

This free back issue of
THE QUICK ANSWER is provided courtesy of...

Inside
Sesame.com

The Monthly Newsletter for Sesame Database Manager
<http://www.insidesesame.com>
[Read a Free Issue of Inside Sesame!](#)

AND

L A N T I C A

S O F T W A R E L L C .

Makers of [Sesame Database Manager](#)
Compatible with Symantec Q&A™
<http://www.lantica.com>

Faster Retrieves for Variable Date Ranges

GORDON MEIGS

If you run reports, mass updates, merges, or other Retrieve Spec-sensitive operations with macros, try this innovative technique to set variable date ranges and speed the record selection part of the process.



I was scrolling through the Symantec Q&A for DOS forum on CompuServe the other day and spotted an interesting request for help. The Q&A 5.0 user was using Retrieve Spec expressions to select by date range the records to include in a series of reports, and he expressed his disappointment at the speed at which Q&A 5.0 retrieved the records. Here's a synopsis of the user's message, followed by my solution to the problem.

The problem

I've been using @XLookup expressions as my Retrieve Spec for several of my reports that require a date range, and I didn't have a performance problem until I upgraded to Q&A 5.0.

Using a retrieval expression, Q&A 4.0 scans my 7,000-record database in about 15 seconds. With Q&A 5.0 and the same 7,000-record database, it takes about 12 minutes.

My Retrieve Spec includes two @XLookup expressions to retrieve the starting and ending dates from an external database, like this:

```
>={@XLu("SETDATE", 9999, "Key", "Start date")}  
..<={@XLu("SETDATE", 9999, "Key", "End date")}
```

The XLookup database (SETDATE.DTF) contains one record with these three fields: Key, Start date, and End date. The Key field in the one record contains "9999."

When I use a standard Retrieve Spec parameter like the following, Q&A rapidly scans the database and selects the records that fall within the variable date range (in this case from June 1, 1996, through June 6, 1996):

```
>=6/1/96..<=6/6/96
```

However, I use a custom menu selection that prompts for the start and end date (which might be a few selected days, a week, or a month) and the series of reports are then run from a menu selection without requiring any further operator intervention.

I sure would appreciate a fix or workaround to this performance problem so that I can continue to run the reports with my automatic menu-driven method.

A change of technique

It's true that when performing some operations Q&A 5.0 isn't as fast as 4.0.

The Quick Answer™

The Independent Guide to Q&A Expertise

August 1996

Volume 7, Number 8

- 1 **Faster Retrieves for Variable Date Ranges**
Gordon Meigs
- 2 **Editorial**
Tom Marcellus
- 4 **More Windows-Like Gadgets**
Tom Marcellus
- 7 **All About Speedy Fields**
David Dvorin
- 9 **QuickTip: Improving Your Screens**
Tom Marcellus
- 10 **@Help**
Dave Reid
- 12 **Play That Tune!**
Tom Marcellus
- 14 **The Program Spec: Developer a Programming Style**
Jeff Nitka
- 16 **QuickTip: Useful Tricks with @Answer and @Msgbox**
David Dvorin
- 16 **QuickTip: Counting Your Blessings II**
Gordon Meigs

Continues on page 3

README.1ST

TOM MARCELLUS, EDITOR

VETERAN *Quick Answer* contributor **Gordon Meigs** likes a good challenge. Recently, he spotted a CompuServe Q&A forum message from a user vexed by how long it was taking to churn out a group of reports. The fellow was setting his variable date range parameters in an external database, then using a pair of @XLookups to it as his Retrieve Spec. Well, at two lookups per record and thousands of records to check for each report, it was bound to take Q&A some time. By saving the starting and ending dates to disk files, and turning the @XLookups into @Inserts, you'll discover how Gordon was able to cut the time by more than 75 percent.

Why should you index your database fields? How do you decide which ones to index? Can you safely undo an index? And when can too many indexed fields knock the wind out of Q&A? **David Dvorin** answers these questions and more as he tells you all about Speedy fields.

Too many keystrokes make me a Grumpy Old Man—I'm a sucker for gadgets that let me click my way through tasks. Lately, I've been quite the mad scientist, concocting more ways to make Q&A for DOS 5.0 easier to use by mimicking the mouse-activated controls you find in Windows programs. How about a spin control, a hideable check box, or a multi-button options box? Find out how these can simplify your work and make Q&A more fun than ever to use.

When I'm not grumpy, I like a little music. So I created a database that helps me compose, play back, edit, and save my favorite ditties to play via Q&A 5.0's @Play command. Why not have some fun around the office? I'll show you how you can *play that tune* whenever the spirit moves you or a specified database event occurs. **QA**

Spend a Day with The Quick Answer Masters You Read Every Month!

Now you can **meet** and **hear** these Q&A experts and **learn** their best-kept **secrets**, their **coolest tricks**, and their **power techniques**. **Discover** new ways to coax more out of Q&A and **extend** the functionality of your applications. **Watch** what they do with Q&A 5.0 and come away with a wealth of know-how. These speakers will be available all day to answer your questions and help you plan your next Q&A database. Then stay for the weekend and attend the National Q&A User Group's 6th Annual **BASH!**

David Dvorin Bill Halpern
Tom Marcellus Gordon Meigs

October 18!

10a.m–6p.m.
Friday, October 18, 1996
The Radisson Inn
Colorado Springs, Colorado

Sponsored by Marble Publications, Inc., publisher of *The Quick Answer* in association with the National Q&A User Group

Register Now! 800-780-5474

The Quick Answer™

The Independent Guide to Q&A™ Expertise

Editor Tom Marcellus
Publisher Michael Bell
Copy Editor Laurie Moloney
Production Editor Paul Gould

The Quick Answer (ISSN 1052-3820) is published monthly (12 times per year) by Marble Publications, Inc., 9717 Delamere Ct., Rockville, MD 20850.

Cost of domestic subscriptions: 12 issues, \$79; 24 issues, \$142. Outside the U.S.: 12 issues, \$99; 24 issues, \$172. Single copy price: \$10; outside the U.S., \$12.50. All funds must be in U.S. currency. Back issues are available upon request, for the same price as a single copy.

Periodicals postage paid at Rockville, MD.
POSTMASTER: Send address changes to
The Quick Answer, PO Box 9034, Gaithersburg,
MD 20898-9034.

Copyright © 1996 by Marble Publications, Inc. All rights reserved. No part of this periodical may be used or reproduced in any fashion whatsoever (except in the case of brief quotations embodied in critical articles and reviews) without the prior written consent of Marble Publications, Inc.

Address editorial correspondence, @HELP questions, or requests for special permission to: Marble Publications, Inc., *The Quick Answer*, PO Box 9034, Gaithersburg, MD 20898-9034. Phone 800-780-5474 or 301-424-1658. Fax 301-424-1658. CompuServe 73370,1575. Prodigy NEPY97A.

For Q&A technical support, call Symantec: 503-465-8600.

Q&A is a trademark of Symantec Corp. Other brand and product names are trademarks or registered trademarks of their respective holders.

