASE IV displays a prompt box indicating that no database is in use.

3. Press **Esc** to return to the dot prompt.

4. To reactivate the relationship between Stokname and Stokpric, use the SET VIEW command. Enter:

```
. SET VIEW TO Stoktrac
Master index: SYMBOL
Master index: SYMBOL
```

   dBASE IV opens the Stoktrac view file, along with Stokname and Stokpric, and re-establishes the parent-child relationship that existed before you entered CLOSE ALL in step 2.

5. Enter:

```
. LIST Symbol, Name, Price->Date_enter, Price->Price_clos
```

   dBASE IV displays the same parent-child data as in step 1 under Displaying Child Records earlier in this chapter.

NOTE
You also can store parent-child relationships in a query (.qbe) file. Enter the CREATE VIEW command to display the queries design screen and use the menus to open files, specify file relationships, and create filters and field lists. For information about using the queries design screen to relate files, refer to Chapter 6 of Using dBASE IV.

Quitting dBASE IV

To exit from dBASE IV and return to the operating system, enter:

```
. QUIT
```

Summary

In this chapter, you used dot prompt commands to open a number of database files in dBASE IV work areas and to relate and display information from these files. The principles of using work areas and relating files are summarized below.

To refer to a currently open work area:

Use numbers 1 through 40; the name of the file currently open in the work area; or an alias assigned using the ALIAS modifier of the USE command.

To return the alias name of the current work area:

Enter the ? ALIAS( ) command.

To choose the next available work area and open a database file:

Enter the SELECT SELECT( ) command, then enter the USE <filename> command. Or, enter the USE <filename> IN SELECT( ) command. To change the default alias for the work area (the name of the open database file), include the ALIAS <work area alias name> clause (for example, USE Stokpric IN 2 ALIAS Price).

To relate parent and child files:

Select the work area containing the child file and establish as its master index a field that is shared by both parent and child files. Select the work area containing the parent file and enter the SET RELATION TO <master index name> INTO <child work area alias> command.

To display both parent and child records:

Enter the LIST <parent field name,...,child work area alias->field name,...> command to display parent data and data for the first child record that is related to each parent record. To display data for all child records that are related to each parent record, enter the SET SKIP TO <child work area alias> command before entering the LIST command. Enter SET SKIP TO to turn off the skip feature.

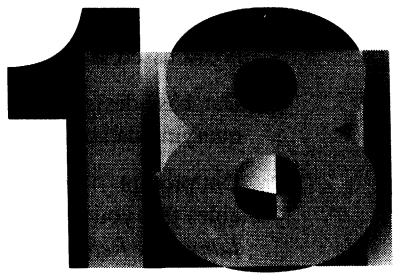
To save the parent-child relationship:

Enter the CREATE VIEW <view name> FROM ENVIRONMENT command.

To reactivate the parent-child relationship:

Enter the SET VIEW TO <view name> command.

Where to Go from Here



Working through the exercises in Exploring dBASE IV should have given you a broad working knowledge of dBASE IV. This chapter describes the reference manuals provided with dBASE IV so that you can add to your knowledge.

Other Manuals

dBASE IV provides the following additional manuals:

- *Using dBASE IV* — describes the menu system and gives procedures for designing databases and organizing database files; displaying, adding, modifying, and organizing data; creating views; extracting and filtering data; using update queries; designing forms and reports; using mailmerge; creating labels; printing; and using Control Center tools, the program editor, and the dCONVERT utility. Also explains the basics of application design and how to create applications without writing programs.
- *Language Reference* — provides a complete guide to dBASE commands, functions, system memory variables, and dBASE IV specifications.
- *Programming in dBASE* — provides information for designing and writing applications using the dBASE IV language, describes the dBASE IV template language used for generating design surface programs, explains how you can use SQL commands in dBASE IV, and provides tips for optimizing dBASE's performance.
- *Quick Reference Guide* — provides a quick guide to dBASE IV commands and functions, terms and symbols used in the dBASE language, system memory variables, and cursor movement keys.

