

# TECHNOTES

TECHNICAL SUPPORT NEWSLETTER

## dBASE IV

No. 87 May, 1992

### Q&A

From your BBS, I've downloaded a utility which consists of a couple of assembly-language files (.BIN) but I don't understand how to prepare them for use with dBASE IV. Do I link or compile them in some way?

.BIN files don't need preparation. In your dBASE IV program, use the LOAD command to load a .BIN file into memory. You may have up to 16 loaded into memory at once. Use the CALL command or the CALL() function to run the desired .BIN file. If you need the .BIN routine to return a value, the CALL() function is preferred. When you want to release them from memory, use the RELEASE MODULE command.

More Q&A on page 10

### FEATURED IN THIS ISSUE

Closest Thing to PRFect .....	14
Depth Charges .....	17
Memos Under Glass .....	29

### DEPARTMENTS

Q&A .....	10
-----------	----

FOR THE USERS OF dBASE IV

## The Undocumented Mouse

BILL RAMOS

There is undoubtedly much to be said about the "pet of choice" for PC users now that version 1.5 has arrived

dBASE IV version 1.5 represents the first time you can "mouse around" inside of dBASE IV without an external utility. But the use of a mouse had to have as small an impact on the core of dBASE IV as possible; so don't look for sizable BROWSE windows, scroll bars, and clipboards. What you will find is a no-nonsense implementation of the mouse that allows you to click through menu pads, popups, BROWSE, EDIT, READ and WAIT commands.

There is more to say about mouse implementation than what is within the pages of the documentation. In this article you'll go beyond learning how to use the mouse in your programs. You'll also find two .BIN files and several UDFs that will give your programs more control over the mouse.

### Making Your Programs Mouse Friendly

A typical application written in dBASE IV involves a number of diverse elements. But before you can use the mouse with your programs, you have to know where the traps are.

#### Pause Until Click

As dBASE programmers know, a common practice for pausing in a program is to use the INKEY() function. Unfortunately, INKEY() does not respond to a mouse click. Hold on, the WAIT command does. PROCEDURE Err\_Box in the dBASE sample program LIBRARY.PRG demonstrates how you can display a message in a box and allow the user to either press a key or click the mouse to continue processing.

Continued on page 3

BORLAND

# "Temporary" Explanations

and other brief discourses in the world of database usage

I've read in previous TechNotes issues about references to dBASE IV using temporary files for various operations which will speed things up on a network. Exactly what operations make these files? We don't use the Control Center at all — only programmed applications. Does this still affect us in our situation?

dBASE IV creates temporary files the moment you start the software. So, regardless of whether you run a program or the Control Center you will be using temporary files. These files start with the letters TMP and have file extensions beginning with a \$.

Also, certain commands will create temporary files regardless of whether they are issued from the Control Center or via a program. Two such files with extensions .T44 and .W44 are temporary work files used in conjunction with the SORT or INDEX commands. They are normally erased automatically once the SORT or INDEX operation is complete. They may however exist on your hard drive if those operations, for any reason, did not complete normally. If you have such files on your hard drive, you may safely erase them after quitting dBASE IV.

## The Evolution of VALID and REQUIRED

I'm creating a screen format file via the Control Center and am trying to set a particular field so that the value entered must pass several tests before the information is accepted into the data file. I've specified my VALID clause but I noticed that the VALID testing is only done if the user enters something in that field. I'd like to force the user to enter something as well as accept it only if all the criteria are met. I've read the manual and know that I need to specify the REQUIRED clause but I'm at a loss on how to achieve this through the user interface. I can modify the .FMT file and make that change but I'd prefer to leave this as a last resort. Is there any way to get my screen file to

add the REQUIRED keyword to a particular field which has a VALID clause as well?

The version of FORM.GEN which came with the x509 version of dBASE IV version 1.1 does have the ability to add the REQUIRED keyword to the VALID and RANGE clauses. To do so, create a DOS environmental variable by typing the following at the DOS prompt PRIOR to starting dBASE IV:

```
SET DTL_REQ=ON
```

Once in dBASE IV, regenerate the dBASE IV format file code by erasing the corresponding .FMT file, modifying the screen file layout, and saving. dBASE IV checks to see if the DTL\_REQ variable exists and if it equals ON, all VALID clauses will automatically be set to REQUIRED as well. Unfortunately, this DTL\_REQ affects all VALIDs and RANGES and cannot selectively affect one field and not others.

