

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>

Add a Pop-Up Callback Scheduler To Your Database

TOM MARCELLUS



If the kind of work you do involves scheduling follow-up calls with people (or scheduling tasks in general), an automated “callback scheduler” in your database might be just the ticket.

In the sample application I’ll show, you go to the **CallbackDate** field in your database and press Tab or Enter. If **CallbackDate** is empty, a pick-list of callback intervals pops-up (see Figure 1). You select the interval you want and Q&A figures out what the date should be and pops it into the field. If **CallbackDate** already has a date in it and you want to schedule a new date, you simply press Shift-F4 to delete the old entry first. **Callback.dtf** is included in this month’s download file for Online Edition subscribers.

The callback intervals on the list are fully customizable. You can revise the programming to accommodate longer intervals or any range of intervals you might need. We’ll get into the programming momentarily. Once you see how it derives an actual date from the various interval options, you should be able to set your own custom intervals.

Callback report a plus

With nothing more than this callback scheduler gizmo in your database, you’ll have to search the database each morning on the **CallbackDate** field to find the calls you need to make today. This might not be practical for the way you work. So, I’ll also show you how to design a report (that you can print to screen, if you like) that shows who needs a callback today. I’ll give you a few options for this report as well, such as one that shows today’s and tomorrow’s callbacks, one that shows callbacks scheduled for Saturday and Sunday only when you run the report on Monday (or Friday), and one that shows only the scheduled callbacks for the particular user who’s running the report when the database is shared.

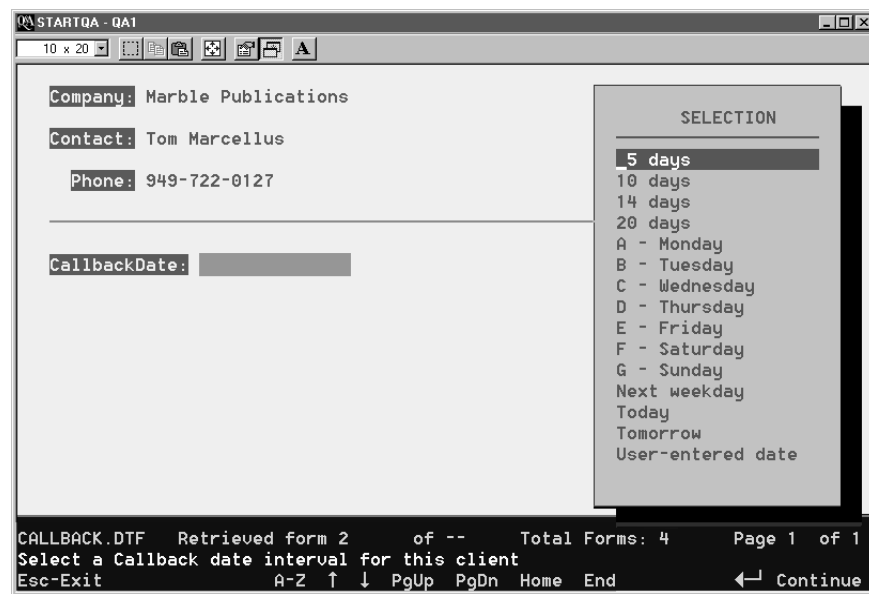


Figure 1. The pop-up scheduler makes it a snap to schedule your callback or reminder dates.

The Quick Answer

The Do-It-Yourself Guide to Q&A

July 2001

Volume 12 Number 7

- 1 Add a Pop-Up Callback Scheduler to Your Database
Tom Marcellus
- 2 *Tip*—Date Programming Tricks for Merge Documents
- 2 Letters
- 4 The Mechanics of Speedy Fields
John Dow
- 6 @Help—*Edited by Bill Halpern*
 - *When @Macro Won't Work*
 - *Date Field Problems*
 - *Deleted Macros Resurfacing*
 - *Goto Commands in Mass Updates?*
- 10 *Tip*—Auto-Add New Records to a History Database
- 10 *Tip*—On the Blink in Full-Screen Mode
- 11 *Tip*—Temporarily Disable a Block of Programming Code

MARBLE
PUBLICATIONS

The database

Callback.dtf as shown in Figure 1 is a bare bones affair designed to showcase only the callback feature. It contains fields for the **Company**, **Contact**, and **Phone**, along with the two fields required for the scheduler. The main field, **CallbackDate**, must be formatted “D” for Dates. **CallbackTemp** (a text field you don’t see because it doesn’t have a field label and is Read-Only) is needed for interim calculations.