These manuals contain detailed information about the following topics:

- Applications — creating and using applications are described in the Using the Applications Generator section of *Using dBASE IV.*
- Catalogs — grouping related files into catalogs is discussed in Chapter 3 of *Using dBASE IV.*

- Control Center — using the Control Center to access the dBASE IV menu system is discussed in *Using dBASE IV.*
- Database design and organization — designing, organizing, displaying, and querying database files are discussed in Chapters 2 through 8 of *Using dBASE IV.*
- Dot prompt — the dBASE IV commands, functions, and system memory variables that you can use at the dot prompt are described in Chapters 2 through 5 of *Language Reference.*
- Forms — forms design is explained in Chapter 9 of *Using dBASE IV.*
- Labels — designing labels is discussed in Chapter 12 of *Using dBASE IV.*
- Printing — printing database records, reports, forms, and labels is discussed in Chapter 13 of *Using dBASE IV.*
- Program editor — using the MODIFY FILE full-screen text editor to edit files and programs is described in Chapter 15 of *Using dBASE IV.*
- Reports — designing custom reports is described in Chapter 10 of *Using dBASE IV.*

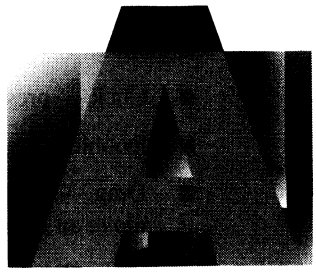
On Your Own

The ways in which you use dBASE IV will depend on your needs and on your style of working. As you become more familiar with the program's capabilities and flexibility, you will begin to see how dBASE IV can make your life easier. If you ever find yourself in unfamiliar territory, the Help screens and the manuals will reorient you.

Further exploration of dBASE IV will be an interesting and rewarding experience.

Appendix

Using a Mouse with dBASE IV



You can use a mouse to perform operations throughout dBASE IV. If you have a mouse driver loaded in memory, dBASE IV displays the mouse pointer as a block on the screen, as shown in Figure A-1. If not, load Mouse.com from your Autoexec.bat file, or Mouse.sys from your Config.sys file.

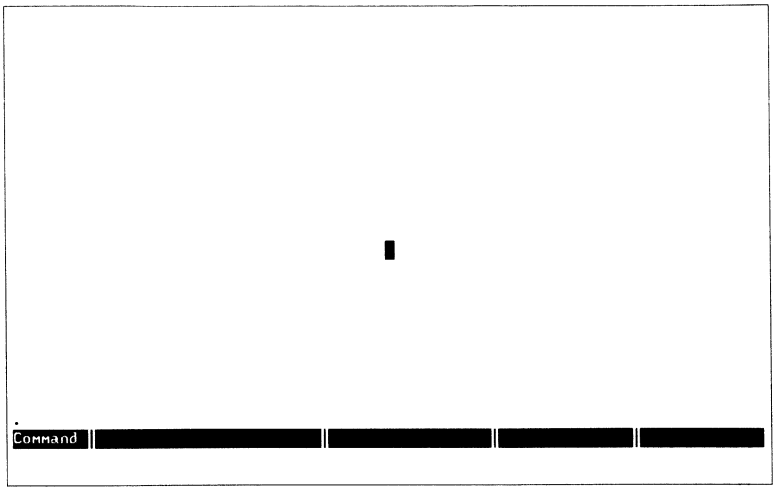


Figure A-1 Mouse pointer at the dot prompt

Manipulate the mouse to point to any screen item and choose or activate it. An item can be a menu option, a filename or anything displayed in a list, or a group of contiguous characters.

NOTE
If you are using a mouse in dBASE IV under Microsoft Windows 3.0 (386 Enhanced), install the mouse driver before starting Windows. Make sure Display usage in DBASEIV.PIF is set to "full-screen."

If your mouse has two buttons, use only the left button to *click*, *double-click*, or *drag* an item:

- Click — press the button and release it.
- Double-click — click twice very quickly.
- Drag — hold down the button and move the mouse to highlight a selection or move an object.

This appendix describes how to use a mouse in the following areas of dBASE IV (listed alphabetically):

- Applications Generator
- Boxes (data entry, error, prompt, and warning)
- **Browse**, **Edit**, and **Form** screens
- Control Center
- Database design screen
- Design surfaces (forms, labels, reports)
- Dot prompt
- **Help** screen
- Lists
- Macros
- Menus
- Navigation line
- Program Editor
- Queries design screen
- Windows
- Word wrap mode (dot prompt editing window, memo fields, report mailmerge layout, report word wrap bands)

Applications Generator

This section summarizes the use of the mouse in the Applications Generator.

Table A-1 Mouse operations in the Applications Generator

Operation	Action
Select an object	Click on the object's border or its interior. This also activates the last menu used with the object.
Position the scroll bar	Click on the scroll bar to activate inside an object
Move an object	Click on the left or top border of the object and drag to the desired position
Size an object	Click on the right or bottom border of the object and drag to the desired size
Use menus	Refer to the Menus section in ths appendix

Boxes

This section summarizes the use of a mouse in data entry, error, prompt, and warning boxes.

