Information and Learning Resource Services

MIDDLESEX UNIVERSITY

Computing Services

dBASE III Plus Reference PC1

Table of Contents

Pam Dillon

September, 1995

Should you wish to comment on any aspect of this document, or associated software, please direct your comments to:

User Services Group
Computing Services
Bounds Green

1 Introduction

This document is an introduction to the dBASE III PLUS package on the NOVELL Network. It is not a complete guide, if you need extra help refer to the Operations Manual held at Computer Reception. A database broadly consists of a group of records of the same structure. For example a collection of books in a library or a list of names and addresses.

2 Entering and Exiting dBASE III PLUS

From the Novell main menu choose the site menu option, the package will be selectable if it is available on your campus. Do not try to enter from DOS, the Novell system sets parameters to allow correct use of the package.

Select dBASE III PLUS from the menu and press ↵, you will enter a sub-menu allowing you a choice of printers and an option to use the tutorial. You should choose Option 1 to use the Network printer on your campus. Only choose the Local printer if you have a printer attached to your machine.

The tutorial option loads an introductory package (Learning dBASE III PLUS) on the system. This uses the Assistant and is an excellent way of demonstrating the concepts. You will need to borrow "Learning dBASE III PLUS" from Computer Reception.

To leave dBASE III PLUS select **Quit dBASE III PLUS** from the **Set Up** Menu and press ↵.

3 Initial Preparation

dBASE III PLUS allows for the collection and management of data in two ways:

- Via the **Assistant**, a series of menus simplifying routine tasks.
- Using the **Command language**.

An entry in the database relating to one item is called a **record**, and items of information within the record are stored in **fields**. Each field is given a name and description. The field name should be relevant to the information held in it, the field description includes the length and the type of data.

Consider the structure of the data to be entered. **Take time to plan this.**

- What name should be given to each field? Make names meaningful, they can be up to 10 characters long, contain letters or numbers and the underscore character(_).
- How many fields do you need for each record? Think about how you are going to access the data. All the information could in theory be held in one field but this would make it difficult to search for specific information within the database.
- How wide should each field be (maximum)? Some standard field types have set widths (LOGICAL **-** 1, DATE **-** 8, MEMO **-** 10), for others you decide. Allow for the largest entry in your database, or consider abbreviations.
- What type of data is to be held in the field? Is the data to be used in calculations or decisions?

4 The Assistant

The Assistant provides you with a menu based selection for file management. As the name implies it is there to help you build a database using selected commands. The command in use is built up as you select from options presented in sub-menus. It is simpler to use than the control language.

On entering dBASE III PLUS you are presented with a **'.'** prompt. Type **ASSIST** to use the Assistant.

4.1 Using the Menu

Select a Menu with ανδ Oρ type the first letter of the menu name. Choose a menu option with ↑ and ↓, press ↵ to select it, brief instructions are displayed on the bottom of the screen.

You cannot select options from most menus unless you have opened a database file; your first selection should either be to create a new database, or select an existing one.

In some modes you are presented with a sub-menu holding lists of files, field names or options. If the list is larger than the window use ↑ and ↓ to scroll. Where a sub-menu allows you to select more than one item the selected item will be marked with ➤ , continue your selections then press ← or → to leave the sub-menu.

Where you are setting up additional files such as reports use F10 to display the fields available from the database in use. If you are asked to enter a parameter at the keyboard, for instance, when setting conditions to be met, the information box will inform you whether quotes '..' are expected.

Care: while field names are not case sensitive, the data held within the field is, an important consideration when searching for specific records.

The **status bar**, at the bottom of the Assistant menu displays the selected menu, the drive and file in use, and the current position and number of records in the file. Below the bar is brief help information. Directly above it the **active command** is displayed. The active command is built as options from sub-menus are selected.

4.2 Getting Help

Function Key 1 gives additional help on the **highlighted** option in the current menu. For instance for information on the Edit command. Select Update to open the Update Menu, select Edit, then press **F1**. A window, with a brief description and syntax of the command appears. Press any key to leave help mode.

4.3 Simple Steps for the Beginner

Choose the following sequence of menus and selections from them, then refer to the relevant instructions in Section 5.

1. **Create -** select **Database file** to define your record structure
2. **Set up -** select **Database file** to use your new file.
3. **Modify -** select **Database file** to amend the file structure.
4. **Update -** select **Append** to add records to the file.
5. **Update -** select **Edit** to alter records in the file.
6. **Retrieve -** select **List** to produce a simple list of the records.
7. **Set Up -** select **Quit** to finish your session.

5 The Assistant Menus

The menu bar offers a choice of eight menus, namely **Setup, Create, Update, Position, Retrieve, Organise, Modify** and **Tools**.

5.1 Setup

The Setup menu is used to open files and Exit from dBASE III PLUS. You choose the file type to open, the Drive on which the file resides and finally the file name.

5.1.1 Database file

Select an existing database. The hard disk work area is highlighted; choose **A:** (or **B:**) to read from your disk and press ↵. Select the filename from the list of files displayed. You will be asked if the file is indexed. If you answer **Y** you must choose index file(s) from the list provided (the first one chosen is the Master).

5.1.2 Format for screen

Change the default screen format of a record. You need to create the format file first. Refer to the manual for details.

5.1.3 Query

Look at specific records, a Query file containing details of the conditions to be met must exist. Refer to the manual.

5.1.4 Catalogue

Group your files. Used when holding several databases on a drive or disk. Refer to the manual.

5.1.5 View

Select files to view, links several files, and links index files to a specific database. A view file must exist first containing the relevant selection data. Refer to the manual.

5.2 Create

The Create menu allows creation of database and other file types. As in the Setup menu you must select a drive before entering your filename.

5.2.1 Database file

A sub-menu lists the drives available. The hard disk work area is highlighted, choose **A:** (or **B:**) to write direct to your personal disk. If you use the work area you will need to copy the relevant data to your own disk at the end of the session.

> **Enter the name of the file:** type in your new file name and press ↵. Make the name relevant, do not enter a file extension, dBASE III PLUS provides **.dbf**. A window appears allowing specification of the file structure.

