

Bad behaviour

The strange behaviour of FoxPro 2.6a under Windows has not gone unnoticed by concerned developers. Mark Whitehorn offers a forum for complaints, and some sympathy.

There appears to be growing concern among FoxPro developers about the behaviour of FPW 2.6a under Windows 95.

The following arrived from Matthew Cook-McQueen:

"I've been wondering about your opinion of Microsoft's treatment of developers. As you are probably aware, there are some problems with running FPW 2.6a under Windows 95.

"The following ones give a flavour and are explained further in a document available from the Microsoft Software Library; the file name is FW_WIN95.EXE.

1. The FoxPro icon is displayed in the Explorer for a distributed application, even if the developer has included another icon.
2. The FoxPro 2.6 icon is displayed in the Title Bar and Task Bar.
3. The FoxPro 2.6 icon is displayed periodically in a screen's Title Bar.
4. The Close menu does not work.

"Microsoft say this about the close button problem —

'Under Windows 95, commands available on the system menu are also accessible through icons on the right of the title bar. Icons are greyed if the command is disabled to correspond to greyed options on the system menu. The Close icon is always disabled for FoxPro 2.6, regardless of whether Close is enabled.

'Developers can exit FoxPro 2.6 via the Close option on the System menu, using the Exit option on the File menu, or by typing Quit in the Command Window.

Users of FoxPro 2.6 applications can use the Close option on the System menu or methods such as a Quit option included in the application.'

5. Screens with a BorderType of System, Panel, or Double cannot be restored after being minimised.

"Microsoft say:

'A screen which is defined as to have a BorderStyle of System, Panel, or Double cannot be restored once the user minimises it under Windows 95. To work around this problem, change the BorderStyle to None or Single or disable the minimise option.'

"Microsoft appears unwilling to issue a maintenance release (2.6b?). At the same time, it is unreasonable to expect all applications to be converted to Visual FoxPro in order to get round these problems. Kevin McNeish (CompuServe 74504,3723) has been organising a petition on CompuServe regarding these problems, the hope being that Microsoft will eventually issue a bug-fix. He provided the following:

'There have been a few other problems, the most serious being memory leaks. Apparently, if you

(1) Tony Wall's sample database CBDEMO.MDB. This single table list 77 products, complete with details

Synchronised combo-boxing

In the January issue of PCW I included a code fragment from John S Graham. He had identified a problem with combo boxes. He wrote: "If the combo displays a value and the user moves to a different record (using the form navigation buttons), the value in the combo remains static and does not follow the form. My code corrects this so that the combo 'follows the form'."

Tony Wall replied: "The example code appeared to be a little convoluted which prompted me to send in the following example" which is on the cover disk as CBDEMO.MDB. I'm not sure about the relative complexity. I suspect that hard-line coders will prefer John's solution while GUI users will prefer Tony's. Either is excellent if it solves a problem in the easiest way.

continually run a screen with picture radio buttons it exhausts system resources." I contacted Microsoft about this and the official line is:

"We are continuing to sell FoxPro 2.6 for Windows against the usual software industry policy to stop selling a version of a product when the newest version is released.

"As in all development efforts, we have limited development resources available. Releasing a patch to 2.6 for Windows, which isn't even the most current version of the product, would significantly impact the development schedule for our future FoxPro products like Visual FoxPro for Macintosh 3.0, and future Visual FoxPro for Windows products. Hence we opted to post solutions for the five most common areas of interest to help customers work around these issues."

My sympathies lie firmly with Matthew on this one. If Windows 95 is a new OS, then products to run under it (like Visual Fox) are new products, not "newest

versions". If Windows 95 is simply an upgrade of Windows 3.x, then old versions of software (especially those from the company which makes the OS!) ought to run, or at worst be patched until they do.

The electricity meter problem

Last month I described a problem which a colleague and I encountered. It involved matching records for electricity meter readings with the records containing the previous reading from the same meter. I said at the time that we had produced an SQL solution which worked fine but was flawed in terms of the SQL model. In other words, we cheated.