Data Entry Box

You use a data entry box to enter a description, a condition, or other information. For example, a data entry box appears when you select the **Edit description of catalog** option from the Control Center **Catalog** menu.

Table A-2 lists each mouse operation that you can perform for a data entry box.

Table A-2 Mouse operations for a data entry box

Operation	Action
Open a box	Click menu option
Move cursor to a position in box	Click the position
Zoom to larger data entry area	Double-click in data entry box
Zoom back to data entry box	Double-click in zoomed area
Abandon data entry	Click outside the data entry box
Accept data entry	Click inside the box but outside the data entry line
Use navigation line	Refer to the Navigation Line section in this appendix

NOTE
If a data entry box can't be zoomed, double-clicking in the box moves the cursor to the position of the mouse pointer, just as clicking does.

Error or Prompt Box

An error box appears when you attempt an operation that dBASE IV doesn't permit. A prompt box appears when dBASE IV needs more information to perform an operation.

Each type of box contains response words — such as **Use file, Display data, Overwrite, Cancel, Edit,** or **Help** — that specify actions you can take to respond to the error or prompt.

Table A-3 lists each mouse operation that you can perform for an error or prompt box.

Table A-3 Mouse operations for an error or prompt box

Operation	Action
Select an action	Click the response word or words
Abandon operation (if a response word or some other means isn't provided)	Click outside the box

Warning Box

A warning box appears when you attempt a contradictory action, such as using the **Add file to catalog** option of the Control Center **Catalog** menu to add a file that is already in the current catalog. A warning box may contain:

- Response words, similar to those used in an error or prompt box
- A message followed by a prompt, such as **Press any key to continue** or **Press Spacebar to continue**

Table A-4 lists the mouse operations that you can perform to respond to a warning box and continue working.

Table A-4 Mouse operations for a warning box

Operation	Action
Select an action	Click the response word or words
Abandon the current operation (if a response word or some other means isn't provided)	Click outside the box

Browse, Edit, and Form Screens

This section summarizes the use of a mouse on **Browse** and **Edit** screens, and on custom data entry and edit forms.

Browse Screen

Table A-5 lists each mouse operation that you can perform on a **Browse** screen. Any equivalent keystroke is indicated in parentheses.

Table A-5 Mouse operations for a **Browse** screen

Operation	Action
Highlight a field (**Tab**, **Shift-Tab**, ↑, or ↓)	Click the field, or the memo icon (**memo** or **MEMO**)
Move cursor to a position in a non-memo field (→ or ←)	Click the position
Move highlight right or left one field (**Tab** or **Shift-Tab**)	Click right or left browse table border
Move highlight to far right or far left field (**End** or **Home**)	Double-click right or left browse table border
Move highlight to preceding or following record (↑ or ↓)	Click top or bottom browse table border
Move highlight up or down one page (**PgUp** or **PgDn**)	Double-click top or bottom browse table border
Size a column	Drag the right column border left or right
Open a memo field (**F9 Zoom**)	Double-click the memo icon (**memo** or **MEMO**). (If the icon is already highlighted, click the icon.)
Edit in memo field	Refer to the Word Wrap Mode section in this appendix
Use lists	Refer to the Lists section in this appendix
Use menus	Refer to the Menus section in this appendix
Use prompt boxes	Refer to the Boxes section in this appendix

NOTE
If you plan to use menus on a **Browse** *screen, make sure STATUS is set to ON.*

Edit Screen

Table A-6 lists each mouse operation that you can perform on an **Edit** screen.

Table A-6 Mouse operation for an **Edit** screen

Operation	Action
Move cursor to a field	Click the field, or the memo icon (**memo** or **MEMO**)
Move cursor to a position in a non-memo field	Click the position
Open a memo field	Double-click the memo icon (**memo** or **MEMO**). (If the cursor is already on the icon, click the icon.)
Edit in a memo field	Refer to the Word Wrap Mode section in this appendix
Use lists	Refer to the Lists section in this appendix
Use menus	Refer to the Menus section in this appendix
Use prompt boxes	Refer to the Boxes section in this appendix

NOTE
If you plan to use menus on an **Edit** *screen, make sure STATUS is set to ON.*

Form Screen

Use the mouse on a custom data entry and edit form much as you do on an **Edit** screen. With the mouse, you can move the cursor directly to a field whose information you want to change without pressing cursor keys.