Enter your field specifications:

1. Type in the field name and press ↵.

2. Select a field type **-** press the space bar until the correct type is displayed or type the first letter of the field type, and press ↵.

3. Enter the field length (maximum width) and press ↵. Note, decimal places and the decimal point are included in the length.

4. For Numeric field types enter the number of decimal places.

Repeat these steps until all field descriptions are entered.

Save the completed structure with **Control-End**. You will be prompted:

> **Press ENTER to confirm. Any other to resume.**
> Press ↵ to end the file creation. You will then be asked **-**
>
> **Input data records now? (Y/N)**
> type **Y** to enter data, **N** to return to the Assistant menu. Entering data is explained under APPEND mode.

5.2.2 Format

Create a customized screen for data input using APPEND or EDIT mode. This allows you to display text on the screen, and to put more than one field on the same line thus enabling large records to be displayed on a single screen.

5.2.3 View

Creates a view file defining the relationships between database(s), index files etc.

5.2.4 Query

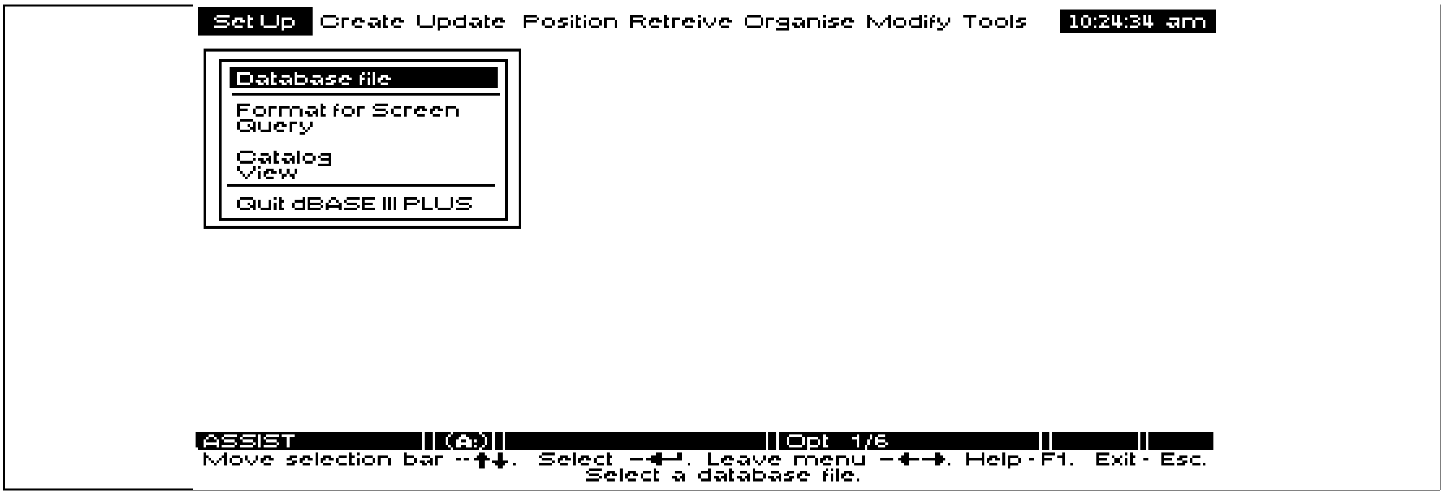
Create a query file to limit the records accessed in the database in use. Only those records meeting the conditions will be accessed.

5.2.5 Report

Defines a file specifying the layout of a report, usually for printing. Select the drive the report file is to be written to (choose the same drive as you chose for your database) and enter a filename for your report. Relate the filename to the database; do not enter an extension, it will be given an extension of **.frm**.

The **Create Report** screen appears, with the **Options** menu open. Highlight the option to use/alter and press ↵. Type in your specific data/information.

→ to the **Groups** menu to produce group reports. The database file must be sorted or indexed on the field within the database relating to a specific group, then by selecting that field you can have page breaks and/or group totals between groups of similar types of records.



```
ASSIST              ||(A:)||                    ||Opt  1/6           ||        ||
Move selection bar -↑↓.  Select -↵.  Leave menu -←→.  Help - F1.  Exit - Esc.
                        Select a database file.
```

Figure 1- The Assistant Menu

→ to the **Columns** menu to set up the main body of your report. Select **Contents** and press ↵ then F10 and select the field to be displayed. Notice the width will alter to the size of the selected field. Select the **Heading** and press ↵, type in the column heading required. Headings can be up to four lines of text, use the semi-colon to insert blank lines or to force text onto the next line.

Press **PgDn** to move to the next column and repeat the process until you have defined all the columns on your report. **Care:** plan the report first, to print on the default printer you cannot define a report width of 300 characters if your printer default is only 80 characters wide.

To insert a column **- PgUp/PgDn** to move to the column after the desired insertion, and press **Control-N.** You will be given a blank columns menu.

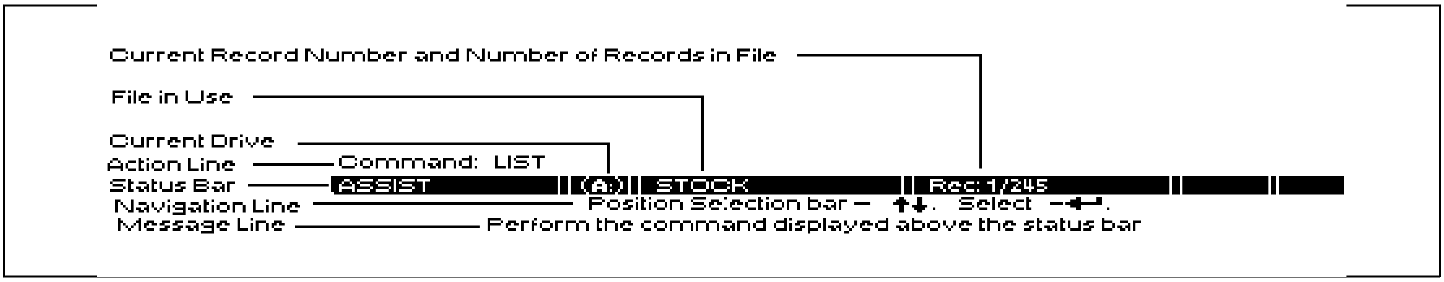
To delete a column **- PgUp/PgDn** to desired column, press **Control-U**.