In dBASE IV version 1.5, two additional options have been added to the Edit Options submenu. If a value is specified on either upper or lower range, you have the option to also specify whether this range must ALWAYS be met (which is the same functionality as the REQUIRED keyword). A similar setting is available for the VALID clause as well. These options show up in the Edit options menu as: **Range must always be met** and **Value must always be valid**, the default for which are both NO.

## Stamping out Generics

I'm printing to an HP LaserJet and have selected one of the HP drivers through the DBSETUP program. I've also selected this driver to be the default driver on start-up of the software. But each time I create a report and print it, the printer model which shows is GENERIC.PR2 and my type styles don't print properly. Why?

First, perform a double-check. When you enter

dBASE IV, press F5 or type **DISPLAY MEMORY** from the dot prompt, you should see the appropriate system memory variable called \_pdriver set to your printer driver. If it does, then indeed, the printer driver has been set correctly.

When you create a new report or label, dBASE IV needs values for page length, page ejects, printer driver, number of output copies, beginning page number, and so on in order to print your report. These settings are stored in print form (.PRF) files. Unless a particular print form file is specified in the Print menu: Use print form option, dBASE IV uses default values which are stored in a file called REPORT.PRF. If you do not see a value in the Print menu/Use Print Form option, the settings used are being taken from the REPORT.PRF. This print form contains a reference to the GENERIC printer driver, among other default settings. So although you've set the default printer driver to suit your system, in essence, the REPORT.PRF file is *resetting* your setting of default printer driver back to GENERIC!

To correct for this, modify any report or create a new one. Open the **Print:Destination** sub-menu and set the report to the appropriate driver. Close the Destination sub-menu and make any other changes you would like to be your default settings (such as page length). Once completed, select the **Print: Save Settings to Print Form** option. When the print form file name box is presented, it automatically has the name of the report and a .PRF extension. Change the file name to read REPORT.PRF and press Return. Exit from report design but do not save the report. (Don't worry! The print settings were just saved to the REPORT.PRF file.)

Changing these settings will only affect reports which do not already have an associated Print Form setting, in addition to all new reports. You will still have to modify existing reports that are not already set to your HP printer.

The label feature has a corresponding print form file called LABEL.PRF. Performing the above operations similarly to the label print form file will change the default settings for your labels as well.

### Expression Building in the Report Generator

I have a report and I want to print a few words in the Page header of my report for each page of output AFTER the first but not on the first page. I tried to use the IIF() function with PAGENO, one of the predefined fields available, but I got an error message when entering the expression in

### and couldn't get beyond this. Help!

The reason you got the error message (most likely "Variable not found") is that while you're in the report designer, the PAGENO predefined field is *not* available to you when building expressions for calculated fields. (Try pressing Shift-F1 at the point where you're building an expression and you'll notice the calculated, predefined and summary fields are not available.)

You can still accomplish this by creating a calculated field in your report and testing the value of the system memory variable \_pageno within this calculated field. If it is less than 2, print nothing. If it is 2 or higher, print the text you'd like to appear.

For example, let's say you wanted the word "Continued" to appear in the page header at row 2 column 65 on pages 2 through the last page of the report but NOT on the first. Modify your report and place the cursor at row 2 column 65 and press F5 to add a field. On the Calculated panel, press return on <create>. For the expression, press Return then type the following command:

```
IIF(_pageno < 2, "", "Continued")
```