As with cursor keys, using a mouse to enter data on a form is affected by sequential processing of the GETS used to create the form, by RANGE REQUIRED or VALID REQUIRED checks, and by nested READs. (Refer to the @ and READ commands in *Language Reference* for more information.)

Table A-7 lists each mouse operation that you can perform on a custom form.

Table A-7 Mouse operations on a form

Operation	Action
Move cursor to a field	Click the field, the memo field icon, or the memo field window
Move cursor to a position in a non-memo field	Click the position
Open a memo field window	Double-click inside the window. (If the window is already highlighted, click inside the window.)
Edit in a memo window	Refer to the Word Wrap Mode section in this appendix
Close a memo window and save an edit	Press **Ctrl-End** or select **Save changes and exit** from the **Exit** menu. (If no menu bar is displayed, press **F10 Menus**.)
Close a memo window without saving an edit	Press **Esc** or select **Abandon changes and exit** from the **Exit** menu. (If no menu bar is displayed, press **F10 Menus**.)
Use menus	Refer to the Menus section in this appendix
Use prompt boxes	Refer to the Boxes section in this appendix

NOTE
If dBASE IV returns a RANGE REQUIRED *or* VALID REQUIRED *error message when you move the mouse pointer to a field, the dBASE cursor moves to the first intervening field whose RANGE or VALID requirement was not met.*

Control Center

Table A-8 lists each mouse operation that you can perform in the Control Center.

Table A-8 Mouse operations in the Control Center

Operation	Action
Highlight a filename in a panel	Click the filename
Activate a file	Click the highlighted filename (if Instruct is ON)
Create a new file	Double-click **<create>** marker
Use lists	Refer to the Lists section in this appendix
Use menus	Refer to the Menus section in this appendix
Use navigation line	Refer to the Navigation Line section in this appendix
Use prompt box	Refer to the Boxes section in this appendix

NOTE
If Instruct is OFF when you activate a file, a prompt box is not displayed and the file is put in USE. Clicking on a filename already in USE closes the file.

Database Design Screen

Table A-9 lists each mouse operation that you can perform on the database design screen.

Table A-9 Mouse operation on a database design screen

Operation	Action
Highlight a field definition row	Click the row
Move cursor to a field	Click the field
Move cursor to a position in an editable field	Click the position
Cycle through choices in a **Field Type** or **Index** field	Click the field until the desired choice appears
Create a new field definition row	Click below the last field definition row
Use lists	Refer to the Lists section in this appendix
Use menus	Refer to the Menus section in this appendix
Use navigation line	Refer to the Navigation Line section in this appendix
Use prompt box	Refer to the Boxes section in this appendix

Design Surfaces (Forms, Labels, Reports)

Table A-10 lists each mouse operation that you can perform on the forms, labels, and reports design surfaces.

Table A-10 Mouse operations on design surfaces

Operation	Action
Add a field	Double-click the position where the field is to be added and use the fields list that appears to define the field. Or click **Addfield:F5** on the navigation line.
Modify a field	Double-click the field
Highlight or unhighlight item	Click the item or outside the item
Move or copy text, a field template, box, or window	Highlight the item on the navigation line, choose **F6 Select**, and press ↵. On the navigation line, choose **F7 Move** or **F8 Copy**, move the item to the desired destination using the mouse, and click.
Create box or line	Choose **Box** or **Line** from the **Layout** menu, click the position and drag to draw, then release when you're finished
Size field template, box, or window	Click the template. On the navigation line, select **Size: Shift-F7**, move pointer to desired size, and click
Open/close a report band	Double-click band border. (If the band is already highlighted, click the band border.)
Edit report word wrap band	Refer to the Word Wrap Mode section in this appendix
Use lists	Refer to the Lists section in this appendix
Use menus	Refer to the Menus section in this appendix
Use navigation line	Refer to the Navigation Line section in this appendix
Use windows	Refer to the Windows section in this appendix

Dot Prompt

Table A-11 lists each mouse operation that you can perform at the dot prompt.

Table A-11 Mouse operations at the dot prompt

Operation	Action
Move cursor to a position on the command line	Click the position
Open editing window	Double-click the command line
Edit in editing window	Refer to the Word Wrap Mode section in this appendix

Help Screen

Table A-12 lists each mouse operation that you can perform on a **Help** screen.