This publication is intended as a general guide. It covers a highly technical and complex subject and should not be used for making decisions concerning specific products or applications. This publication is sold as is, without warranty of any kind, either express or implied, respecting the contents of this publication, including but not limited to implied warranties for the publication, quality, performance, merchantability, or fitness for any particular purpose. Marble Publications, Inc., shall not be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this publication. Articles published in *The Quick Answer* do not necessarily reflect the viewpoint of Marble Publications, Inc.

Faster Retrieves ...

Continued from page 1

This might be due to the additional overhead created by the many new features. On the other hand, I've often found a way to use a new Q&A 5.0 feature in situations like this to regain most, if not all, the lost speed.

For the case at hand, the following modifications to the macro-driven procedure recover much of the lost speed, and still allow any number of Q&A reports or other procedures to make use of the general Retrieve Spec parameters you're using in the one-record SETDATE database. Here are the modifications I recommend:

1. Add two text fields named Start and End to the SETDATE database.
2. Program the database to copy the Start Date (#20) and End Date (#30) values as text values to the corresponding Start and End fields (#40 and #50). Here's a sample statement set to execute on leaving the End Date field (#30)

```
>#30: #40 = @Str(#20); #50 = @Str(#30)
```

This way, if the Start Date field contains June 1, 1996, and the End Date field contains June 6, 1996, the program will fill the corresponding Start and End fields with the following values:

```
Start: 1996/06/01
End: 1996/06/06
```

3. When you press F6 in a text field, Q&A opens the Expanded Field editor, allowing you (or a macro) to copy the field's contents to a disk file. Accordingly, modify the macro that sets the start and end dates so that it moves to the two new text fields in SETDATE and "presses" the following keys to save the retrieval parameters to disk files named C:\TEMP\START.TXT and C:\TEMP\END.TXT, respectively:

```
F6, Ctrl-F5, End, F10, Shift-F4, C:\TEMP\START.TXT,
Enter, Y, F6
Tab (to move to the End field)
F6, Ctrl-F5, End, F10, Shift-F4, C:\TEMP\END.TXT,
Enter, Y, F6
```

The "Y" in each case tells Q&A to overwrite the existing file of the same name. (Both files must exist when recording and running the macro.)

This part of your macro should now look like this:

```
<f6><ctrlf5><end><f10><capsf4>
c:\temp\start.txt<enter>y<f6><tab>
<f6><ctrlf5><end><f10><capsf4>
c:\temp\end.txt<enter>y<f6>
```

Specifying a local path to START.TXT and END.TXT (in this example, C:\TEMP\) is advisable if you're running Q&A on a network. It speeds writing the file to disk and retrieving it later. It also lets each user specify a custom date range. On a single-user system, the path isn't needed—Q&A will save the text files to your default documents subdirectory.

4. Change your reports' Retrieve Spec parameters from the two @XLookup expressions to this expression in the date field:

```
>={@Insert("c:\temp\start.txt")}..
<={@Insert("c:\temp\end.txt")}
```

Here, Q&A 5.0's new @Insert command replaces the slower multiple @XLookups, retrieving and inserting the disk files with the starting and ending dates into the report's Retrieve Spec. You can use this same date range retrieval technique for Search/Updates, Mass Updates, Copy operations, merge printing—for virtually any operation where you need to select records based on a variable date range.

I tested the two techniques on a database of more than 4,000 records. Using the two original @XLookup expressions, the scan took more than four times as long as the @Insert approach. I also tested the @Insert approach on a Windows 95 peer-to-peer network with the search database on another drive. The C:\TEMP local path for the text files made a crucial difference: the scan took 32 seconds with the date range files stored in the local c:\temp directory versus more than five minutes if they were stored on the non-local drive that contained the database. **QA**

Gordon Meigs is vice president and general manager of Professional Computer Technology Associates of Newtown, Pennsylvania. He teaches courses and does corporate training on Q&A, and has been designing and installing advanced Q&A business applications for more than nine years. 215-598-8440, CompuServe 71023,356.

[When you need a macro to select the same records for several consecutive operations, don't rule out Q&A 5.0's Clipboard. In this case, you could use the Clipboard to pass the date range to the Retrieve Spec, and use a literal search value instead of a slower Retrieve Spec expression.]

To do this, you could add a text-formatted DateRange field to SETDATE.DTF, and program it with an on-field-entry statement that concatenates the StartDate and EndDate values, like this:

```
< DateRange = ">=" + @Str(StartDate) + "...<="
+ @Str(EndDate)
```

Continues on page 11

More Windows-Like Gadgets

TOM MARCELLUS

Your editor has been working overtime, dreaming up more ways to make Q&A for DOS 5.0 easier to use by mimicking the mouse-activated controls found in Windows programs.



WHEN built-in mouse support arrived with Q&A 4.0, I thought of the Intelligent Assistant—another gimmick only a gullible novice would use. It took Dave Dvorin to convince me that the rat had a place in Q&A for DOS. Today, with Q&A 5.0, I'd be lost without it.

In the January 1996 issue ("Streamline Tasks with Custom Windows-like Gadgets"), I showed you how to add click-on buttons, check boxes, dialog boxes, and the like to your Q&A 5.0 forms to make them mouse-able—that is, less keystroke-intensive. Since then, I've developed three other Windows-like gadgets—a spin control, a hideable check box, and an options box. I'll show you where these can come in handy, and how to add them to your applications.

The spin control

The spin control (also called an UpDown control) is a gadget that increments or decrements a value. You click on a graphical up arrow to increase the value and a down arrow to decrease it.

You could use a spin control with a number field, to set the number of copies of an invoice to print. The default might be 2, but there might be times when you need to print three, four, or just one copy of an invoice. With a spin control, you wouldn't have to move to the field and type a number, you could simply click on one of the arrows to set it.

In my QuickClick Calendar database, I use a spin control to change the calendar year. **Figure 1** (top) shows the calendar year spin control as it looks on the form. **Figure 1** (bottom) shows it at the database design screen.

It takes three labelless fields to make this spin control—one read-only field to display the year, and one for each set of arrows. The control's elements are set with on-record-entry commands like these:

```
YearDn = "▶▶";
YearUp = "◀◀";
```

```
Year = @Left(@Year(@Date), 1) + " "
      + @Mid(@Year(@Date), 2,1) + " "
      + @Mid(@Year(@Date), 3,1) + " "
      + @Right(@Year(@Date), 1);
```

Ordinarily, a Year field would be numeric. For appearance, though, I wanted a space between each digit, so I made it a text field and used the previous string statement to add the spaces.

The Year, YearDn, and YearUp fields, respectively, contain the following programs:

```
< Year = @Left(Year,1) + " " + @Mid(Year, 2,1) + " "
      + @Mid(Year, 3,1) + " " + @Right(Year,1); Chome
< Year = @Num(Year) - 1; Goto Year
< Year = @Num(Year) + 1; Goto Year
```

If Year were a regular numeric field, The Year program and @Num functions wouldn't be needed, and Chomes would replace the Goto Year commands.

Other spin control candidates

You can set a spin control to increase and decrease a value by any range. You could "spin" a money value by cents or whole dollars, or a discount value by a factor of 5 so any discount would be a multiple of 5 percent.

You can also set upper and lower bounds, and even rotate the value if it exceeds either one. For example, you might want to restrict a value to 1 through 9.