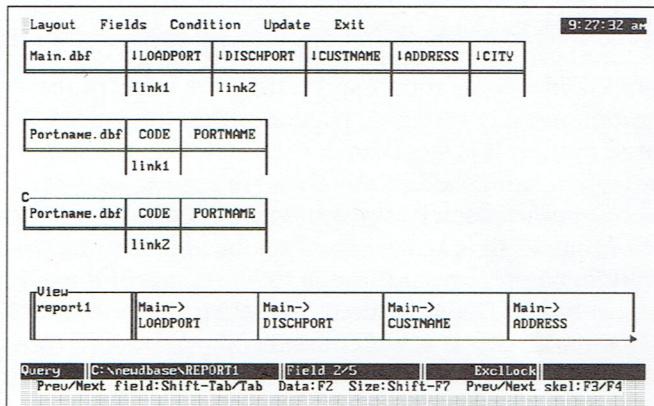
This function tests the system memory variable PAGENO which contains the page number being printed. If less than 2 it returns an empty string. If the value is 2 or greater, it returns the word "Continued". Since this is a calculated field in the context of a report, dBASE will print this returned value.

### Handy and Reusable Files

I'm trying to create a query that links two files together, each containing shipping information. My first file has two character fields which specify the load port and discharge port. The contents of both are abbreviated according to a standard list of valid codes we use. My second file contains all the abbreviated codes for the various ports and the full name of the port. I need to print a report which will output the full name of the port but I can't seem to get this to work in the query. I can get the load port to work, but then the discharge port doesn't. When the discharge port works, the load port is blank. Should I have set my database up differently?

You needn't restructure your database file. What you can do in the query generator is to add both files to the query and create a link between the two using the load port field in the first file and the abbreviated port field in the second. Then, add the second file.

that has the full port names *again* so that you see three data files on your query design screen of which two of them are the same! Create a second link between the discharge port in the first file and the port name in the file that was added again. Add both of the full port names to the view and test the results by pressing F2 to view your data.



## Proportional Blues

I've just set up dBASE IV to print using a proportional font on my HP LaserJet III just as a test. When I'm in the design screen for Reports, I can see my font on the menu and can set various fields and text in the report to this font without apparent problem. When I print though, the report page ejects all over the place and only prints a couple of lines per page. It seems to be okay until it hits one of the words which I've set to the proportional font and then goes a bit haywire. I know that dBASE IV won't calculate proportional font spacing but I still expected the report to print out somewhat normally.

True, dBASE IV does not do any calculations based on proportional spacing but it still should print out somewhat normal looking output. What is affected with the proportional font is the right margin of the page, for better or worse. If you've checked your printer manual and the code setting to the proportional font is correct, check your ending code. Did you use an {ESC} + "E"? If so, this is what is causing the page ejects. These codes *reset* the printer and are intended to be specified at the beginning and ending of a print job to clear any settings currently in the printer and to eject the current page. If issued during a print job, it will do the same, print whatever is in the print buffer currently, eject the page, then continue on with the report. This usually isn't something desired within a report.

Dot matrix printers, for example, have a starting code to set the printer to a particular typeface then an ending code to turn the typeface off. Laser printers do not have equivalent ending codes to turn off a particular font and therefore, you need to choose a style which will be your "normal" style to reset the printer to when you've finished printing the font desired. Since most laser printers user courier 10 as the default font, this is often a good type style to set the printer to as the ending codes for your fonts.

## Indexing in the 90's

I've got dBASE IV version 1.1 and I'm converting some programs from dBASE III PLUS to dBASE IV. I'm having a weird problem once the code is ported to dBASE IV and can't seem to figure it out. I open my data file and create my .NDX index file at the dot prompt. I get an appropriate message saying that dBASE IV indexed all 19 sample records in my database. When I reopen the file with the index using the following command:

```
USE CLIENTS
INDEX ON LASTNAME TO CLIENTS
```

I get a message that the file is in natural order. This is the index file that I JUST created. I've looked at my data with the index file active immediately after I've created the index file and it looks fine, with all my data in the order I expect it to be in. I know I should convert to .MDX tags but from what I've read, the .NDX files should work as well. And now that I've come across this, it's really got me curious...What is going on?

With the changes in dBASE IV in the area of multiple index files, several commands changed slightly. In addition to the INDEX ON and SET ORDER TO commands, the USE command is also affected. It now allows for a parameter telling dBASE IV to use a .MDX file other than the default production index and dBASE is confused regarding whether to look for a .MDX file or a .NDX file. dBASE IV logic interprets that you are instructing it to look for a .MDX file by the name of CLIENTS.MDX but since a tag name was not specified on the command line, it left the file in natural order, hence that message.