Table A-12 Mouse operations on a **Help** screen

Operation	Action
Select a button (**CONTENTS**, **RELATED TOPICS**, and so on)	Click or double-click the button
Highlight item in a list (of contents, command names, and so on)	Click the item
Select item in list (of contents, command names, and so on)	Double-click the item
Move highlight up in a list	On the top box border (below the title): click to move up one item; double-click to move up one page
Move highlight down in a list	On bottom border of box (below the listed items): click to move down one item; double-click to move down one page
Exit from Help	Click outside the box
Use navigation line	Refer to the Navigation Line section in this appendix

NOTE
If there are more items in a list than appear on the screen, the mouse pointer becomes a ↑ *or* ↓ *when on the top or bottom border. Double click the top border for* **PgUp**. *Double click the bottom border for* **PgDn**.

Lists

This section summarizes the use of a mouse on any list. There are several types of lists:

- Files, fields, functions, operators (for example, the list displayed by **Add file to catalog** on the Control Center **Catalog** menu)
- Expression builders (for example, the list displayed when you press **Shift-F1** on the **FOR clause** option of the **Browse**, **Edit**, or database design screen **Organize** menu)
- Control Center panel filenames

Table A-13 lists each mouse operation that you can perform on a list.

Table A-13 Mouse operations on a list

Operation	Action
Highlight list item	Click item
Select list item	Double-click item
Move highlight down	On bottom border of box, click to move down one line; double-click to move down one page.
Move highlight up	On the top border, click to move up one line; double-click to move up one page.
Open or close a list option (identified by a word in angle brackets, such as **<parent>**)	Click the option or open another option
Close list	Click anywhere outside the list box except on a navigation line keystroke identifier
Use navigation line	Refer to the Navigation Line section in this appendix
Use prompt box	Refer to the Boxes section in this appendix

NOTE
If there are more items in a list than appear on the screen, the mouse pointer becomes a ↑ or ↓ when on the top or bottom border. Double click the top border for **PgUp**. *Double click the bottom border for* **PgDn**.

Macros

You can record only keystrokes in a macro, not mouse movements. The mouse is turned off during macro recording. If a macro that you are playing back pauses for a response, you can use the mouse to respond.

Menus

This section summarizes the use of a mouse on any menu, such as the **Tools** or **Catalog** menu on the Control Center menu bar, and their submenus.

There are several types of menu options:

- Action — causes an action to occur (example: the **DOS utilities** option of the **Tools** menu)
- Fill-in — displays a data entry box (example: the **Edit description of catalog** option of the **Catalog** menu)
- List — displays a list of choices (example: the **Add file to catalog** option of the **Catalog** menu)
- Prompt — displays a prompt box (example: the **Remove highlighted file from catalog** option of the **Catalog** menu)
- Submenu — opens a submenu containing further options (example: the **Macros** option of the **Tools** menu)
- Cycle — changes to the next available setting or toggles a setting on or off (example: the **Options** submenu of the **Settings** submenu of the **Tools** menu)

Table A-14 lists each mouse operation you can perform on a menu.

Table A-14 Mouse operations on a menu

Operation	Action
Open a menu on the menu bar	Click the menu name
Activate a menu option	Click the option
Cycle (or toggle) option setting	Click option until the desired setting appears
Move highlight up or down through options	With pointer on the top or bottom menu border, click to move up or down one option; double-click to move to the first or last option
Close menu	Click outside the menu
Close submenu	Click outside the submenu
Use data entry box	Refer to the Boxes section in this appendix
Use lists	Refer to the Lists section in this appendix
Use navigation line	Refer to the Navigation Line section in this appendix
Use prompt box	Refer to the Boxes section in this appendix

NOTE

- *The number of clicks that it takes to open or close a menu or to activate an option depends on whether another menu or submenu option is active. For example, if the* **Edit description of catalog** *option of the* **Catalog** *menu is active, you will need to click twice to open the* **Tools** *menu.*
- *You can use the mouse on user-defined menus.*

Navigation Line

A navigation line can contain two types of keystroke identifiers: single and double. In Figure A-2, **Data:F2** and **Quick Report:Shift-F9**, on the Control Center screen, are single-keystroke identifiers. You highlight each by moving the mouse pointer to it.

However, **Prev/Next field:Shift-Tab/Tab** and **Prev/Next skel:F3/F4**, on the queries design screen, each represent two keystrokes: **Shift-Tab** and **Tab**, and **F3** and **F4**. You must highlight each keystroke individually. See Figure A-3.