The **Locate** option displays a list of fields selected in each column. Use it to check that all report fields are entered correctly.

Save the report, → to **Exit Menu** and select **Save**. Choose **Abandon** if you want to cancel the operation.

5.2.6 Label

Create a label form file. You define the label size, the number of columns, and enter the fields to be printed in a similar manner to the Report function.



```
Command:  LIST
ASSIST              ||(A:)|| STOCK              || Rec: 1/245         ||       ||
Position Selection bar -  ↑↓.  Select  -↵.
Perform the command displayed above the status bar
```

Figure2- The Status Bar

5.3 Update

The Update menu allows full screen editing of records within the database in use. You can add, alter, or delete records. Any index files relating to the database in use are automatically updated.

5.3.1 Append

On selection the screen shows a blank record structure. Type in the field details. If you fill the field you are automatically moved to the next and the bell sounds. Press ↵ if the data does not fill the field. **Care:** if you do press when the bell has sounded - the will act as the end of the next field and will move you on too far, you will need to use your cursor keys to return to the correct place to continue.

Move to the next record by pressing **PgDn**.

When all the data is entered press **Control-End** to save the final record and leave append mode. (Pressing ↵ at the beginning of a blank record will also end APPEND mode.

You can also use **INSERT** and **BROWSE** to add records to the file.

5.3.2 Edit

Select to alter a record. Use the **Position** menu to select the correct position in the file first. The current record is displayed on the screen. If the correct record is not selected, move one record at a time using **PgDn** and **PgUp**. Edit using the ↑, ↓, →, ←, to move to the field to be altered and retype the character(s) necessary. Delete information using **Control-Y**. The edit menu at the top of the screen shows necessary instructions and cursor movements. Press **Control-End** keys to save your alterations or use the **Esc** key to abort.

5.3.3 Display

Shows up to 15 records at a time.

5.3.4 Browse

Display edit and append records. Useful if you need to alter several records. It displays several records at one time, and the necessary instructions are displayed in a menu at the top of the screen. If the records are longer than 80 characters you cannot see all the record on the screen at once. You can move across the record a field at a time using **Control** → and thus display the relevant field for alteration.

5.3.5 Replace

Use to replace the contents of a field for all records or for records meeting a specified condition. Refer to the manual for further information.

5.3.6 Delete

Records can be marked for deletion or permanently removed from the database. Choose **Build a search condition** from the sub-menu. Select the desired field and operation and type in the string to be searched for (eg FIELDX = ’mytext’). When you have entered your conditions choose **No more conditions** and select **Execute the command**. All records matching the chosen conditions will be flagged for deletion. Use the **Browse** command to check that the correct records have been marked before permanently removing the records.

5.3.7 Recall

Records flagged for deletion can be **Recall**ed at this point if necessary.

5.3.8 Pack

Remove flagged records permanently.

5.4 Position

The **Position** menu allows you to move to a specific record in the database dependant on specified conditions.

5.4.1 Seek

looks for a specific key within an indexed database. See the manual for further information.

5.4.2 Locate

looks for a specific condition and stops at the the first record meeting it. The default scope is to search the whole database. Choose **Build a search condition,** select the field to be examined from the field list on the screen, choose the condition to be met (equal to, greater than etc) and type in the value, this will be an actual name or value you know should appear in the field selected. Choose **Execute the command** and the file is searched. Use a command such as edit to display and alter the record.

5.4.3 Continue

Used in conjunction with locate to find further records meeting the condition(s) specified in the previous locate command.

5.4.4 Skip

Moves the record pointer forward or back a specific number of records. Select skip and enter a number, a negative value moves backwards.

5.4.5 Goto Record

Moves to the Top or Bottom or to a specific record number.

5.5 Retrieve

The **Retrieve** menu covers report options. The first two commands produce a simple list of fields and if the field sizes to be viewed are wider than the screen or paper then the results will be untidy. When creating final reports you will need to use the **Report** function.

5.5.1 List

Produce a list of all or part of the information in the database. From sub-menus select **Construct a field list** and choose the desired fields, select **Build a search condition** and specify conditions to be met. Finally choose **Execute the command** and answer **Y/N** to the question **Direct the output to the printer?**

5.5.2 Display

Similar to list except it pauses after displaying 20 records prompting **Press any key to continue...** useful when viewing records on the screen.

5.5.3 Report

Uses a report specification held in a previously created report form file. Select the file to be used from the sub-menu then choose **Execute the command** and answer **Y/N** to the question **Direct the output to the printer?**

You can print reports for specific records by selecting the search and scope conditions. See the manual for further assistance.

5.5.4 Label

Uses previously defined label file to print columns of labels.

5.5.5 Sum

Use on numeric data to obtain the sum of all selected numeric fields within the database that apply to the selected conditions.

Select **Construct a field list** and select the field(s) to be summed, then **Execute the command** to sum all the records within the database. If you want to sum only specific records then choose **Build a search/scope condition** before execution and specify the parameters to be met.

5.5.6 Average

Obtain the arithmetic mean of selected numeric fields within the database applying to selected conditions. Similar to **Sum**.

5.5.7 Count

Gives the number of records matching specified conditions.

5.6 Organise

The Organise menu arranges the order of the records in your database.

5.6.1 Index

Allows you to set up an index for the database in use. You are prompted:

**Enter an index key expression - ** press F10 to select a field from the field list. If you need to index on more than one field separate your field selection with +, press ↵ to complete index key selection.

Select the drive on which to write the index file, choose the same drive as the original database, and **enter name of the file**, the file will automatically be given the extension **.ndx**. The data is now selected in the order of the index key.

The records in the database are still in the same physical order and the database may be selected with or without the index in use. When the index is in use new records added to the file are automatically indexed.