An attempt to make it less than 1 would rotate it to 9, and trying to set it higher than 9 would rotate it to 1. The following sample



Figure 1. (Top): With this spin control, you click on the left or right arrow to decrement or increment the calendar year. (Bottom): Adding the spin control during database design.

programs for the CopiesUp and CopiesDn fields would do this for a numeric Copies field:

```
< Copies = Copies + 1;
  If Copies > 9 Then Copies = 1; Chome

< Copies = Copies - 1;
  If Copies < 1 then Copies = 9; Chome
```

Spinning dates

Another handy use for the spin control is to increment or decrement a date field by a day, week, or month. You might want to set variable deadline or follow-up dates in your records, but instead of typing them, you'd rather just click your mouse.

Figure 2 (Top) shows a date spin control as it might appear on a form. Figure 2 (Bottom) shows how it's designed.

The Figure 2 spin control includes an optional feature. It displays—on the message line—the current date, the weekday of the date in the date display field (DateBox), and the number of days between the current date and the displayed date.

In the Figure 2 record, July 15 is the current date, and DateBox has been "spun" to October 18, 1996. (Hey, that's the date of the Q&A Master Seminar in Colorado Springs!) So the message line reads *July 15, *** Friday ****, and 95 days between July 15 and October 18. The programs that take care of this are in the DateBox, DateUp, and DownDn fields, respectively:

```
< @Msg("Today is " + @Month$(@Date) + " "
+ @Str(@Dom(@Date))
+ @Text(30, " ") + " *** " + @Dow$(DateBox) + " ***"
+ @Text(10, " ") + @Str(DateBox - @Date)); Chome

< DateBox = DateBox + 1; Goto DateBox

< DateBox = DateBox - 1; Goto DateBox
```

The elements of the date spin control are set by this on-record-entry program:

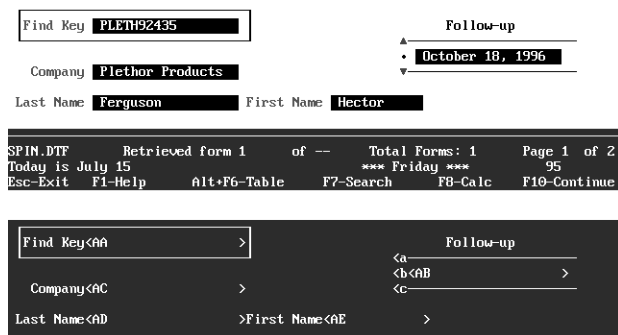


Figure 2. (Top) Portions of a form with a date spin control in use. (Bottom) The same control as it looks on the design screen.

```
< DateUp = "▲";
DateDn = "▼";
Reset = "•";
If DateBox = "" then DateBox = @Date
```

The bullet character you see between the DateUp and DateDn arrows is a Reset field—you click on it to reset the displayed date to the current date using this program:

```
< DateBox = @Date; Chome
```

The hideable check box

A check box is a gadget you click on to toggle something on and off. I use two of them in my Tune-o-matic database—one to toggle the tone recorder on and off, and another to toggle tone durations on and off (see Figure 3).

When I want to record some tones for use with @Play commands, I click on the Record check box. If I want to add other than the default duration to the tones, I click on the Durations check box, too. Then, when I select a tone, the program prompts me to specify its duration, which it records with the tone. If I just want to play a few notes without recording them, I uncheck the Record check box.

Since durations can't be recorded if Record is off, unchecking Record also unchecks Durations. You can use these gadgets in tandem or in a series this way.

Figure 3 (Top) shows the Record and Durations check boxes as they appear on a form. (In this case, Record is checked, but Durations isn't.) Figure 3 (Bottom) shows them at the design screen.

The Record check box in Figure 3 is a nine-character labelless field named Record. Likewise, the Duration check box is a 12-character labelless field named Durations. Each is filled with the left bracket, a space, the right bracket, then the field name. This way, I can click anywhere on the field to toggle them, and temporarily hide them by re-coloring them with @Color, or by clearing them, then putting them back when I want them available.

By adding a command like the following to the beginning of a check box field's program, simply clearing the field (with a Clear command elsewhere) will prevent

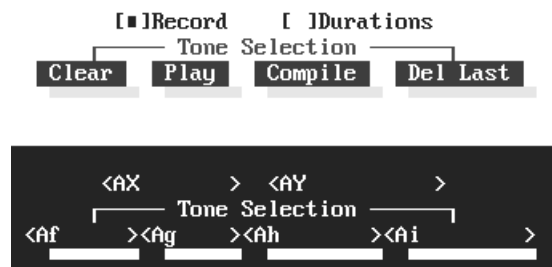


Figure 3. (Top) Two check boxes and several buttons on a form. (Bottom) Adding the check boxes at the design screen.

the rest of check box's program from executing:

```
< If Record = "" Then Chome;
```

I wanted the Record check box to be checked by default, but not the Durations check box, so I used the following on-record-entry commands:

```
Record = "[■]Record";
Durations = "[ ]Durations"
```

The programs for the Record and Durations fields, respectively, look like this:

```
< If @Instr(Record, "■") > 0 Then {
  Record = "[ ]Record"; Durations = "[ ]Durations" ; }
Else Record = "[■]Record"; Chome

< If @Instr(Durations, "■") > 0 Then
{ Durations = "[ ]Durations"; Chome };
If @Instr(Durations, "■") = 0 Then
Durations = "[■]Durations";
If @Instr(Record, "■") = 0
and @Instr(Durations, "■") > 0
Then { @Msg("Record is off. Can't record durations");
Durations = "[ ]Durations" }; Chome
```

The "■" character (ASCII 254) serves as the check mark, and the toggle action is based on its presence or absence in the field. Programming elsewhere in the database reads the status of the two check box fields and acts accordingly.

Option controls

You might need a way to prompt the user to choose one option among several. The @Askuser command limits you to a Yes or No response, so it won't do. You can't make a Userselect list look like anything but a raw list, control the order in which its selections appear (unless you number them), or use it outside a database. So it might not do, either.

But you can display a custom menu wherever you can run a macro—including via an @Macro command. The selections on it can appear in any order you like, and you can make them look and work like graphical buttons in a real options box. Figure 4 shows such an options box displayed in a contact database, though you could display this gadget anywhere in Q&A.

You use the Application Menu Options screen (Shift-F2 / Create Menu / New Menu) to create the three-button options box shown in Figure 4. First, create the button borders on the Menu Items lines, using ASCII 196 for the horizontal borders, ASCII 179 for the vertical borders, and ASCII 191, 192, 217, and 218 for the corners. When you've drawn the borders, type the button text inside them on lines 2, 5, and 8 (see Figure 5).

The first button occupies the first three Menu Item lines, so you add the associated macro name (in this case, *Merge Print Letter*) to the first three adjacent Macro Name lines. For the second button, you add its macro name to the next three lines, and do likewise for the third button. This way, you can "push" a button by clicking anywhere on its label or border, or by highlighting any part of it and pressing Enter.

You'll need a macro that does nothing but display the options box. In this case, it's named *Print Options* to tie it to the *Choose Print Option* menu name and title.

If you use an @Macro command to display the options box, you can make any button the default by invoking another macro that "presses" the appropriate number of Down arrows. In the Figure 4 options box, the *Print Whole Record* button is made the default by invoking a macro that "presses" Down arrow seven times.