Now that the mystery is over, consider using the dBASE IV .MDX tags as they are by far, more efficient and faster than former .NDX files.

## Edit-Able, Delete-Able, Append-Able?

I've heard a lot about the edit-able multi-file views in dBASE IV version 1.5 but nothing more specific. I'd like to be able to modify information contained in several linked files but also be able to add records if a particular record isn't found in the linked set. Is this possible?

No. The edit-able multi-file views are indeed just that — only edit-able. You can make changes to the information presented in your QBE with a few notable exceptions.

- You cannot append or delete records. dBASE IV has no way to determine which file or files to do delete from or append to. For example, a match is found in the child file for a particular record in the parent. Fields from both files are selected in the view and you specify a delete record operation... should dBASE IV delete the record in the master file as well as the child file? Or if only one file is affected, on which should the delete operation be performed? Similarly on an APPEND, to which file should the blank record be added or should it be added to both?
- We also restrict editing to the link fields because to change the values of the key fields on which the relationship is set would cause record pointers between the master and child files to be reset, thus breaking the relation.

## Taking the "Byte" out of your .MDX

My application has several index tags associated for various sorting sequences in the reporting area. Since I can't tell dBASE IV when to perform the re-indexing, and when to wait, I'd like to delete the .MDX file completely. As a test, I deleted the .MDX file and inserted code into my program which would recreate the necessary tags later when they would be needed. When I ran the application, I got an error message that the .MDX file was not found. Is there any way of eliminating the error message?

The reason why you are getting this message is that a particular byte in the data file header is telling dBASE to look for a corresponding .MDX file, and dBASE is unable to find it. This byte appears at position 28 of the data file header and contains a 01H if a production .MDX file exists, or a 00H if not.

If you delete each tag, the .MDX file is deleted when the last tag is deleted and this byte 28 is set to 00H. When opening the file, dBASE IV will know to

not look for a .MDX file for this datafile.

If you delete the file using the ERASE command or by using the DEL command in DOS, you've literally erased the actual file, but the reference still exists within the data file header. To reset this byte, you can use the new low level file functions in version 1.5 to open your file and to reset this one byte.

Assuming that your file is called CLIENT.DBF, the following code will adjust this value appropriately:

```
mhandle = FOPEN("<filename>.dbf", "rw")
```

opens the file with read/write access and stores the file handle number to variable mhandle.

```
? FSEEK(mhandle, 28, 0)
```

moves the pointer in the file to position 28 in the header.

```
? FWRITE(mhandle, chr(0))
```

&& writes 00H at byte 28

```
? FCLOSE(mhandle) && closes the file
```

If you're using an older version of dBASE IV, resetting of this byte from within dBASE IV is not possible by any direct means. One option is to delete each tag using the DELETE TAG command from within dBASE IV. Alternatively, you can check the Borland Download Machine (see back page Masthead for details) for a utility entitled MDXBYTE.ZIP. This utility is a .BIN file which must be loaded and called from within dBASE IV. It modifies the .DBF file so that dBASE IV no longer expects to find an .MDX file and hence you will not get the error message that the production .MDX file cannot be found. Deleting the .MDX file and running this utility against the data file should alleviate the dialog box.

## Compatibility with DOS 5.0

I've upgraded my system to DOS 5.0. I've noticed on the dBASE package that DOS 5.0 isn't indicated as one of the certified operating systems. Are there any known problems with this?

dBASE IV was certified to work with DOS 5.0 last year but this information may not be reflected in the packaging of versions prior to 1.5. Although DOS 5.0 is a larger program than its ancestors, it can be loaded into the high memory area of a 286 machine or above, and thus, will leave more memory for programs such as dBASE IV. It also allows for programs to be loaded in high memory if available and thereby frees up even more memory for dBASE IV. On the other hand, DOS 5.0 may automatically install SMARTDRIVE for you in your CONFIG.SYS.

If this is so, you should not use the dBASE cache. ■