5.6.2 Sort

Rearranges the records in the database by creating a new file, the sort can be made in ascending or descending order. To examine records after sorting you must reopen the **Setup** menu and select the sorted file as the file in use. Remember if you update after sorting you will need to re-sort.

5.6.3 Copy

Duplicates the database in use. The whole file or selected parts can be copied by selecting specific fields to be copied and/or defining parameters to be met before a record is copied.

5.7 Modify

The Modify menu allows amendments to be made to file specifications. It is identical to the Create menu except that you are offered a list of existing files instead of being asked to enter a new filename. You can insert or delete existing records or fields. The menu at the top of the screen will display the cursor controls necessary.

5.7.1 Database file

Change the structure of the selected database. All records in the database will be copied to the new file structure when you have finished, data from existing fields will only be copied if the field name still exists.

5.7.2 Format

Modify a customised screen format file.

5.7.3 View

Modifies a view file from the working environment. Stores the current settings and relationships.

5.7.4 Query

Modify a query file

5.7.5 Report

Create/edit a report form file

5.7.6 Label

Create/edit a label form file

5.8 Tools

Allows the use of certain DOS utilities to save exiting the package. In all options you are prompted.

5.8.1 Set Drive

Alter the default drive used to save work on. This is currently set to the hard disk work area. You may prefer to set it to your floppy disk in Drive **A:**.

5.8.2 Copy file

Copy files to/from the floppy disk. You select the drive and name of the file to be copied and then select the drive and type in the name to be given to the copy.

5.8.3 Directory

Select drive to be examined and file types to be displayed.

5.8.4 Rename

Change the name of a file.

5.8.5 Erase

Delete a file, you cannot delete a file that is in use.

5.8.6 List Structure

Allows you to list the structure of the database in use to the screen or printer.

5.8.7 Import

Creates dBASE III PLUS files from a file exported for use in other packages. Refer to the manual.

5.8.8 Export

Creates a single file from the dBASE III PLUS files in use for use by other packages. Refer to the Manual.

6 The Command Language

The command language allows entry of an individual command from the keyboard. The command is entered at the dot prompt. Commands can also be entered in a program file using a text editor, the program file would then be called up from the dot prompt. Syntax for commands is explained in detail in Section 8.

```
General   Syntax:

VERB [<scope>]<expression  list.] [FOR<condition>] [WHILE <condition>]

Examples:
? 17 - 11 * 100         returns the result of the calculation -  600.

LIST OFF name,address,city,pcode FOR country = ’U.K’
                        lists the contents of the fields specified for records referring to U.K.
                        The OFF means the record number will not be printed.

LOCATE REST FOR name = ’Smith’
                        search the current database from the current point to the end of file
                        and return the record number of the first record with a name field of
                        Smith.

CONTINUE  continue the locate and return the record number for the next
                        occurrence of name = ’Smith’.
```

This section lists commands available in dBASE III PLUS no further instruction is offered here, it is expected that those using the command language will refer to the reference manual available from Computer Reception.

?/?? <expression list>
display evaluate an expression

expression>   [PICTURE <clause>]]

[GET <variable>   [PICTURE <clause>]   [RANGE <low>,<high>]]]/[CLEAR]
display information in a specific format at the co-ordinates given.

AR]   TO <row2,col2>   [DOUBLE]
draws/clears a box

ACCEPT [<PROMPT>] TO <memvar>
prompt for keyboard entry

APPEND [BLANK]
add record(s) to the active database.

APPEND FROM <filename/?> [FOR <condition>]   [[TYPE]<file type>]
copy data from an existing file to the end of the database in use

ASSIST
use the Assistant Menu.

AVERAGE [<expression list>   [<scope>]   [WHILE <condition>]
[FOR <condition>]   [TO <memvar list>]
compute the arithmetic mean.

BROWSE [FIELDS <field list>]   [LOCK <expN>]   [FREEZE <field>]   [NOFOLLOW]   [NOMENU]
[WIDTH <expN>]   [NOAPPEND]
full screen viewing of records allowing editing and appending.

CALL <module name>   [WITH <expC> / <memvar>]
calls a loaded binary file.

CANCEL
stop execution of a command file.

CHANGE [<scope>]   [FIELDS <field list>]   [WHILE <condition>][FOR <condition>]
full screen editing. Identical to EDIT.

CLEAR
erases the screen

CLEAR ALL
close all open database files, release all memory variables

CLEAR FIELDS
releases field lists created by SET FIELDS TO

CLEAR GETS
clears all pending GETS

CLEAR MEMORY
release all memory variables

CLEAR TYPEAHEAD
empty type-ahead buffer before requesting keyboard input.

CLOSE <filetype>
close files of a specified type where the file type refers to one of the following **-** ALL,ALTERNATE, DATABASES, FORMAT, INDEX, PROCEDURE.

CONTINUE
continue a search specified in the last LOCATE command

COPY FILE <file1> TO <file2>
copy file1 to file2

COPY STRUCTURE TO <filename> [FIELDS <field list>]
copy the structure of the file in use

COPY TO <new file>[<scope>] [FIELDS <field list>] [FOR <condition>]
[WHILE <condition>[[TYPE] <file type>]
copy all/part of the database in use to a new file.
Options for <file type> are **-**

DELIMITED [WITH BLANK/<delimiter>]for ASCII text files	
SDF	for system data format ASCII file
DIF	for VisiCalc format
SYLK	Multiplan spreadsheet format
WKS	Lotus 1-2-3 FORMAT

COPY TO <new file> STRUCTURE EXTENDED
create a database file containing details of the structure of the existing database. The fields in the file will be FIELD_NAME, FIELD_TYPE, FIELD_LEN, and FIELD_DEC. This file can be used to create a new database using the CREATE FROM command.

COUNT [<scope>] [WHILE <condition>] [FOR <condition>] [TO <memvar>]
counts the number of records in the active database matching the conditions specified.

CREATE <.dbf filename>
create a new database.

CREATE <.dbf filename> FROM <structure extended file>
create a new database using the structure specified in the structure extended file.