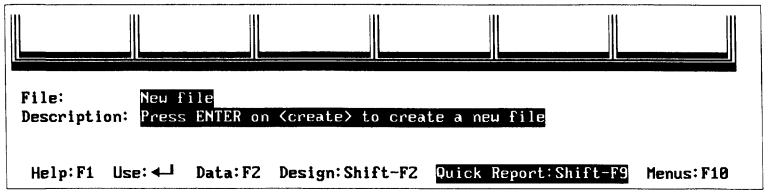


Figure A-2 Single-keystroke Identifier

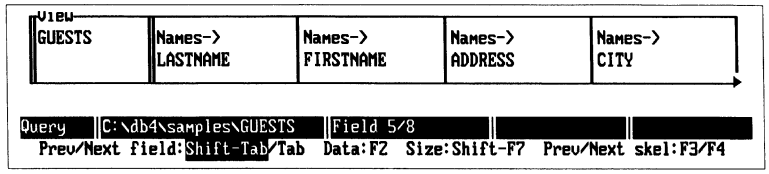


Figure A-3 Double-keystroke Identifiers

Table A-15 lists each mouse operation that you can perform on a navigation line.

Table A-15 Mouse operations on a navigation line

Operation	Action
Select single keystroke	Click keystroke identifier
Select double keystroke	Click the desired keystroke (for example, **F3** or **F4**).

Program Editor

Table A-16 lists each mouse operation that you can perform in the program editor.

Table A-16 Mouse operations in the program editor

Operation	Action
Move cursor to a position on a program line	Click the position
Select a segment of text	Drag to highlight the segment
Unhighlight a selected segment of text	Select a different segment or press **Esc**
Scroll program lines up or down	Drag and pull the pointer toward the bottom or top of the screen
Use menus	Refer to the Menus section in this appendix
Use data entry box	Refer to the Boxes section in this appendix
Use lists	Refer to the Lists section in this appendix
Use navigation line	Refer to the Navigation Line section in this appendix
Use prompt box	Refer to the Boxes section in this appendix

Queries Design Screen

Table A-17 lists each mouse operation that you can perform on the queries design screen.

Table A-17 Mouse operations on the queries design screen

Operation	Action
Highlight a field in a file skeleton	Click the field
Move highlight right in a file skeleton	Click **Tab** on the navigation line. (To highlight a field that is partially visible on the right edge of the screen, click the visible portion.)
Move highlight left in a file skeleton	Click **Shift-Tab** on the navigation line. If a ◄ symbol appears to the right of the filename, click the symbol to move the highlight left one field at a time. Double-click the symbol to move the highlight to the left side of the skeleton.
Move cursor to a position in a field	Click the position
Zoom from field to larger data entry area	Double-click in the field
Zoom back to field	Double-click in the zoomed area
Add a field to or remove a field from a view skeleton	Double-click the fieldname in the file skeleton
Highlight a field in a view skeleton	Click the top border of the field
Move highlight right in a view skeleton	Click the ► symbol at the bottom right of the skeleton to move the highlight one field at a time. Double-click the symbol to move the highlight to the right side of the skeleton.
Move highlight left in a view skeleton	Click the ◄ symbol that appears at the bottom left of the view skeleton to move highlight one field at a time. Double-click the symbol to move the highlight to the left side of the skeleton.
Use data entry box	Refer to the Boxes section in this appendix
Use lists	Refer to the Lists section in this appendix
Use navigation line	Refer to the Navigation Line section in this appendix
Use prompt box	Refer to the Boxes section in this appendix

Windows

In dBASE IV windows created using the DEFINE WINDOWS command, you can use the mouse only to operate in the currently active window, in a menu, and on the navigation line.

Word Wrap Mode

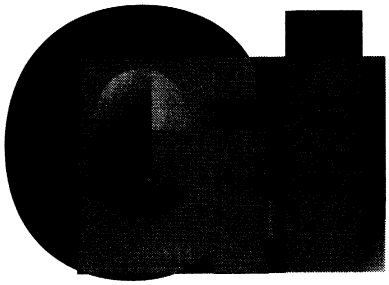
Table A-18 lists each mouse operation that you can perform in word wrap mode in a dot prompt editing window, memo field, report mailmerge layout, or report word wrap band.

Table A-18 Mouse operations in word wrap mode

Operation	Action
Position cursor on a line	Click the position
Select a segment of text	Drag to highlight the segment
Unhighlight a selected segment of text	Select a different segment or press **Esc**
Use menus	Refer to the Menus section in this appendix
Use data entry box	Refer to the Boxes section in this appendix
Use lists	Refer to the Lists section in this appendix

Glossary

Glossary



This section defines terms used in the Exploring dBASE IV section of this manual.