The scheduler pick-list is designed to accommodate a variety of ways of

Continues on page 3

Quick Tip

Date Programming Tricks for Merge Documents



I print my monthly billing statements using a Q&A for DOS merge document. I always print them within the first few days of the month, but since the work I'm billing for was performed during the previous month, I want the printed "Statement Date" to be the last day of that previous month. Here's how I program the merge document to get this:

```
Statement Date: *Pg { @Mt$ (@TD (@Mt$ (@Da) + " " + "1, " + @Str (@Yr (@Da)) - 1) ) } * *Pg { @Dom (@TD (@Mt$ (@Da) + " " + "1, " + @Str (@Yr (@Da)) - 1) ) } * *Pg { @Left (@Yr (@TD (@Mt$ (@Da) + " " + "1, " + @Str (@Yr (@Da)) - 1) ) , 1) } **Pg { @Mid (@Yr (@TD (@Mt$ (@Da) + " " + "1, " + @Str (@Yr (@Da)) - 1) , 2, 3) } *
```

There are actually four `*Pg { . . } *` commands here. The first one returns the previous month and the second one finds the last day of that month. The third and fourth parse then reassemble the year because Q&A returns "2001" as "2,001" (with an unwanted comma).

With this programming, if I print my statements on any day in July 2001, the statement will include the line:

```
Statement Date: June 30, 2001
```

This way I hope to keep clients from getting the idea that it's a "July" bill that they can sit on for "Net 30 Days" and pay in August (for work that might have been performed two months earlier!).

For added incentive, I also include in the document:

Please Mail Your Check by *Pg { @Mt\$ (@Date) + " 15th" } *

This tells the client that I expect them to mail their check by the 15th of the *current* month.

This programming saves me from having to change the date information in my merge document before printing my statements each month.

Here's a database Retrieve Spec expression (for a field named "Date") that finds or includes all records dated any date in the month previous to the current month:

```
{ @Left (Date, 7) = @Left (@Date - @Dom (@Date) , 7) }
```

Tom Marcellus

The Quick Answer™

Editor / Publisher
Tom Marcellus

The Quick Answer (ISSN 1052-3820) is published monthly (12 times per year) by Marble Publications, Inc., 1927A Harbor Blvd., Costa Mesa, CA 92627 USA

Cost of domestic subscriptions: 12 issues, \$109; Outside the U.S., \$129; Online edition, \$79. Single copy price: \$11; 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.

POSTMASTER: Send address changes to The Quick Answer, 1927A Harbor Blvd., Costa Mesa, CA 92627.

Copyright © 2001 by Marble Publications, Inc. All rights reserved. No part of this periodical may be used or reproduced in any fashion (except in the case of brief quotations embodied in 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, 1927A Harbor Blvd., Costa Mesa, CA 92627 Phone 800-780-5474 or 949-722-9127. Fax 949-722-9127, mailbox@quickanswer.com

On the Web at www.quickanswer.com

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, 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.

Reach Us

Phone 800-780-5474 / 949-722-9127
Fax 949-722-9127
Email mailbox@quickanswer.com
Web <http://www.quickanswer.com>
Mail Marble Publications
The Quick Answer
1927A Harbor Blvd.
Costa Mesa, CA 92627 USA

Letters

Comments on the June 2001 Issue

Regarding your article on hot-links and switching between related databases: While Q&A for Windows has no way to switch between databases, with the addition of my DTFWIN you have several options. If you have more than one database open, pressing F11 will cycle through all of them. (If there are just two open, F11 will toggle back and forth between them.) Using DTFWIN's scripts, you can switch to a specific database by making a menu selection, by clicking on a button, or by having it under the control of the database programming.

Regarding maintaining unique database values when importing new records (page 6), I've added to DTFDATA several options for detecting and reporting duplicate records. (See the November 2000

issue.) It is very easy to generate a report that shows all records that are duplicates, whether there are two or 200 records in a set. DTFDATA automatically puts a blank line between sets of duplicate records.

Regarding page defaults in Q&A for Windows (page 7). There are other printing options that Q&A doesn't remember, not just page margins. A solution is to use DTFWIN which, like a Q&A for DOS macro, can change things for you on the fly. Just select the report in question from the menu and the script would set the margins before running the report.

John Dow

thinking in terms of future dates. Some of the items on the list are expressed simply as a number of days from the current date, while others accommodate people who work (and think) in terms of “next Wednesday” or whatever the next weekday is if they happen to be working on a Saturday or Sunday. But as I mentioned earlier, everything on the pick-list is customizable.