Following is a program that displays the options box over a database record and highlights its third button, making it the default:

Continues on page 11

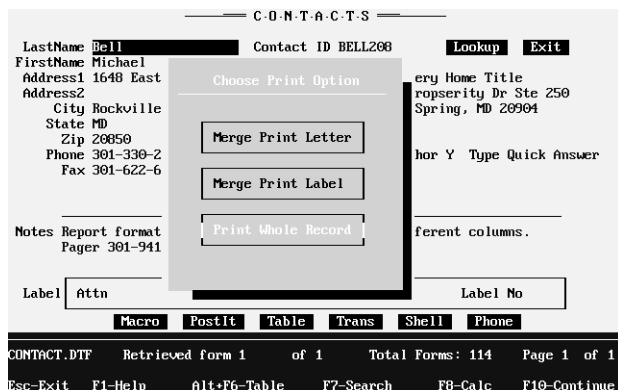


Figure 4. A custom menu designed to look and work like an options box.

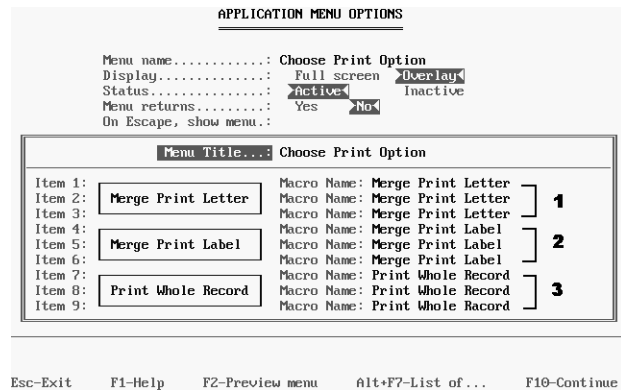


Figure 5. Creating a custom menu that works like a three-button options box you can display anywhere in Q&A.

All About Speedy Fields

DAVID DVORIN

Indexing the right database fields can put a tiger in Q&A's tank. But index too liberally and you can actually hamper performance.



WHEN you fill out a Retrieve Spec and press F10, Q&A finds the matching records in one of two ways. By default, Q&A checks every record in the database, culling those that meet your retrieval parameters. Or, Q&A finds the matching records immediately. When does Q&A use the faster second method, and how can you make sure it does? The answer is indexing.

What's an index?

Q&A stores records in roughly the same order as you add them to the database, which in most cases is no useful order at all. When you search on a field that isn't indexed, Q&A has to examine record after record until it finds the one that matches. It might be the last one you entered.

On the other hand, if you index the field—as a publisher might index topics in a book—Q&A can find the matching records (occurrences of that topic) immediately.

When you index a field, Q&A duplicates its entries in the .IDX file associated with the database. If you then search on that field, Q&A consults the index, finds out which record(s) contain that entry, and fetches them immediately—it doesn't have to scan the entire database.

A database really consists of two files. The .DTF file contains the records, fields, and their data, along with the various specs, reports, and so forth, and the .IDX file contains the indexes.

In Q&A for DOS, you index a

field by making it *Speedy*. In Q&A for Windows, it's called *indexing the field*. Same difference. (See the sidebar, "How to Index Fields.")

The importance of indexed fields

In addition to Search/Update tasks, indexed fields can speed up most any database operation that involves searching, retrieving, and optionally sorting records, including the following:

- Mass Update
- Remove selected records
- Copy selected records
- Export
- Reports
- Merge documents

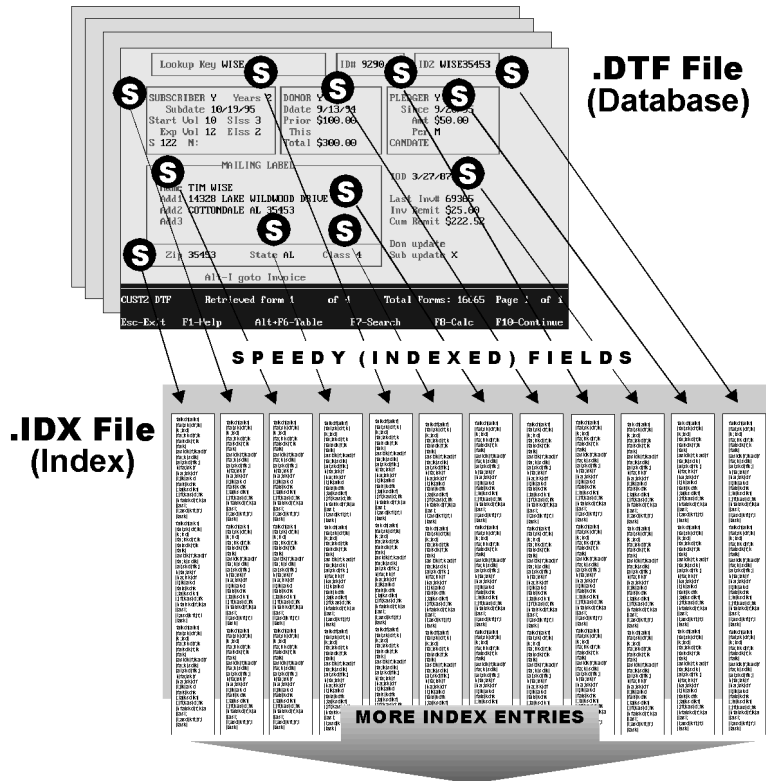


Figure 1. How an .IDX file gets huge and makes Q&A sluggish. The more Speedy fields in the database, the more indexes Q&A must maintain. Here, Q&A updates 12 indexes (in the one .IDX file) as a new record is saved, and the result is an appreciable delay before the next record appears.

Of course, to reap the benefits of indexed fields, you'll have to retrieve and sort on them. Even though several fields in the database might be indexed, they can't help when you retrieve or sort on other non-indexed fields.

Which fields should you make Speedy?

When it comes to database performance, I'm often asked, "Why not just index every field?" Although Q&A allows up to 115 indexed fields per database, it doesn't mean you should go that far. Too many indexed fields can actually hamper database performance. (Symantec recommends limiting indexed fields to two or three per database.)

In general, each value in an indexed field requires an entry in the index. So, when you save a record, not only does Q&A save the data to the .DTF file, it must also update the index or indexes (see **Figure 1**).

The more indexed fields in the database, the more time it takes Q&A to update the associated indexes. If the database contains a vast number of records and lots of fields, many of which are indexed, Q&A can feel sluggish. You might notice a lag between the time you press F10 to save a record and when the next record appears, or a slowdown in search and sorting speed.

What's more, too many indexed fields can make the

.IDX file larger than the .DTF file! So, if you want Q&A to save new and updated records faster, keep your indexed fields to a minimum.

Making the decision

In deciding which fields to index, focus on those you normally use to select and sort records for various tasks. If you frequently retrieve records on the same field or combination of fields, these are the ones to index. (I've dramatically increased macro performance by indexing fields included in the saved Retrieve Specs used by the macro.)