The cheat is hidden in that manipulation of the Counter field. Our solution involved ensuring that the records were in a particular order and numbering them. The number assigned to each record was then incremented by one as the records were copied to another table. Finally, the records in the two tables were joined on those numbers. Described like this, the cheat is clearer; effectively we were using the counter field to find for each record the one above it, which, as with our original solution, offends the relational model.

The solution we have devised is fine in terms of functionality; it works wonderfully and is currently in use. However, as I said at the time, we suspect there is something fundamental we are missing here. There must be a more elegant solution. Any ideas?

Speed glorious speed

Over the last few issues the topic has been speeding up your databases. First we looked at query optimisation. The take home message was that your RDBMS should have a query optimiser which will look after optimisation for you in single queries. If you build a series of queries,

— no nose clamp required

(2) The combo boxes along the top of the form list in numerical or alphabetical order the contents of three fields in all 77 records. By popping down one of the lists and selecting an item, the form will automatically jump to that record

(3) and the other combo boxes update

(4) In addition, if you use the record navigation keys at the bottom of the form to move to the next record, the combo boxes automatically update as well

each based on the previous one, make sure they work in a sensible order — sensible in this case meaning “optimised for speed”.

Next on the agenda for examination for possible speed improvements was the size of the tables and the way the data is disposed between them. Normalising the data is the best strategy if the data in the tables is subject to continual update and addition. When the data is static and only queried, then normalisation may be a mistake. Putting all the data into one table which contains redundant information and is heavily indexed may give a better response time.

It is, of course, quite possible to find that you have both requirements (rapid update and rapid querying) in the same database. You can resolve these conflicting requirements by splitting the database. Leave a fully normalised database for data entry and update, but clone it every night, via one or more queries, to form a single, highly indexed table which can then be used for querying. True, the queries will all yield answers that are one day out of date, but for many queries this is immaterial.

Clearly there are a variety of factors you can balance here — how often you rebuild the mega-table (one a night, once a week, etc), choosing which queries can be run

Table 1	
Table	Records
Post Codes	80,000
Customers	8,000
Orders	12,500
Order Details	51,000
Items	200

against the mega-table and which should be run against the normalised tables. As in any balancing act, trial and error (or “trial by error” as a friend of mine calls it) together with common sense will serve you well.

Finally, possibly the most important factor in speed considerations — indexing. I realise that I will be teaching many of my grandmothers by covering this, but remember that statistic in the January issue: “about 70 percent of those databases had no indexing whatsoever (presumably apart from the Primary keys)”. It implies that only 30 percent of readers can ignore this section with the contented feeling of a job well done.

Given a set of tables like those shown in Fig 5, and the associated query, we would normally expect the tables to have

Tips & Tricks: Postcodes

Discussion on input masks for postcodes and phone numbers appeared several times last year. I recently received the following information from John Douglas re. the current and future formats of postcodes.

The current PostCode is composed of two subfields, Outward and Inward. This is soon to be extended by a third subfield, the Delivery Point Suffix (DPS).

Generic field	Length		Input Mask	Note
	Min	Max		
Outward postcode	2	4	>LAaa	uppercase
Inward postcode	3	3	>0LL	uppercase
Delivery Point Suffix	2	2	>0L	uppercase

Source: Royal Mail Postal Address Book 95-96, About Addresses and Postcodes Customer Barcoding Trial Report & Technical Specification, Issue 2 March 1994

Summary notes UK Postcode Format

Example:

Current P01 3AX max. width 4 + space + 3 = 8
 Future P01 3AX 1F max. width 4 + space + 3 + space + 2 = 11

Elements

- Outward Code
 - Area PO
 - District 1
- Inward Code
 - Sector 3
 - Unit AX
- Delivery Point Suffix
 - DPS 1F

Outward Code

The Outward Code is the first part of the Postcode. This is divided into two parts, the first part being letters which define the Postcode Areas. These letters are derived from the letters of a city, town or district in the area.

The second part of the Outward Code being a number and possible letter. These refer to the District within a Postcode Area. This number can be one or two digits.