Let's get to the programming that makes the scheduler work.

Programming

The only programming you need goes in the **CallbackDate** field. (See the sidebar.)

As you can see, the program executes as you exit the **CallbackDate** field.

An initial check is made on the **CallbackDate** value. If it contains a 20th century date, it's automatically converted to a 21st century date. This lets you manually enter callback dates with just the two-digit year, such as 7-12-01 (or just 7-12 if the callback date is in the current year), saving you a few keystrokes. Without this conversion, a date entered literally as 7-12-01 would be interpreted and stored by Q&A as 7-12-1901. (This is one reason why the global date formats in all your databases should be set to one of the options that displays the full four-digit year.)

If **CallbackDate** field is empty, the scheduler (via the **Userselect** command) pops up for you to choose a callback interval, and copies your selection to the **CallbackTemp** field. The programming as it stands forces a selection if a user-entered date hasn't been entered.

The next block of programming looks at the temporary value in **CallbackTemp**. If you selected one of the weekday selections (A - Monday through G - Sunday), it uses the **@Dow\$** (day of the week) function plus the

current date to determine the date that that next day of the week will be. The programming is rigged so that if today is Tuesday and you select Tuesday, the callback date will be set to *next* Tuesday. But if today is Tuesday and you select Wednesday, tomorrow's date will be inserted in the **CallbackDate** field.

The next block of programming handles manually entered dates. If you select *User-entered date* from the pick-list, the program prompts you to type in the date and press Enter to confirm it.

Concludes on page 11

```
> If CallbackDate <> "" and @Yr(CallbackDate) < 1950 Then
CallbackDate = @Replace(CallbackDate, @Yr(CallbackDate),
@Yr(CallbackDate) + 100);

If CallbackDate = "" Then {
@Msg("Select a Callback date interval for this client");

Userselect("User-entered date,Today,Tomorrow,Next weekday,
5 days,10 days,14 days,20 days,A - Monday,B - Tuesday,
C - Wednesday,D - Thursday,E - Friday,F - Saturday,
G - Sunday", CallbackTemp);

If @Instr(CallbackTemp, "-") = 3 Then
{
CallbackTemp = @Del(CallbackTemp, 1, 4);
If @Dow$(@Date + 1) = CallbackTemp Then CallbackDate = @Date + 1
Else If @Dow$(@Date + 2) = CallbackTemp Then CallbackDate = @Date + 2
Else If @Dow$(@Date + 3) = CallbackTemp Then CallbackDate = @Date + 3
Else If @Dow$(@Date + 4) = CallbackTemp Then CallbackDate = @Date + 4
Else If @Dow$(@Date + 5) = CallbackTemp Then CallbackDate = @Date + 5
Else If @Dow$(@Date + 6) = CallbackTemp Then CallbackDate = @Date + 6
Else If @Dow$(@Date + 7) = CallbackTemp Then CallbackDate = @Date + 7;
Clear(CallbackTemp); CNext};

If CallbackTemp = "User-entered date" Then { @Msg("Type in the
Callback Date and press Enter");
Clear(CallbackTemp,CallbackDate); Goto CallbackDate};

If CallbackTemp = "Today" Then {CallbackDate = @Date;
Clear(CallbackTemp); CNext};

If CallbackTemp = "Tomorrow" Then {CallbackDate = @Date + 1;
Clear(CallbackTemp); CNext};

If CallbackTemp = "Next weekday" Then
{
If @Dow$(@Date) = "Friday" Then CallbackDate = @Date + 3
Else If @Dow$(@Date) = "Saturday" Then CallbackDate = @Date + 2
Else If @Dow$(@Date) = "Sunday" Then CallbackDate = @Date + 1
Else CallbackDate = @Date + 1;
Clear(CallbackTemp) };

If CallbackTemp = "" then Goto CallbackDate
Else {CallbackDate = @Date + @Tn(@Num(CallbackTemp));
Clear(CallbackTemp); }
};

If CallbackDate > @Date + 30 Then If
@Askuser("Follow-Up is more than 30 Days away!","Do you really
want to leave it that way?"," ")
Then CNext else Goto CallbackDate;

If CallbackDate < @Date Then If
@Askuser("Follow-up is earlier than today!","Do you really
want to leave it that way?"," ")
Then CNext else Goto CallbackDate
```

The Mechanics of Speedy Fields

JOHN T. DOW



THIS article is about Speedy fields. It should help you when designing or maintaining a database so that you get good performance when searching for records and when adding and updating records.