CREATE LABEL <.lbl filename>/?
create a new label form file

CREATE QUERY <.qry filename>/?
create a query file to limit the records accessed in the database in use. Only those records meeting the conditions will be accessed.

CREATE REPORT< .frm filename>/?
create a report file specifying layout for printed reports. The page size and layouts are specified in menus. Column totals can be accumulated and printed. Fields from the active database or any associated databases can be printed.

CREATE SCREEN <.scr filename>/?
create a customised screen for data input using APPEND or EDIT mode. This allows you to have instructions on the screen when entering data, and to put more than one field on the same line.

CREATE VIEW <.vue filename>/?
Creates a view file defining the relationships between database(s), index files etc.

CREATE VIEW <.vue filename> FROM ENVIRONMENT
builds a view file from the working environment. Stores the current settings and relationships.

DELETE [<scope>] [WHILE <condition>] [FOR <condition>]
marks records for deletion

DELETE FILE <filename>/?
delete a file

DIR [<drive:>] [<path>\] [<skeleton>]
show files on the specified drive.

DISPLAY [<scope>] [FIELDS] <expression list>] [WHILE <condition>] [FOR <condition>] [[OFF] [TO PRINT]
displays up to 20 records at a time on the screen or the printer

DISPLAY HISTORY [LAST <expn>] [TO PRINT]
displays the commands in the history buffer.

DISPLAY MEMORY [TO PRINT]
display details on active memory variables

DISPLAY STATUS [TO PRINT]
show the status of all active databases

DISPLAY STRUCTURE [TO PRINT]
displays the structure of the file in USE.

DO <filename> [WITH <parameter list>
executes a program/procedure.

DO CASE
CASE <condition>
<command>
[OTHERWISE]
<commands>

ENDCASE
allows execution of commands when a certain condition is met.

DO WHILE <condition>
<commands>

ENDDO
allows execution of commands while a condition is true.

EDIT [<scope>] [FIELDS <list>] [WHILE <conditions>] [FOR <conditions>]
full screen editing of the database in use

EJECT
advance the paper to top of page

ERASE <filename>/?
delete a file

EXIT
transfer control to the ENDDO of a DO WHILE statement

EXPORT TO <filename>   [TYPE] PFS
build a pfs file from a dBASE III PLUS file, all ancilliary files in use will be exported into the one file.

FIND <character string>/<n>
find the first indexed record matching the character string

GO/GOTOBOTTOM/TOP/<expN>
position record pointer to a specific record.

HELP [<keyword>]
access the help file for assistance.

IF <condition>
<commands>

[ELSE]   _
<commands>

ENDIF
conditional execution of commands

IMPORT FROM <filename> [TYPE] PFS
creates database and ancillaries from the pfs file.

INDEX [ON <key expression> TO <.ndx filename>   [UNIQUE]
create an index file for the database in USE, the records will then appear in the order specified in the key.

INPUT [<prompt>]   TO <memvar>
keyboard entry of a memory variable

INSERT [BLANK]   [BEFORE]
insert a record in a specific position

JOIN WITH <alias> TO <newfile> FOR <condition> [FIELDS <fields list>]
combine records and fields from two databases.

LABEL FORM <.lbl filename>/? [<scope>]   [SAMPLE]   [WHILE] <condition>]
[FOR <condition>]   [TO PRINT/TO FILE <filename>]
print labels using the label form file.

LIST [OFF] [<scope>]   [<exp list>]   [WHILE <condition>]   [FOR] <condition>]   [TO PRINT]
list/print records for file in USE.

LIST HISTORY [LAST <expN>] [TO PRINT]
display/print commands stored in the history buffer.

LIST MEMORY [TO PRINT]
display/print the memory variables

LIST STATUS [TO PRINT]
display/print information on current session.

LIST STRUCTURE [TO PRINT]
display/print structure of file in USE.

LOAD <binary filename>[<.extension>]
load a binary program file to memory

LOCATE [<scope>]   [WHILE <condition>]   [FOR <condition>]
finds first record satisfying condition specified.

LOOP
transfer control to start of DO WHILE loop

MODIFY COMMAND <filename>
create/modify a command file.

MODIFY LABEL <.lbl filename>/?
create/edit a label form file

MODIFY QUERY <.qry filename>/?
modify a query file

MODIFY REPORT <.frm filename>/?
create/edit a report form file

MODIFY SCREEN <.scr filename>/?
modify a customised screen format file.

MODIFY STRUCTURE
amend the structure of the file in USE.

NOTE/*<undelimited character string>
add comments to a program file.

ON ERROR/ESCAPE/KEY <command>
execute the specified command when an error occurs or the escape key/any key is pressed

PACK
permanently removes records marked for deletion from the database in USE.

PARAMETERS <parameter list>
specify memory variables to be passed by DO...WITH

PRIVATE [ALL [LIKE/EXCEPT <skeleton>]]/[<memory variable list>]
allows creation of memory variables with the same name as those in a higher level program.

PROCEDURE<procedure name>
define the start of a new procedure

PUBLIC <memory variable list>
allow memory variables to be used/altered by other dBASE programs.

QUIT
end dBASE III PLUS session closing all files.

READ [SAVE]
allows entry of data in GET fields

RECALL [<scope>] [WHILE <condition>] [FOR <condition>]
reinstate records marked for deletion.

REINDEX
rebuilds active index files

RELEASE <memvar list>/[ALL [LIKE/EXCEPT <skeleton>]]/[module]
deletes memory variables or removes a loaded module from memory.

RENAME <current filename> TO <new filename>
rename a file

REPLACE [<scope>] <field> WITH <exp>[,<field2> WITH <exp2>,...][WHILE <condition>] [FOR <condition>]
replace field(s) with a new value(s).

REPORT FORM <.frm filename>/?<scope>] [WHILE <condition>]FOR <condition>] [PLAIN] [HEADING <exp>[NOEJECT] [TO PRINT] [TO FILE <filename>] [SUMMARY]
display/print/write to file a tabular report.

RESTORE FROM <.mem filename> [ADDITIVE]
retrieve memory variables from a memory file.

RESUME
resume processing of a suspended program

RETRY
error recovery option **-** re-execute the command.