Append The operation for adding new records to a database file.

Application A collection of dBASE command *program* modules that perform one or more database management tasks. Program module files are identified by a .prg extension. When compiled, a program file has the .dbo extension.

Backup A file that is copied to a disk and kept as insurance against damage to the original.

Browse screen A tabular display of the data in a database file. Each record is displayed as a row and each field is displayed as a column.

Catalog A .cat file, which stores information (such as name and location) about each of the files that you use for a particular job. This information enables you to reference these files easily. When you activate a catalog, the Control Center panels display the filenames contained in the catalog.

Character field A field in a database file that can contain printable characters (letters, numbers, special symbols, and blank spaces).

Character string A sequence of ASCII characters — letters, numbers, and symbols — enclosed by single or double quotation marks or brackets ([]).

Clause A portion of a command that performs a specific function. For example, the FOR clause of the INDEX command specifies a condition for including records in an index.

Code page An IBM term for a character set. A code page defines the characters required by a specific country or group of countries.

Command An instruction, entered at the dot prompt on a command line, for performing an operation. A command consists of a verb and clauses that are tailored to suit the requirements of the operation.

Command line The line opposite the dot prompt on which you type a command.

Condition An expression (for example, *total_bill >= 1000*) that returns a true or false value.

Constant An expression (often called a *literal*) that specifies a character string, date, logical value, or number.

Control Center A graphic interface from which you access the dBASE IV menu system. Control Center panels display the database, query, form, report, label, and application files in the active catalog.

Database A system, such as dBASE IV, used for storing, organizing, retrieving, and using data for applications.

Database file A file with the .dbf extension, used for storing and organizing information.

Data Type management system A classification that reflects the type of data (for example, character, date, or logical) that can be entered in a field of a database file.

Default The current value of a command option, or a value (such as ON or OFF) that is automatically supplied by the system if you supply no value.

Default directory The current operating system subdirectory (for example, C:\DBASE). Within dBASE IV, you can change the default directory using the **DOS** menu of the **DOS utilities** screen (which appears when you select the **DOS utilities** option of the Control Center **Tools** menu) or the SET DIRECTORY command at the dot prompt.

Dot prompt The command line on which dBASE IV commands are entered, so named because it is indicated by a dot (.) on the screen. You work with dBASE IV using the dot prompt or the Control Center.

Edit screen A display of the data in a database file in which each record appears on a separate screen within a data entry form.

Expression One or more data values, often containing operators or functions. An expression evaluates to a single character, date, logical, or numeric value. For example, "Los Angeles" is a constant-value expression and *Unitcost*10* is a variable expression.

Field A category of information in a database file record.

Field type The type of data that a field can contain: character, date, logical, memo, or numeric (fixed or floating point).

File extension A three-character suffix that reflects a file's function or type. For example, *.dbf* is the extension for a database file.

File type Identifies the kind of information a dBASE IV file contains. For example, a *database* file contains data; *report*, *label*, *query*, and *index* files contain instructions for manipulating or formatting data; and *application program* files contain instructions for performing tasks.

Filter A set of instructions for selecting specific records, fields, or data items in a database file.

Form A structure for presenting data to a screen, printer, or file, or for capturing data from an input device, such as a computer keyboard. dBASE IV contains screen forms, report forms, and label forms.

Function A dBASE language element that you can use to: return a constant value, for example, PI(); modify a data item, for example, LOWER(); hold a value or condition that can be queried, for example, EOF(); evaluate an input parameter, for example, SQRT(); or perform an action and return a value, for example, SEEK().

Function key One of the keys on your computer keyboard that performs an assigned function. In dBASE IV, the function assigned a function key may change according to the environment. For example, in the Control Center, pressing the **F2 Data** key while highlighting the name of a database file displays the data in the file; pressing the same key at the dot prompt displays the Control Center.

History buffer An area in memory that stores the commands that you enter at the dot prompt. To re-enter a command from the buffer, press the ↑ key to redisplay the command and press ↵. By default, the buffer stores up to 20 commands, but you can use the SET HISTORY TO command to change the default.

Index A file that lets you quickly locate database file records using the values (*index keys*) in one or more fields. You define an index using the INDEX command; you specify the index that you want to use using the SET INDEX command. When you specify an index, database records are automatically displayed in order by index values. Unlike sorting, indexing does not rearrange records in a separate database file.

