A date to remember

Why write chunks of code to make text dates recognisable by Access when you could do it the quick way? Mark Whitehorn shows you how.

John Proud writes on the subject of data conversion:

"I have nearly 8,000 data records in comma delimited form but the date field comes across as text. Other than re-keying the lot (gulp!) I am stumped. Is it possible to convert a text date (such as 23rd December 1997) into a date recognisable by Access? Any offers excluding the above re-keying option would be very welcome." Now, I know I said no more dates, but I also know that I am never going to get away with that in the *Databases* column. In addition, this is really a data conversion rather than a date problem (no, really, it is next to nothing at all to do with dates).

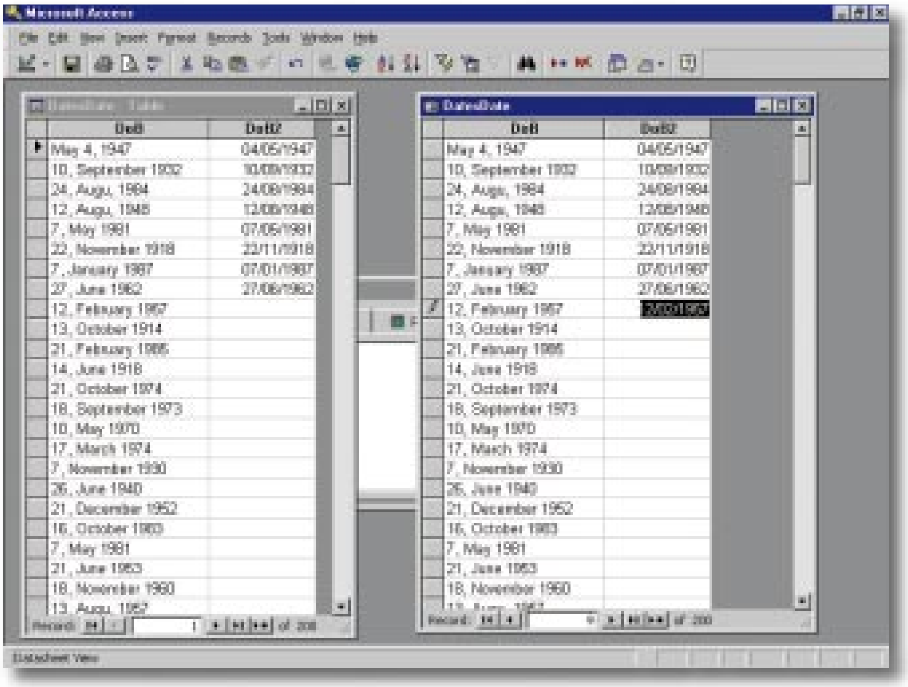
John's dates are in the form of text strings like "23rd December 1979" and Access 97 has a function called CDate that very nearly works on strings such as this, but in fact just happens to fail completely. What it objects to is the "rd". If you pass CDate a string like "23 December 1997" it works perfectly and, what's more, it is sensibly liberal in its interpretation of shortened forms of months. Since the original data comes as text, import it into a text field (called, say, DOB) in an Access table. This will then have 8,000 records.

Now search for "st" and replace with " " and do the same for "nd", "rd" and "th". Anyone really on the ball (I wasn't) will realise that all instances of "August" will now read "Augu", but thanks to CDate's liberal views we can happily ignore this side-effect. Next, add a second field to the table, call it DOB2 and make it of type date. At this point I almost made the mistake of writing a boring chunk of code to start at the first record, look at the string, convert it, write it into the date field and continue thus to the end. But then I thought, why not make it fun? Having been depressed by the idea that you might have to convert all the little charmers by hand, why not set up a form so that you can actually see them all being converted in front of your very eyes? Here's how you do it.

- **Build a form** containing both fields. In the On Got Focus event of the DOB2 field add a single line reading:

```
DOB2=CDate(DOB)
```

- **In the form view**, put your cursor into the DOB2 field and hold down the down arrow key.
- **As the cursor** jumps to each record, the date converts and on it leaps to the next one. This is on the cover CD-ROM as DATESDAT.MDB. I've done a few, but the rest of the 200 are yours to experiment with [Fig 3].