RETURN [TO MASTER]
the last command in a program or procedure.

RUN <command>
execute a DOS command or executable program

SAVE TO <.mem filename> [ALL LIKE/EXCEPT <skeleton>
save current memory variables

SEEK <expression>
find the first record whose key matches the expression.

SELECT <work area/alias>/?
activate the specified work area

Note: in SET commands the default is shown in upper case letters.

SET
set dBASE III PLUS control parameters

SET ALTERNATE on/OFF
start/stop sending output to disk.

SET ALTERNATE TO [<filename>]
creates type .TXT file for saving screen output

SET BELL ON/off
turn bell on/off.

SET CARRY on/OFF
copies data from last record to new record in append mode.

SET CATALOG ON/off
adds/does not add new files to the catalogue file, set OFF until SET CATALOG TO is used.

SET CATALOG TO [<.cat filename>]/?
creates /opens the specified catalogue.

SET CENTURY on/OFF
shows/does not show century in date displays.

SET COLOR ON/OFF
toggle between colour/monochrome, default depends on the system

SET COLOR TO [<standard>[,<enhanced>] [,<border>] [<,background>]]
set screen display attributes.

SET CONFIRM on/OFF
press/do not press ↵ to move to next field during edit mode.

SET CONSOLE ON/off
send/do not send screen output from commands.

SET DATE AMERICAN/ANSI/BRITISH/ITALIAN/FRENCH/GERMAN
determine format of date fields, default is normally AMERICAN.

SET DEBUG on/OFF
error location, ON sends output of SET ECHO commands to the printer.

SET DECIMALS TO <expN>
set minimum number of decimal places displayed

SET DEFAULT TO <drive>
select the default drive

SET DELETED on/OFF
when ON records marked for deletion are not processed.

SET DELIMITERS on/OFF
use characters delimiting fields when ON.

SET DELIMITERS TO [<expC>/DEFAULT]
defines character(s) to delimit fields, default character is colon :.

SET DEVICE TO PRINT/SCREEN
route results of @...SAY commands, default is screen.

SET DOHISTORY on/OFF
record/do not record command files in HISTORY.

SET ECHO on/OFF
display/ do not display command lines.

SET ESCAPE ON/off
allows program interruption when the <ESC> is pressed.

SET EXACT on/OFF
requires/does not require exact comparisons of character strings.

SET FIELDS on/OFF
use/do not use the specified field list.

SET FIELDS TO [<fields list>/ALL]
define fields to be used with the active database.

SET FILTER TO [FILE <.qry filename>/?]   [<condition>]
use a query file or specify a condition to be met to determine the records to be viewed. If no parameters are entered the whole of the database in use is viewed.

SET FIXED on/OFF
if ON is used then the number of decimal places displayed is determined by the previous SET DECIMALS TO command.

SET FORMAT TO [<.fmt filename>/?]
open a format file, to use customised display

SET FUNCTION <function key number> TO <expC>
program a function key, defaults are shown in the table in the dBASE standards below.

SET HEADING ON/off
show/do not show column field names in LIST or DISPLAY mode.

SET HELP ON/off
prompt/do not prompt for help on error.

SET HISTORY ON/off
store/do not store previous commands in history buffer.

SET HISTORY TO <expN>
set the number of previous commands to be stored, default is 20. Up to 16,000 can be stored.

SET INDEX TO [<.ndx file list>/?
opens the specified index files

SET INTENSITY ON/off
use/do not use enhanced display

SET MARGIN TO <expN>
set the left print margin, default is zero.

SET MEMO WIDTH TO <expN>
set width of memo field output. Default is 50.

SET MENUS ON/off
shows/does not show a menu during full screen commands.

SET MESSAGE TO <cstring>
display a user defined message on the bottom of the screen when SET STATUS IS ON

SET ODOMETER TO <expN>
set the update interval for record count displays, default is 1.

SET ORDER TO [<expN>|
choose a master index file from the open index files. Zero means use the file in the unindexed order, the number entered refers to number of the index file in the list of files specified (1-7).

SET PATH TO [<path list>]
specify a search path

SET PRINT on/OFF
send/do not send output to the printer.

SET PRINTER TO <DOS device>
select device for printer to one of LPT1, LPT2, LPT3, COM1, COM2. On the Novell network you would normally be connected automatically.

SET PROCEDURE TO [<procedure filename>]
open a procedure file, if no procedure name is specified the current procedure file is closed.

SET RELATION TO [<key>/RECNO( )/ <expN> INTO <alias>]
link databases by a common key

SET SAFETY ON/off
asks if OK when a file is to be overwritten.

SET STATUS ON/off
status line is displayed.

SET STEP on/OFF
debugging tool, when on program pauses after every command.

SET TALK ON/off
display/suppress results of commands on the screen.

SET TITLE ON/off
prompts/does not prompt for title when adding a new file to the catalogue.

SET TYPEAHEAD TO <expN>
the number of characters you can type ahead, default is 20.

SET UNIQUE on/OFF
when ON the first/all records with the same key are indexed.

SET VIEW TO <.vue flename>/?
open a view file

SKIP [<expN>
[+/-n] move forwards/backwards through the file in USE.

SORT <scope> TO <new filename> ON [/A] [/C] [/D] [,<field2> [/A] [/C] [/D]...] [WHILE <condition>] [FOR <condition>]
creates new database file in sorted order, use /A for ascending, /D for descending order. /C differentiates between upper and lower case. Default is /A, up to 10 fields can be specified as the sort key.

STORE <exp> TO <memvar list>[,<memvar list>]
store an expression in memory variable(s).

SUM [<scope>] [<exp list>] TO [<memvar list>] [WHILE <condition>]
[FOR <condition>]
sums fields in the active database.

SUSPEND
debugging tool, option invoked by pressing <ESC> and choosing suspend. You can then amend a command in the HISTORY buffer and execute the RESUME program execution.