This can be easier said than done, though. The problem is trying to figure out if you use a field often enough to merit indexing it. Sometimes, the only way to find out is by trial and error. Many factors are involved, such as the number of fields and records in the database, and data entry requirements. The best way to decide which fields to index is to look over your Retrieve Specs for the database and ask yourself, "Which fields do I normally use?"

A special case is Q&A's *Remove duplicate records* command. If you use this feature frequently, consider indexing those fields you specify at the Duplicate Spec. If you don't, Q&A will create a temporary index each time you run the command, adding considerably to the time it takes to perform the task.

Mandatory Speedy fields

The only fields you *must* index are those that serve as external key fields for XLookups, Posting, and similar commands that manipulate an external database. For example, if an XLookup command in an INVOICE database retrieves information from the CUSTOMER database record with the matching Customer ID, then the Customer ID field in CUSTOMER (but not in INVOICE) must be indexed. In this case, the Customer ID field in CUSTOMER should also be unique; otherwise, Q&A might retrieve information from the wrong record.

If you index no other fields than these, your database will still operate properly, but it might not run at peak efficiency. In deciding which fields to index, .IDX file size and the time it takes Q&A to update indexes during data entry are the relevant issues.

How to Index Fields

To index a field in Q&A for DOS, you display the Speed-up Spec (File / Design a file / Customize / Speed up searches), and type an *S*, *SU*, or *SE* in each field you want to index.

An *S* (for Speedy) tells Q&A to index the field's values. You can follow it with a *U* to tell Q&A that entries in this field should be unique—that is, the entry should occur in only one record in the database. Examples of fields you might want to make *SU* are company names and social security numbers—unique values you're likely to use to search the database.

Or, you can make a field *SE* (Speedy Exists) to tell Q&A that anything entered in the field should already be in the database. Examples of *SE* fields might be states or departments. *S* and *SU* indexes are the most commonly used.

To index a field in Q&A for Windows, pull down the Select menu, choose Database structure, and double-click on the Index column to display the Edit Index dialog box. Select the field to index, and click on the type of index you want Q&A to create. (*All Values* is the same as *S*; *Unique Values* is equivalent to *SU*; and *Existing Values* is the same as *SE*.)

If *SU* (unique) or *SE* (exists) rules are violated during data entry, Q&A displays a warning message as you leave the field, but won't stop you from leaving the entry as is.

Making a field *SE* or *SU* has no effect on field values already in the database (other than to add them to the index). So, if the same value is repeated in several records, and you change the field to *SU*, Q&A won't notify you that the values already in that field aren't unique.

Playing with indexes

Any mistake you might make indexing a field can always be undone with no harm to your database, so it's all right to experiment. You can index a field, perform some tests, and later remove the index by deleting its indexing code. It takes no time to delete an index, but to create one Q&A must examine every record in the database. With a large database (or on a network or a slow computer), expect the process to take some time.

Recovering a database

Q&A's *Recover Database* utility can work miracles with a damaged database or index file. (In Q&A for DOS, select File / Utilities / Recover database. In Q&A for Windows, close any open database, pull down the File menu, and select Recover.) Recover Database is typically used to repair a database damaged by a power glitch, or to recompact one whose records are scattered all over the hard disk due to frequent adds and deletes. But Recover Database also reindexes a database using its current index specifications. It's good preventive maintenance for a database—particularly one where you frequently add, delete, and update records—to periodically back up it up (both the .DTF and .IDX file) and run Recover on it, even if aren't experiencing any problems with it. **QA**

David Dvorin owns Phoenix Solutions of Hillsborough, New Jersey, which specializes in tailoring off-the-shelf software for a broad range of business needs. 908-281-6272, dvorin@bms.com.

When Your .IDX File Goes South

Without an associated .IDX file, Q&A won't let you use the database. I've had occasions when someone forgot to give me the .IDX file with a database, or an existing .IDX file somehow got corrupted. Fortunately, there's a way to re-create a bad or missing one.

Begin by copying another database's .IDX file to the name of the database with a damaged or missing .IDX file. For example, if you have a CUSTOMER database with a bad or missing index file, you can copy the .IDX file from any other database to the name CUSTOMER.IDX.

When you've done that, go immediately to the Speed-up Spec for the problem database, clear it by pressing F3, then resave it by pressing F10. (In Q&A for Windows, use the Edit Index dialog box to delete any indexes, then click OK.)

Return to the Speed-up Spec. This time, fill it out the way you want it and resave it by pressing F10. (In Q&A for Windows, use the Edit Index dialog box to reindex the pertinent fields, then click OK.)

Quick Tip

Improving Your Screens



In Windows databases, the information blank portions of fields are usually visible even when empty, making them easier to

spot and click on when updating information in a record. You can mimic this feature in a Q&A 5.0 database by setting the background color of your fields to one that contrasts with the form's background, and setting the field text color to one that contrasts with the field background color.

You'll run into a problem, though, if you place information blanks for two fields on adjacent rows—the background colors of the two will join, making them look like a single two-line field. To get around this, you have to skip a line between each field.

Figure 1 shows an address database where the information blanks are visible even when empty. The form's background is set to white (a light gray), and the information blanks are set to bright white text on a black background. Using the "<" character instead of a colon to mark the beginning of the field imparts an even smoother look. **QA**

Find Key PLETH92435 Follow-up August 31, 1996
Company Plethor Products
Last Name Ferguson First Name Hector
Address1 143 Warwick Ave
Address2
City Tustin State CA Zip 92435
Phone 714-722-9896 Fax
Last Ord Amt
Notes
SPIN.DTF Retrieved form 1 of — Total Forms: 1 Page 1 of 1
Esc-Exit F1-Help Alt+F6-Table F7-Search F8-Calc F10-Continue

Figure 1. Coloring your field information blanks and aligning them makes the form highly readable and a snap to navigate with a mouse.

Tom Marcellus



EDITED BY DAVE REID

Fix Label Creep on A4 Paper

I've been trying without success to print labels on A4 (European letter size) paper using my HP DeskJet 500 and Q&A 4.0's Mailing Labels facility. I've tried different Label Height and Lines per Label Sheet settings, but I can't prevent the text on the labels from creeping up. I believe the correct settings should be 1 5/8" for Label Height, and 69 lines per Label Sheet. Is there another setting I'm missing?

J.B. Harper, Blackburn, United Kingdom

The two settings you mention are the most critical when it comes to label creep. When labels creep up, it usually means the Label Height needs to be increased. In your case, I suggest specifying the setting in whole lines rather than inches. (Most printers print six lines to the inch.) Doing the math yields a setting of 9.78, but you'll have to enter a round number, so use 10 as the Label Height. You have seven rows of labels per sheet. If each label is 10 lines high, then you should change the Lines per Label Sheet to 70.

From the looks of your printout, Q&A might not be printing the full length of the A4 sheet. If the previous settings don't work, try a Lines per Label Sheet setting of 65, and change Blank Lines at Bottom to 5. This will leave 60 lines of printable area, and Q&A should print six 10-line labels down the page. The downside is that you'll waste one row of labels on each sheet.

Getting Check Amount Subtotals

I use a Q&A database to track payments received. I enter each payee, check amount, and so forth, print a report sorted by check amount, and attach it to my bank deposit slip. Now, my bank wants a column that shows a count of the number of checks in each check amount group, and I'm stumped.