Index key The expression used for defining an index.

Insert mode The mode in which typing characters moves existing characters at the same cursor position to the right. When insert mode is off, typing characters replaces (types over) existing characters at the same cursor position. Insert mode is toggled on and off by pressing the **Ins** key.

Keyword A word that introduces a command clause. In this manual, keywords appear in syntax in uppercase letters; however, they can be entered in uppercase or lowercase letters.

Language driver Tells dBASE which DOS code page (character set) and language tables to use so that it can create indexes properly, know which characters are valid, and convert characters from uppercase to lowercase (and vice versa).

Language table Contains information about the characteristics of a language, for example, which characters are alphabetic, the alphabetic order of characters, and the mapping of lowercase characters to their uppercase equivalents (and vice versa).

Locking In multi-user dBASE IV, the feature that lets you lock data in files to prevent users from concurrently updating the same data. You specify locking using the SET LOCK command.

Logical field A logical data type field, which can contain only data values that evaluate to true or false (such as .T. or .F., .y. or .n., and so on).

Memo field A field in a database file in which you can enter descriptive text. Memo data is stored in a separate memo (.dbt) file that is referenced every time you enter or edit data in a memo field.

Memory variable A device used to store data in memory temporarily for use by dBASE programs. A *memvar* can contain numeric, character, date, and logical values.

Memvar Memory variable.

Menu bar A horizontal list at the top of the screen that shows the menus that are available on that screen.

Navigation line A line at the bottom of the screen that lists the keys that you can use to move the cursor on the current screen.

Numeric field A field (data type, binary coded decimal or floating point binary) in which you enter numbers that are used in mathematical calculations.

Operator A symbol that indicates how two expressions are to be related (for example, the equals operator (=) that specifies that the expressions must match exactly). Operators include logical, mathematical, relational, and string.

Option A menu selection.

Pack Erases records marked for deletion.

Parameter A value that defines or limits the effect of a command.

Precedence The natural order in which the system applies the operators contained in expressions. For example, logical operators are applied in the following order of precedence: NOT, AND, and OR. Precedence can be overridden by grouping using parentheses.

Program A set of instructions, written in a programming language such as the dBASE language, that is executed to perform an application.

Prompt A message line or box that appears on screen to suggest an action.

Query A set of instructions for viewing or updating a subset of the records and fields in a database file. A view query displays a subset of the records; an update query contains instructions for changing a subset of records.

Record A group of related field information in a database file. For example, a record in an address file would contain information for name, address, and telephone fields for a given person.

Record number A number that identifies the current physical position of a record in a database file.

Record pointer An indicator of the current record in an open database file.

Report A form for displaying or printing information from one or more database files.

Rollback Backing out changes made to a database file during processing of commands contained in a transaction that is interrupted for any reason. During rollback, data is restored to its original state, before it was changed by the transaction. Rollback is accomplished using the ROLLBACK command.

Sort A rearrangement of database file records in a new database file.

Status bar The horizontal line at the bottom of the screen that displays information about the current activity, operation, or command, and the name of the file currently in use.

Structure The names, data types, and arrangement of fields in a database file.

Submenu A subsidiary menu that is displayed by selecting a menu option.

Syntax The rules governing the structure of a command.

Toggle An option or mode that is switched on and off by pressing the same key.

Transaction One or more commands that are processed as a unit of work using BEGIN TRANSACTION and END TRANSACTION commands. For a transaction to be successful, all of its commands must be executed successfully. If any command fails, transaction processing fails. If data is changed during processing of a failed transaction, changes can be backed out using the ROLLBACK command.

UNIQUE A keyword of the INDEX command that finds only the first sequential record (by record number) that contains the same entry for an index key field.

Update query A query that contains instructions for changing a database file.

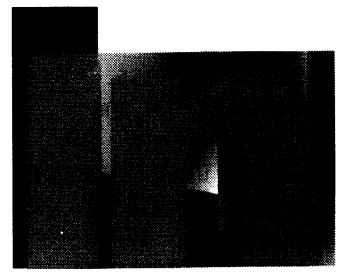
View query A query that is used to display a subset of the fields and records in a database file.

Wildcard Characters used with the LIKE() function to compare character strings. The asterisk (*) character is used to represent any group of characters in a comparison; the question mark (?) character is used to represent a single character.

Work surface The major part of a menu system screen, bordered at the top by the menu bar and at the bottom by the status bar. The work surface of the Control Center screen, for example, consists of the six panels that list the files in the current catalog.

Index

Index



Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P