TEXT <text> ENDTEXT
simple method of sending text to screen/printer

TOTAL ON <key field> TO <filename> [<scope>] [FIELDS <field list>]
[WHILE <condition>] [FOR <condition>]
create a summary database of totals

TYPE <filename> [TO PRINT]
display file contents

UPDATE ON <key field> FROM <alias> REPLACE <field> WITH <exp>
[,<field2> WITH <exp2>...] [RANDOM]
modify fields in active database for records with matching keys

USE [<filename>/?] [INDEX <index file list>] [ALIAS <alias name>]
open the database to be used and its associated index files.

WAIT <prompt> [TO memvar]
halts processing until the key specified is depressed. If no prompt is specified the prompt ’Press any key to continue...’ is displayed.

ZAP
deletes all the records in the active database.

[<command>]&& <text>
add comment to same line as command

7 dBASE III Functions

These functions enhance commands. Functions allow manipulation, evaluation and conversion of data. Some work only on a specific data type. The main classifications are Date, Character Manipulation, Mathematical and Conversion. All functions, except the & (macro substitution), are followed by parentheses, even though in several cases the function does not require input, for example EOF( ).

7.1 Memory Variable Function

&<character variable> [.<cstring>]
macro substition obtain the contents of a character memory variable.

7.2 Character Manipulation

The following functions are used to manipulate character string data.

Example: The expression in brackets always refers to a character string, a numeric string is often required as several commands work on a specified number of characters within the character string.

If a field named FORENAME is 20 characters long and holds Frederick
and SURNAME is 20 characters long and holds Jones

```
    ? FORENAME,SURNAME
        would return Frederick          Jones

    ? TRIM(FORENAME),SURNAME
        would return Frederick Jones - the comma generates a single space

    ? TRIM(FORENAME)+SURNAME
        would return FrederickJones - the + means no spaces between.
```

AT(<expC>,<expC>) show position of substring within second string.

LEFT(<expC>,<expN>) return specified number of characters from character string.

LOWER(<expC>) convert to lower case.

LTRIM(<expC>) remove leading blanks from string.

REPLICATE(<expC> <expN>) repeat the string a specified number of times.

RIGHT(<expC>,<expN>) returns the specified number of rightmost characters.

RTRIM(<expC>) remove trailing blanks.

SPACE(<expN>) generate specified number of space characters.

STUFF(<expC1>,<start position>, <number of characters>,<expC2>) replace a portion of a character string

SUBSTR(<expC>,<start position> [,<number of characters>])
remove specified characters from string.

TRANSFORM(<ecp1>,<expC>)
uses PICTURE format on data.

TRIM(<expC>) removes trailing blanks from character string.

UPPER(<expC>) convert to upper case.

7.3 Date Functions

Example: The expression in brackets must be a valid date.

? CMONTH(03/30/91)returns the response March

STORE CMONTH(DATE()) to mthname
store the name of the month from the system date to memory variable mthname.

CDOW(<expD>)	obtain name of the day of the week
CMONTH(<expD>)	obtain the name of the month
CTOD(<expC>)	convert character field to date field
DATE()	returns the system date
DAY(<expD>)	returns the numeric day of the month
DOW(<expD>)	returns number for day of week (Sunday = 1)
DTOC(<expD>)	convert date field to character field
MONTH(<expD>)	returns number of month
YEAR(<expD>)	returns four digit year code.

7.4 Conversion Functions

Used to convert one type of data to another

Example: Convert character field to numeric. If a character is non numeric the value zero is returned.

STORE VAL ('1234.56') TO x	x would hold the number 1234.56
STORE VAL ('ABCDE') TO x	x would hold 0.0

ASC(<expC)	convert character to ASCII
CHR(<expN>)	convert ASCII number to character
VAL(<expC>)	convert character to numbers

7.5 Identification Functions

Used to obtain information about the current environment

Example: Find the name of the database file in use.

? DBF()	display the name of the current database.
Filename = DBF()	store the name of the current file in use in the field Filename.

DBF( )	returns the name of the database in use.
FIELD(<expN>)	returns the field name for the specified field number.
FKLABEL(<expN>)	returns the name of the function key
FKMAX( )	returns the number of programmable function keys
GETENV(<expN>)	returns the contents of an operating system variable.
NDX(<expN>)	returns names of active index files.
OS( )	returns the name of the current operating system.
VERSION( )	returns the dBASE III PLUS version number.

7.6 Input Functions

Returns information about key depressions. The code numbers are not given in this document you will need to refer to the dBASE III PLUS manual for more information.

Example: READKEY returns a code number representing the key pressed on exiting from a full-screen command. Eg. CTRL-W to leave edit mode.

IF READKEY( ) = 270	If edit mode was exited with the Ctrl-W or
.....	Ctrl-End command then process the commands
ENDIF	included in the IF...ENDIF.

INKEY( )	returns code for most recent key pressed during execution.
READKEY( )	return exit key code when leaving full screen mode.

7.7 Mathematical Functions

For use in evaluating mathematical expressons.

Example:return the natural logarithm.

? LOG(x * y)	returns the log of the expression x * y.

ABS(<expN>)	absolute value of numeric expression
EXP(<expN>)	return the value of the exponent.
INT(<expN>)	returns the integer.
LOG(<expN>)	returns natural logarithm.
MAX(<expN>),(<expN>)	returns the larger of two numbers.
MIN(<expN>),(<expN>)	returns the smaller of two numbers
MOD(<expN>),(<expN>)	returns the remainder when dividing 2 numbers.
ROUND(<expN>),(<expN>)	round off to specific number of decimal spaces.
SQRT(<expN>)	returns the square root of a positive number.