Janet Goodman, Erie, Pennsylvania

You can make a few modifications to your existing report to get the additional information you need. Go to the Column/Sort Spec for the existing report, press F8, and

create the following three derived columns:

Heading: Countable column
Formula: "x"
Column Spec: 5,I

Heading: Count by!Amount
Formula: @Text (#50 <> #4, @Count (#5, #4))
Column Spec: 6,F(N)

Heading: Delayed copy of check amount
Formula: #4
Column Spec: 50,I

This technique copies the sorted column value to Column 50, then compares the two columns as Q&A compiles the report. It forces the count to print just once for each check amount group. The third column (Column 5) is needed because your sorted column is the last one on the old report, and column 6 needs to count a column to the right of the sorted column.

Print Multiple Fields In One Column

I have a number of reports that use a spacing technique to combine several fields into a single column. Though I like the output, I run into trouble when I print in a proportional font. The second line of data sometimes begins printing at the end of the first line, and so on down the column. Is there another way to get the output I need?

David J. Mount, New York, New York

The trick you refer to is a useful method for combining several fields into one column, and controlling where the lines within the column wrap. I've never found a way to make the technique totally foolproof. As you know, it involves combining a column made up of spaces with the separate pieces of data. The Column/Sort Spec and derived columns for such a report typically look like this:

First Name: 1,I
Last Name: 2,I
Address: 3,I
City: 4,I
State: 5,I
Zip: 6,I

Heading: Spaces
Formula: @Text(30, " ")
Column Spec: 9,I

Heading: Name and Address
Formula: #1+" "+#2+#9+#3+#9+#4+", "+#5+" "+#6
Column Spec: 10, H(30)

The object is to yield a report where each entry in the Name and Address column looks like this:

Jeremiah Johnson
PO Box 12
Heart Butte, MT 37833

The key to the technique is Column 9's 30 spaces. Since Column 10 is limited to 30 spaces, it wraps to accommodate all the data. Q&A removes any spaces at the beginning of a line, so the output looks clean. To modify the technique for use with proportional fonts, try increasing the number of spaces in Column 9. The maximum column width is 80, so change Column 9's Formula to this:

Formula: @Text(80, " ")

The advantage to using 80 spaces instead of 30 is that you can print column 10 in a proportional font, and it should

still look good. (Q&A won't insert any blank lines between the lines.) If you make column 10 wider than about 30, though, then any line with a small amount of data is likely to cause the problem you've already seen.

QA

Dave Reid is the coauthor of *The Q&A 4.0 Wiley Command Reference*, published by John Wiley and Sons, and works as an independent Q&A consultant. PO Box 12083, Eugene, OR 97440.

Have a nagging question? Send it to [@Help, The Quick Answer, Marble Publications, Inc., PO Box 9034, Gaithersburg, MD 20898-9034](#) or fax to 301-424-1658. When writing, please include your name, address, and phone number, along with your Q&A version number (and whether DOS or Windows) and a detailed description of the problem. We will publish those questions we feel are of general reader interest; individual responses are not possible.

Faster Retrieves ...

Continued from page 3

With a StartDate of June 1, 1996, and an EndDate of June 6, 1996, this would produce the following literal DateRange value:

```
>=1996/06/01..<=1996/06/06
```

To run the reports (or other operation), have the macro open SETDATE.DTF, prompt for the StartDate and EndDate,

then tab to the DateRange field, trigger the program, and copy the DateRange value to the Clipboard (F11). Have the macro continue by pasting (F12) the parameter into the Date field of the first report's Retrieve Spec and running the report. Then have the macro run the subsequent reports by pasting the same Clipboard value into their respective Retrieve Specs. Assuming the Date field is Speedy and the database is large, the additional time the macro takes to make this temporary change to each report should be more than offset by the speed of retrieving the records using a literal search parameter.

Be aware, though, that if you add reports to the series, or move any of the fields, you'll have to edit the macro to account for these changes.—Ed.]

More Gadgets ...

Continued from page 6

```
< @Macro("Down7"); @Macro("Print Options")
```

You can display this kind of options box anywhere in Q&A. For example, you might run a macro that mass updates a batch of records then prints a mail merge. As

it's final step, you could have the macro display an options box giving you the choice to print an update report, view the updated records in table view, or return to the original database to continue data entry. **QA**

Tom Marcellus is editor of *The Quick Answer*, and author of *PC World Q&A Bible* (IDG Books). His QuickClick Calendar Plus—a time- and activity-tracking database for Q&A 5.0—is available from Marble Publications, publisher of *The Quick Answer*.

Play That Tune

TOM MARCELLUS

A respected efficiency expert (whose name escapes us just now) once remarked, "If you can't have a little music at work, why go?" Taking this "sound" advice to heart, here's how to create a library of catchy tunes you can play from any database.



If you've been around Q&A 5.0 for a while, chances are you've come across the @Play command. (See the sidebar, "@Play Basics.") Though you can use @Play to make your PC beep or click to alert users to an error, a completed calculation, or some other event, why settle for noises when you can make your PC play real music? By creating a *sequence* of @Play commands—that is, tones and durations—you can make Q&A play just about any tune that strikes your fancy. I'll show you how.

Creating a musical instrument and recorder

When I set out to compose a few tunes in Q&A 5.0, I was tickled by the results, but not the technique. Working out which frequencies matched the various notes of the musical scale was only half the battle. What gave new meaning to the word *tedious* was finding the appropriate duration for each tone, playing back the sequence of tones time and again to "polish" the tune, then packaging the thing in a series of @Play commands to make a program. To play just the first four notes of the old *Dragnet* theme, for example, you need the following sequence of commands:

```
@Play("Sound", "220,600"); @Play("Sound", "247, 200");
@Play("Sound", "262,400"); @Play("Sound", "220, 400")
```

Pretty cryptic—and lots of keystrokes to play just four notes.

Once I'd managed to match the frequencies to notes I could recognize and select (see **Figure 1**), I needed a musical instrument to play the tones, and a recording and playback device to help me test, edit, and compile my tunes. This way, I could concentrate on the tune and avoid programming commands and syntax.

Many who attended the Q&A Master Seminar last

year in Washington, D.C. went home with a copy of my TUNES.DTF database on their free disk. TUNES.DTF plays a few canned tunes, and lets you click on the tone frequencies to hear what they sound like, but that's all.

I've since overhauled TUNES.DTF to incorporate all the tools I need to quickly compose, test, edit, compile, and save tunes. Figure 1 shows my new TUNES2.DTF database, The Amazing Tune-o-matic.

There's far too much to TUNES2.DTF to cover in the space of an article, and you wouldn't want to create it even if I told you how—it would take hours. Instead, I'll show you a TUNES2.DTF feature that lets you store a library of tunes you can XLookup and play in any database.

Adding a playback device to your database

Imagine you run an upbeat sales office, and you want everyone to know when a big, medium, or dinky order is entered by having the computer play a catchy tune.

You want to play *Ode to Joy* if the order is \$1000 or more, *CHARGE!* if the order is \$250 or more, and a funeral dirge if the order is less than \$250.