Possible Formats Example

ANA P01
 AAN BN1
 AANN SE11
 AANA SW1A
 AN L7
 ANN E17
 AAA don't know of an example

(A=alphabetic A..Z, N=numeric 0..9)

Access Input Mask: >LAaa

Current combinations

Postcode Area approx. 130
 Postcode District(Area + District) approx. 3,000

Postcodes (continued)

Inward Code

The Inward Code is the second part of the Postcode. This is divided into two parts, the first part being a single-digit number defining the Sector within a Postcode District.

The second part of the Inward Code being two letters. These refer to the Unit within a Postcode Sector.

Format

NAA Access Input Mask: >0LL

Current combinations

Postcode Sector (Area + District + Sector) approx. 10,000
 Postcode Unit (Area + District + Sector + Unit) approx. 2 Million

Delivery Point Suffix

The Delivery Point Suffix is the third part of the extended Postcode, this being a single-digit number and one letter. These refer to the Delivery Point within a Postcode Unit.

Format

NA Access Input Mask: >0L

Restrictions

digits not 0 1..9 only
 letters excluding C, I, K, M, O and V

Possible combinations: 9 x 20 = 180

Required combinations: 100

Current combinations:

Delivery Point (Area + District + Sector + Unit + DPS) approx. 25M

Did you know the format was due to change? Will your software be able to cope when it does? Perhaps in the future there will be more consultation between organisations which decide formats like these and the increasingly large user-community that has to handle their “standards” electronically. In addition, porcine avians may soon be commonplace.

primary keys and other indices. Purely in the interests of science, I have removed all indices. The tables' sizes are in Table 1.

I timed the query with no indices and then added the significant indices to each table, reading left to right, re-timing the query after each table was modified. (Table 2.)

Fig 5 The tables and query used for the indexing tests

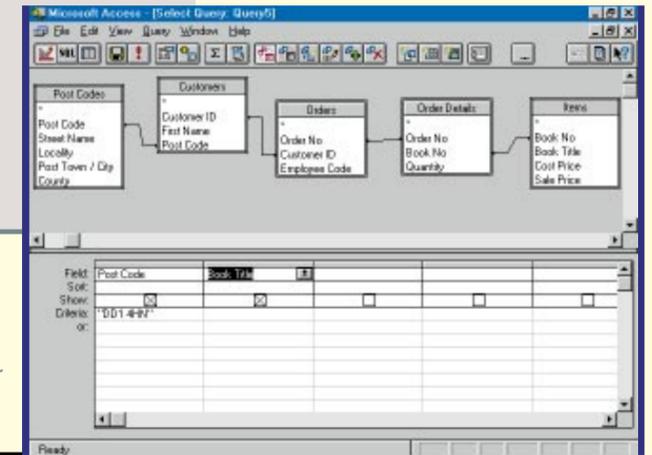


Table 2		
Table with new indices in seconds	Fields indexed	Time for query
None	None	47
POST CODES	[Post Code]	12
CUSTOMERS	[Post Code]	8
ORDERS	[Customer ID]	6
ORDER DETAILS	[Order No]	1
ITEMS	[Book No]	< 1

The important point here is that you need to index fields which will be searched. In this case, that clearly includes fields like POST CODES.[Post Code] since we know that it will be searched for the value “DD1 4HN”. However, fields like CUSTOMERS.[Post Code] also have to be indexed because the query will have to find the relevant records in the CUSTOMER table by searching that field.

In fact, in a database like this, fields like CUSTOMER.[Customer ID] and ORDERS.[Order No] are typically indexed because they would be primary keys. However, in this query they are not searched and therefore indexing them should have no effect. Just to prove the point, I ran the test again with just these two fields indexed and the result was exactly the same as if no indices had been set — 47 seconds.

This is not meant to imply that I think primary key indices are irrelevant; quite the reverse. Most databases have multiple queries run against them rather than the one shown here, and primary keys are the most likely fields to be searched. However, primary keys will be indexed anyway, so which other fields should you set to be

indexed? The obvious answer is every field which is a foreign key (that is, it is the other end of a join from a primary key). In addition, you should index every field that you know will be searched — fields like LastName, Telephone No. etc.

As you can see from the tables, indexing makes a huge difference.

PCW Contact

Mark Whitehorn welcomes readers' correspondence and ideas for the Databases column. He's on m.whitehorn@dundee.ac.uk