	DoB	DoB2
▶	May 4, 1947	04/05/1947
	10, September 1932	10/09/1932
	24, Augu, 1984	24/08/1984
	12, Augu, 1948	12/08/1948
	7, May 1981	07/05/1981
	22, November 1918	22/11/1918
	7, January 1987	07/01/1987
	27, June 1962	27/06/1962
	12, February 1957	
	13, October 1914	
	21, February 1986	
	14, June 1918	
	21, October 1974	
	18, September 1973	
	10, May 1970	
	17, March 1974	
	7, November 1930	
	26, June 1940	
	21, December 1952	
	16, October 1983	
	7, May 1981	
	21, June 1963	
	18, November 1960	
	13, Augu, 1957	

Record: 1 of 200

	DoB	DoB2
	May 4, 1947	04/05/1947
	10, September 1932	10/09/1932
	24, Augu, 1984	24/08/1984
	12, Augu, 1948	12/08/1948
	7, May 1981	07/05/1981
	22, November 1918	22/11/1918
	7, January 1987	07/01/1987
	27, June 1962	27/06/1962
	12, February 1957	12/02/1957
	13, October 1914	
	21, February 1986	
	14, June 1918	
	21, October 1974	
	18, September 1973	
	10, May 1970	
	17, March 1974	
	7, November 1930	
	26, June 1940	
	21, December 1952	
	16, October 1983	
	7, May 1981	
	21, June 1963	
	18, November 1960	

Record: 9 of 200

▲ Fig 3 The conversion form in action

The underlying message is that data conversion is often a multi-stage process. If you're stymied by something, try breaking the problem down. Suppose, for example, that you use Access 2, where CDate is not an option, but have a friend with Access 97. You could do the conversion in Access 97, export it again as text (but this time as "1/4/1998") and import it back into Access 2 as a date.

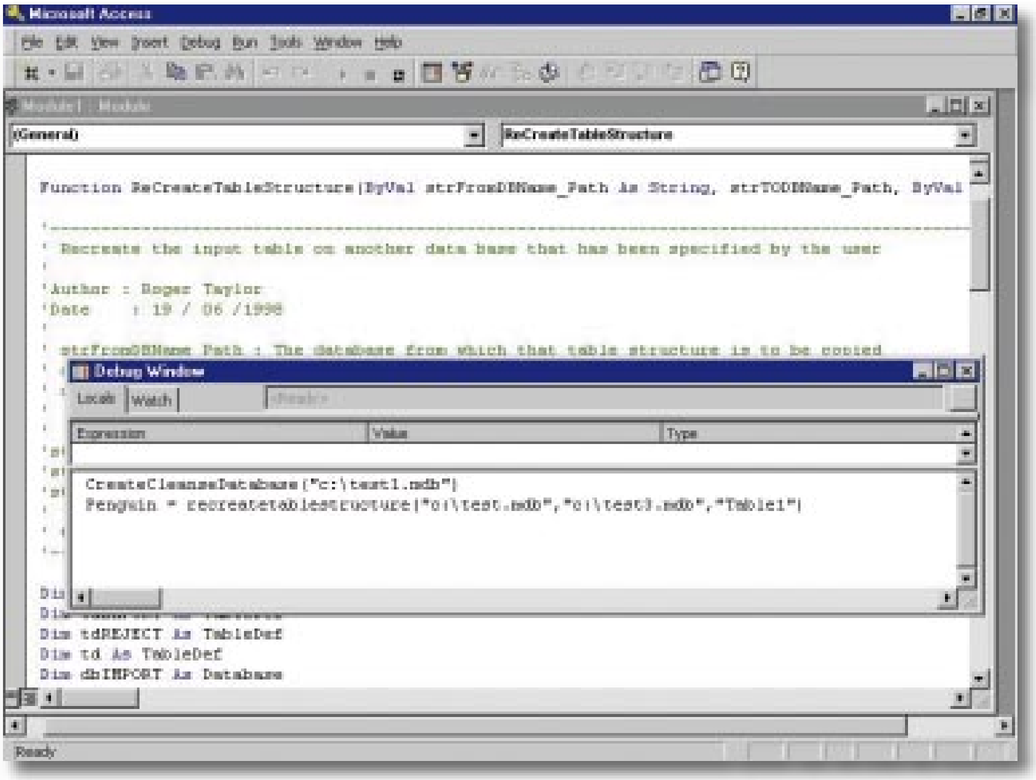
I've done a few, but the rest of the 200 are yours to experiment with

If Access 97 was unavailable, you could perform the manipulations in Excel or by multiple search-and-replaces in Access 2 (replacing instances of "December" with "12/" and so on. Happily, the CDate approach worked for John and he replied: "I transferred all the data and following the find and replace for all 'st', 'nd' and 'rd's to '' the data converted automatically."