Start by adding the following entries to the ORDERS database lookup table:

Key	Column 1
01000	Ode to Joy
00250	CHARGE!
00001	Dirge

Next, add a small text field named PlayString to the ORDERS database. You can place it anywhere, hide it with color if you like, and make it read-only. Add this program to it:

```
< If @Instr(PlayString, " ") > 0
and @Instr(PlayString, ",") > 0
Then { @Play("Sound", @Left(PlayString,
@Instr(PlayString, " ")));
PlayString = @Del(PlayString, 1,
@Instr(PlayString, " "));
Goto PlayString }

Else { @Play("Sound", PlayString);
Clear(PlayString); Chome }
```

The PlayString program performs a loop, playing the first tone/duration, deleting it, then repeating the process until the final tone/duration is played and the field is cleared.

Before the PlayString routine can be of any use, though, you'll need a program that retrieves the appropriate tune (tone/duration sequence) based on the order amount. Here's a programming statement for the Order Amount field that does the trick:

```
> If @Add Then { PlayString = @Xlookup("MYTUNES",
  @LuR(@Rt("00000" + @Str(@Int(Order Amount)), 5) ,1),
  "Name", "Play"); Goto PlayString }
```

@Play Basics

@Play requires one or two parameters, depending on what you want to play. To play a simple beep or click on your computer's internal speaker, use @Play with a single parameter, like this:

```
If @Error Then @Play("Beep")
If Past Due > 0 Then @Play("Click")
```

To make Q&A play a sound other than beep or click, you make *Sound* @Play's first parameter, then add a second parameter for the frequency and duration. Frequency is the tone or note expressed in cycles per second (Hz), and duration is the length of time to play the tone.

Unless you're a dog, tones in the 100 to 1000 Hz range sound pretty good on a PC's internal speaker (better than a kazoo, at least). This gives you a three-octave scale to play with. (See [Figure 1](#).)

You express each tone's duration in 1000ths of a second—a duration of 500, for example, plays the tone for half a second. The following command plays C# on the C4 scale for a quarter of a second:

```
@Play("Sound", "277,250")
```

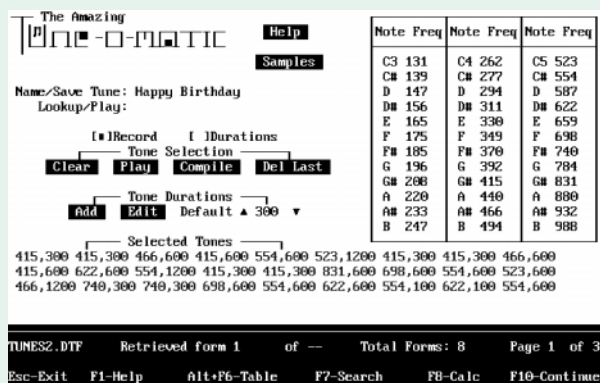


Figure 1. The range of tones from 131 Hz to 988 Hz sound pretty good on your average PC's speaker.

The program creates a five-digit string from the Order Amount. This way, the @LookupR range command can select the appropriate tune from the lookup table. The selected tune serves as the external key field parameter for the @XLookup command, which retrieves the tone/duration sequence from MYTUNES.DTF and places it in the PlayString field.

With these two programs and the supplementary tunes database I'll describe next, Q&A will play the appropriate tune when the cursor exits the Order Amount field in a new order.

Creating a tune storage database

If you don't have TUNES2.DTF, you'll need to create a database (MYTUNES.DTF in this example) to store your tunes, and manually enter the tone/duration sequences for each tune you want available to play. Include in MYTUNES a *Speedy Name* field for the name of the tune, and a *Play* field for the tone/duration sequence that makes up the tune.

For example, The *Play* field of an *Ode to Joy* record in MYTUNES.DTF would contain the following sequence of tones and durations in the *Play* field:

```
440,400 466,200 523,200 523,200 466,200 440,200
392,200 349,400 392,200 440,200 ,100 440,400 392,400
,300 440,400 466,200 523,200 523,200 466,200 440,200
392,200 349,200 ,100 349,200 392,200 440,200 392,400
349,400
```

Be sure there's a comma between each tone and its duration, and a space between each tone/duration in the sequence. Otherwise, the tune won't sound right. You can create a kind of staccato effect by omitting the frequency, and typing just a comma followed by a duration of 100.

Musical options

Though you could optionally create *Ode to Joy* as a full-blown @Play program, and add it to the databases in which you want to play it, doing so would require an additional 300 or so characters for all the syntax.

By storing just the frequencies and durations of your tunes in a supplementary database like MYTUNES, and looking them up from the database that plays them, you gain flexibility and economize on programming, particularly if you want the option to play several tunes in the same database or the tunes are lengthy. All you need in the playback database are the tune retrieval and playback programs shown earlier.

You can optionally equip your MYTUNES database to play whatever tune is in the current record by adding PlayString and PlayIt fields to it. Program PlayString as shown earlier, and program PlayIt like this:

```
> If PlayIt = "Y" Then { Clear(PlayIt);
  PlayString = Play; Goto PlayString }
```

Continues on page 15

Develop a Programming Style

JEFF NITKA

Anyone formally trained in computer programming knows the value of consistent programming style. Good style makes your code understandable and easier to modify and debug. Jeff Nitka discusses two aspects of good programming style.

WHAT makes a programming style “good”? Professional programmers usually point to two aspects: comments and readability. Let’s discuss comments first.

Comment your code

With Q&A 5.0, you use the REM command to add comments to your programs. Q&A 4.0 users can achieve the same effect using @Msg commands, as in the following examples:

```
REM "This section performs a loop";
@Msg("This section performs a loop"); @Msg("");
```

The second @Msg("") statement in the second example prevents the message from displaying during program execution.

Q&A saves either type of comment as executable code that has no effect—a drawback if your available space for executable code is nearing the limit. In such cases, you should comment your code selectively and concisely.

While commenting is straightforward, readability rules aren’t as clearly defined. It can be difficult to know if your style is readable by others—or even yourself, should you forget the purpose of a particular code fragment. Following are some general guidelines.

If-Then-Else statements

The way you structure your If-Then or If-Then-Else statements can be crucial to recognizing the relationship between the *condition* and the *action*. Here’s an example where everything is mashed together:

```
If Jeffs Age > 16 Then If Toms Age > 16
Then Goto See Rated R Movie Else Goto
See Rated PG13 Movie Else Goto
Video Arcade
```

This example fails the readability test because it’s difficult to easily discern the structure. The code can be rewritten like this:

```
If Jeffs Age > 16
Then If Toms Age > 16
Then Goto See Rated R Movie
Else Goto See Rated PG13 Movie
Else Goto Video Arcade
```

See how much easier it is to spot the relationships between the conditions and their associated actions. Also, notice the spacing between the keywords and field names. Spacing is often stressed by programmers, though there are no hard and fast rules as to what constitutes “good” spacing. I like to keep related keywords and actions “lined up,” as shown in the example.

Field names

Choosing descriptive field labels and field names is essential for easy data entry and readable code. I rarely use a field ID number in a program, as it says nothing about the kind of information meant to be in the field. (To see what I mean, replace the field names with field ID numbers in the previous example, and see if you can make sense of the program).

What’s more, if you use field ID numbers and change one, you’ll have to check each field’s program to make sure you’ve changed every occurrence of the old field ID number to the new one.