7.8 Specialised Test Functions

```
Example:Return Logical True if the end of file is found.

DO WHILE .NOT. EOF( )                              execute the DO loop until End File Marker
……                               is found
ENDDO

? EOF( )                         .F. if not at End of File, .T. if at EOF
```

BOF( )	beginning of file
COL( )	column position of the cursor
DELETED( )	identify records marked for deletion.
DISKSPACE( )	returns the number of bytes free on default drive.
EOF( )	end of file
ERROR( )	returns error number relating to error condition.
FILE(<expC>)	returns true if the specified file exists.
FOUND( )	returns true if FIND/SEEK/LOCATE/CONTINUE successful
IIF(<expL>,<exp1>,<exp2>)	shortcut for IF...ENDIF
ISALPHA(<expC>)	returns true if expression begins with alpha character
ISCOLOR( )	returns true if a color graphics card is installed
ISLOWER(<expC>)	returns true if expression begins with lower case.
ISUPPER(<expC>)	returns true if expression begins with upper case.
LEN(<expC>)	returns number of characters in the string.
LUPDATE( )	returns date of last update of database file in use.
MESSAGE( )	returns the error message
PCOL( )	returns current column position on the printer.
PROW( )	returns the current row on the printer.
RECCOUNT( )	returns the number of record in the database in use.
RECNO( )	returns the current record number.
RECSIZE( )	returns the current record size.
ROW( )	returns the row number of the current cursor position.
TIME( )	returns the system time hh:mm:ss
TYPE(<expC>)	evaluate expression type.

8 dBASE III Standards

8.1 File Types

File contents are identified by the file extension. dBASE III PLUS allocates the extension automatically. You allocate a file name of up to 8 characters. You can override the automatic file extension by typing in one yourself but generally it is advisable to use the defaults.

You can have up to 1,000,000,000 records in a file, each record can be up to 4,000 bytes long and can contain a maximum of 128 fields. Up to 15 files of any type, 10 of which can be databases, can be open. Up to 7 index files and 1 format file can be attached to one database file.

File Ext	File Type	Description
.bin	Binary	assembly language programs converted to binary form
.cat	Catalogue	store the names of files used together in a catalogue. Eg. your database file together with the index and query files used regularly
..prg	Program	dBASE programs created with the MODIFY COMMAND
.dbf	Database	main database holding the data in records and fields
.bak	Backup	Backup of a database file made when the structure of the file is modified
.dbt	Memo	test files containing the contents of memo fields.
.tbk	Memo backups	backup of the memo file following modification of file structure
.fmt	Format	your own display format for the database - used for printing and screen display
.ndx	Index	view the data in the order specified by the index
.lbl	Label	information needed to print labels
.mem	Memory	save contents of memory variables
.qry	Query	display only those records meeting the conditions specified in this file
.frm	Report Form	information needed to print a report in a specific format
.scr	Screen	information needed to customize the screen format
.txt	Text	ASCII characters files
.vue	View	contains field names to be displayed from one or more specified databases.

8.2 Field Types

Field names can be up to 10 characters long and begin with a letter. Spaces are not allowe

Field Type	**Size**	**Description**
character	maximum 254 bytes	may include letters, numbers, and punctuation symbols, as well as numbers that will not be calculated, such as phone numbers or STD codes.
numeric	maximum 19 bytes	may include only numbers that can be calculated, a decimal point, and a leading plus (+) or minus (-) sign. Numeric entries represent data that is to be calculated, do not use this data type for phone numbers or STD codes.
date	fixed 8 bytes	a date. dBASE III PLUS specifies the format: for example, 09/28/89 for September 28 1989. **Note:** the format is American - you can set it to British at command level before beginning your session.
logical	1 byte	a true/false or yes/no response (for example, to the question: "PAID?").
memo	maximum 5000 bytes Memo Field fixed 10 bytes	an entry containing text of up to 5,000 characters. It can be created with a text editor or word processor and is stored in a separate file on your disk. dBASE III PLUS automatically gives the field a length 10.

8.3 Command Syntax and Use

The Command line has a maximum length of 254 characters. Commands are entered at the dot prompt. Each command begins with the instruction or **verb** and the general syntax is:

VERB [<scope>]<expression list.] [FOR<condition>] [WHIILE<condition>]

Code	**Description**
VERB	is the dBASE command
[]	optional item - do not type the square brackets.
<>	you enter the item - do not type the angle brackets
<scope>	defines the number of records the command applies to. RECORD <n> refers to a specific record number. NEXT <n> refers to the next 'n' records. ALL refers to the whole of the database in use. REST refers to the records from the current position to the end of the file.
<expression list>	one or more expressions separated by commas, contains field names, memory variables, constants, functions and operators.
FOR	refers to the records meeting the condition following.
WHILE	refers to the records WHILE the following condition is true.
<condition>	is a comparison of items for instance: Name='Jones' or a logical statement like: .NOT.EOF()

8.4 Operators

There are four types of operators used on differing types of data

Mathematical Operators - used on numeric data

+	Addition
-	Subtraction
*	Multiplication
/	Division
** or ^	Exponentiation
()	Grouping

Relational Operators generating logical results True/False, used on character or numeric data, as long as both fields in the individual comparison are the same.

<	Less than
>	Greater than
=	Equal
<> or #	Not equal
<=	Less thanor or equal
>=	Greater than or equal
$	is contained within (a substring of)

Logical Operators give a logical result on comparing two expressions

.AND. .OR. .NOT. ( )	Logical and Logical or Logical not (eg DO WHILE.NOT.condition) Parenthesis for grouping

String Operators join character strings

+ **-**	Concatenation **-** join two or more strings Join two or more strings, move trailing spaces to the end of the single string

8.5 Memory Variables

Data relating to a database can be stored in a memory variable. This could be temporary data (information from a previous record) or more permanent information (counts, totals etc). These variables are used in program applications as a means of control. They can be saved to a file (.MEM) and read in to another program if required.

Memvar names: can be up to ten characters long, must begin with a character and can contain numbers and the underscore character. Do not call a memory variable by a command name.

Memvar Types: can be character, date, numeric or logical.

Memvar Sizes: you can have up to 256 active variables with a maximum combined size of 6,000 bytes.

Memvar Type	**Size**
Character	maximum of 254bytes + 2bytes
Date	8bytes + 1byte
Numeric	8bytes + 1byte
Logical	1byte + 1byte

8.6 Function Key Defaults

The function keys can be reprogrammed as required using the SET FUNCTION command. The exception is F1 which always accesses the help file. The defaults are shown in the table below.

Key	**Value**	**Key**	**Value**
F1	help	F6	display status
F2	assist	F7	display memory
F3	list	F8	display
F4	dir	F9	append
F5	display structure	F10	edit