Lock and role

Several people have sent emails concerning unexpected locking problems in Access when it is used in a multi-user environment. The following background may be helpful.

If a user is altering a record, it is imperative that other users don't try to change that same record at the same time. This much is obvious. Access locks any record that anyone is editing — a process it calls, misleadingly, record locking. Despite what it says in the manuals, Access doesn't lock records, it locks blocks of the file that contains records. So, if you start editing a particular record, Access will lock the block in the file that contains that record. If any other records happen to be embraced by that block, then they are also locked, to the annoyance and chagrin of others. Diagnosing this problem isn't helped by the fact that, although Access usually displays records ordered by primary key value, the storage order on disk is often unrelated to the primary key value. Thus as I am editing record number 4,234, I might be locking record number 3, number 9 and number 342,445.



```
Function ReCreateTableStructure(ByVal strFromDBName_Path As String, strTODBName_Path, ByVal

'-------------------------------------------------------------------------------
' Recreate the input table on another data base that has been specified by the user
'
'Author : Roger Taylor
'Date    : 19 / 06 /1998
'
' strFromDBName_Path : The database from which that table structure is to be copied
```

```
CreateCleanseDatabase("c:\test1.mdb")
Penguin = recreatetablestructure("c:\test.mdb","c:\test3.mdb","Table1")
```

```
Dim tdREJECT As TableDef
Dim td As TableDef
Dim dbIMPORT As Database
```

▲Fig 1 Roger's functions in action. You have to take my word for it that I have run these since the dialog box doesn't give any feedback, but they worked for me, honest

Access code

Roger Taylor writes:
"I've recently had to write a bit of code that will recreate an Access 97 table structure in another database.
I know that in Access you can use the Transfer function to do this, but nothing like it exists in Visual Basic 5.0. The code is a little ropy, but it does the job. Basically it creates another database which you specify in the Param list and recreates the selected tables structure in that database. Anyway, have fun."
Fun always sounds good, so I've included Roger's code in ROGER.MDB from which the functions can be tested using the debug box [Fig 1]. In addition, for those who are not Access 97 users, it is on the CD-ROM as dbdecd98.txt.

Taken to task

James Smith took me to task with:
"I was a little disappointed by your response to the question of back-issue Hands On articles. Why did you suggest that the gentleman in question spend £10 of his hard-earned money on a CD, when all the cover disks contain 12 months' worth of Hands On information on them for free (or £3 if you didn't want the mag)?"
Do they? Oops, sorry. Although it may not be apparent, I work for *PCW* on a freelance basis, hence my mistake.
Smith goes on: "The *PCW* CD is a great reference as all 12 months are kept in a .pdf file called (in your case) database.pdf in a directory called \handson\pdf\. It is a most useful feature, as I can leave all my *PCW* mags at home and keep three CDs in the office (to cover 12 months each) and I have access to the last 36 months of all the Hands On articles."
Thanks for the useful tip. They're always welcome, so keep them coming.

PCW CONTACTS

Mark Whitehorn *welcomes readers' suggestions and feedback for the Databases column. He can be contacted via the PCW editorial office (address, p10) or email* **database@pcw.co.uk**

PLACES TO GO

▲Font from Guy's page

Here's news of an interesting web site from Guy van den Berg: "I've been running the webdb-l mailing list for about a year. It's a list dedicated to discussing web database integration methods and products. I also host a very large database links section and a nice database glossary. I thought some of the information on these pages might be useful for your readers.
"The pages are at http://black.hole-in-the.net/guy/ and http://black.hole-in-the.net/guy/links/."
Thanks for these, Guy. They're certainly well worth browsing — and what a great font.