When you program using field names, you can change a field name by changing its label in database redesign. When you save the redesign—check the Field Names Spec to be sure the field name is what you want—Q&A will update your code, replacing each occurrence of the old field name with the new one.

Logical Yes/No fields illustrate why you might want the field name to differ from the field label. To prompt for a “Y” or “N” response, you might want the field’s label to read *Current?* on the data entry form. Q&A will let you get away with a question mark in a field label, but the program compiler won’t recognize it. You could name the field *Current*, but that doesn’t tell you it’s a Yes/No field. Many programmers handle this by prefacing Yes/No field names with *Is*. This way, you can keep *Current?* as the field’s label, but change the name to *Is Current* at the Field Names Spec.

You can always have two descriptive references to the same field, one on your database form, and the other in your programming code.

Begin-End blocks

Lining up each *END* (or the closing French brace) with its matching *BEGIN* (or the opening French brace) makes your code easier to follow. Some compiler text editors do this automatically, but Q&A doesn't. Here's an example of how it might look:

```
< If      Is Current
  Then {
    REM "This code performs a loop";
  }
Else {
  @Exit;
  REM("Do not allow processing of a non-
    current record");
}
```

Of course, if you're the only one who will ever need to read, debug, or update your programming, then you should use a style with which you're comfortable. My experience is that code is like handwriting—if it's sloppily or inconsistently structured, you might find it difficult to understand later, even if you wrote it yourself. And others are likely to find it *very* hard to decipher.

If @Modified

In June's Program Spec, I said the following If statement would return True (and execute the Then) if the external lookup field value is between 0 and 9:

```
If @Instr("0123456789", @XLu(...)) > 0
Then...
```

True enough, but I failed to mention that in my application the external field contains just one alphanumeric character. Bill Halpern and Gordon Meigs correctly pointed out that the statement would also return True if the XLookup returned a value such as 56.

So, if you use this technique in a situation where the @Xlookup could return a value of any length, you'll need to amend the statement like this:

```
If @Instr("0.1.2.3.4.5.6.7.8.9.", "." + @XLu(...)
+ ".") > 0
Then . . .
```

QA

Jeff Nitka works for a chemical manufacturer and develops Q&A applications part-time for Epoch Software, 908-874-3989. Jeff is the author of the Program Evaluator, a Q&A program debugging utility available from Marble Publications.

[Although a line of Q&A for DOS programming code can exceed 70 characters, we're forced to wrap each line after about 50 characters (including spaces) due to The Quick Answer's two-column format. It pains us to "butcher" our contributors' code this way, and we encourage them to develop a 50-characters-max-per-line style when writing for publication.—Ed.]

Play That Tune ...

Continued from page 13

MYTUNES.DTF would then contain these four fields: Name (the name of the tune), Play (the tune's tone/duration sequence), PlayIt (where you type a "Y" to play the tune), and PlayString (to process the tone/duration sequence, playing the tune).

Now, the next time somebody has a birthday, you can assemble the crew to sing this customary tune, and have Q&A provide the accompaniment:

```
415,300 415,300 466,600 415,600 554,600 523,1200
415,300 415,300 466,600 415,600 622,600 554,1200
415,300 415,300 831,600 698,600 554,600 523,600
466,1200 740,300 740,300 698,600 554,600 622,600
554,100 622,100 554,600
```

QA

Tom Marcellus is editor of *The Quick Answer*, author of *PC World Q&A Bible* (IDG Books), and a wannabe musician. His QuickClick Calendar Plus—a time- and activity-tracking database for Q&A 5.0—is available from Marble Publications, publisher of *The Quick Answer*.

Salieri Would Have Killed for It!*

But you don't have to be a composer to make beautiful music on your PC

Created by *Quick Answer* editor Tom Marcellus, the **Tune-o-matic** database for Q&A 5.0 has everything you need to compose, edit, play & save your favorite tunes.

- Easy to use. Just click on the three-octave table of tones to play audible notes like a musical instrument.
- Record your tones. The **Tune-o-matic** adds them to the sequence as you compose your tune one note at a time.
- If you hit a wrong note while recording, simply click on the Delete Last button and try again.
- Click on the Play button to hear your tune-in-progress. Tones and durations are fully editable.
- Use the pop-up duration selector to add a whole, half, quarter, eighth, or sixteenth note duration to a tone.
- Name and save your finished tune. Use XLookup to play it from another database. Or have the **Tune-o-matic** compile it into an @Play program to copy to another database.
- Mouse driven—no need to type lengthy @Play routines. Built-in Help screens.
- Pre-composed tunes include Ballgame, Ode to Joy, CHARGE!, Happy Birthday, and Dirge.

The Amazing Tune-o-matic

\$49 (Quick Answer subscribers, \$29) + \$2 shipping

Mail check to **Marble Publications, PO Box 9034, Gaithersburg, MD 20898-9034**. Or fax your order with credit card number (Visa, MasterCard, AMX) and expiration date to **301-424-1658**.

*Antonio Salieri (1750–1825). Italian court composer who envied Mozart and plotted against him.

Quick Tip

Useful Tricks with @Askuser and @Msgbox

You can make @Msg messages more descriptive by including in them expressions comprised of functions and fields. The same is true of the @Askuser and @Msgbox commands. For example,

instead of the following non-descriptive three line message:

```
@Msgbox("This person is already scheduled",  
        "with an appointment at",  
        "this date/time.");
```

I can display a more informative message like this:

```
@Msgbox(LastName + " is already scheduled",  
        "with an appointment at",  
        "this date/time. (See record #:"  
        + @XLookupr("Schedule", "ScheduleLink",  
        "RecordNum") + ")");
```

The resulting @Msgbox message looks like this:

```
Dvorin is already scheduled  
with an appointment at  
this date/time. (See record #23)
```

@Msgbox and @Askuser require three lines and three parameters, each separated by a comma and enclosed in quote marks. If you don't have three lines of text to display, you can use "" as the parameter as in the following two examples:

```
@AskUser("", "Continue?", "");
```

```
@MsgBox("Phoenix", "Solutions", "");
```

QA

David Dvorin

Quick Tip

Counting Your Blessings II



In the June *Quick Answer*, Tom Marcellus showed how to count opening and closing parenthesis (or French braces) in a program to make sure they're balanced.

Having encountered the same frustration

many times, I use a similar method you might find even more convenient.

While programming, you're normally working in the Field Editor—a mini version of Q&A's word processor—where you can use the Search dialog box to count your parentheses or braces. To do this, press F7, type an open parenthesis on the *Search for...* line, select Fast Automatic, then press F7 or F10. You'll get a

message like this at the bottom of the screen:

```
Automatic search COMPLETED after 5 matches
```

Press F7 again, replace the opening parenthesis on the *Search for...* line with a closing parenthesis, press F10, and Q&A will report how many of these it found. If the second count is lower, you have too few closing parentheses. If it's higher, you have too many. It's fast, accurate, and a lot easier on the eyes. **QA**

Gordon Meigs

[Yeah, well I prefer manual methods over automated ones.—T.M.]

PO Box 9034
Gaithersburg, MD 20898-9034

Periodicals
Postage Paid at
Rockville, MD