Speedy fields are important for two reasons: 1) retrieving records using a Speedy field can be literally 100 times faster than retrieving using a field that isn't Speedy, and 2) XLookup and related commands require that the "key field" in the target database be a Speedy field.

For those of you who only use Q&A for Windows, a "Speedy" field is an "Indexed" field. The terms "key field" and "key value" are also sometimes used to refer to a Speedy field and its contents, as these are more generic terms.

I'm writing this primarily to take two positions that may be somewhat controversial. First, I don't think there's any absolute number of Speedy fields beyond which performance hits a wall. Second, duplicate values in fields are just fine unless you have far too many of them. I'll explain both in more detail below. I want to make these points because if you think you can't do something and don't, you may be missing some of the value of your database if it turns out you could have done it after all.

I'm going to say almost nothing about the syntax of a Retrieve Spec. Alec Mulvey has covered this issue very competently in past issues. He also reviewed this article and made several suggestions that I have incorporated.

My statements here are based on observation and examination. I am thoroughly familiar with the structure of both the DTF and IDX files but I do not have the source code for Q&A and cannot speak based on an actual review of the code.

By observation, I mean that I have systematically tested Q&A's behavior under conditions that permitted conclusions to be drawn with some certainty. For example, when accessing a large database under bad conditions (for example, a slow network with a slow server), it is possible to actually time the performance in different scenarios with a sweep second hand.

By examination, I mean that to write programs such as DTFDATA and DTFDOCTR, I had to examine and come to understand the structure of the DTF and IDX files in great detail.

How many Speedy fields can a database have?

A database can contain up to 115 Speedy fields.

How long can a key value be?

Q&A uses only the first 16 characters of a text field. Anything beyond that is ignored.

Dates and numbers are converted into a special internal format when stored in the IDX (index) file, so unlike with text fields, length is never an issue.

An interesting detail: using a text field with less than 16 characters might make your DTF file smaller but it won't affect the IDX file. It allocates 16 characters, no more, no less, regardless of what you use.

What is the structure of the IDX file?

Key values are stored in an IDX file in a way that permits Q&A to locate records very efficiently.

Each Speedy field has its own tree structure within the IDX file. Each of these trees uses what's called a B-tree structure. This is a very efficient tree structure, and Q&A's retrieval speeds can be very impressive.

To find a record using an index with a tree structure, you would start at the top and compare the value you want to find against the values presented to you at the top. Based on that comparison, you would then go down to the next level and again compare the value you want to find against the values presented at that level. You would continue doing this until you reached the bottom, where you would find links to data records.

Each node in Q&A's IDX tree can point to as many as 20 or more nodes below it. In practice, they're not full. Suppose they only have 10 on average. That means that if you have 50,000 records, after doing the first comparison you are now only looking at 5,000 records. The second test and you're at 500. The third and it's 50. The fourth and it's 5, at which point you can go directly to the record. So it only takes five reads of the IDX tree to locate a single record among 50,000.

If you have duplicate keys, Q&A stores them very efficiently in the IDX file. It will store 100 duplicate keys in the space it takes for 20 unique keys.

How are Keyword fields handled?

Each value in a Keyword field gets a separate entry in the IDX file.

If your Keyword field in a given record contains "fishing;swimming;biking" and it's Speedy, there will be three entries in the IDX file for this one field in this data record.

Do Speedy fields make adding records slower?

Yes, but the cost is the same for each Speedy field, it doesn't take disproportionately more time with more Speedy fields.

The cost of adding a record varies linearly with the number of keys. Let's say it takes an average of 1 second to add a record across a network even if it has no Speedy fields. Let's say that if you make one field Speedy it takes 1 additional second, for a total of 2 seconds. I believe that 2 Speedy fields would take 3 seconds, 3 would take 4 seconds, 4 would take 5, 5 would take 6, 6 would take 7, and so on. I have no reason to believe that the cost goes up any faster with larger numbers of keys.

Here's what happens to the IDX file when you add records.

If you are adding a new record, then an entry has to be added into the tree structure for each Speedy field. Q&A has to go down its tree to find out where to insert the value for the new record. It can insert the information into a bottom level node with little cost because most of the time there's room for another key. However, if there is no room in the node for another entry, Q&A has to split the node and change the tree structure accordingly.

If there's room for 20 keys at the bottom node and it gets full and has to be split into two, then each of the two new nodes would have only 10 keys. Each could add another 10 before splitting again. Therefore every 10th new key will cause a split, so 9 out of 10 will be very fast while 1 out of 10 won't be so fast. (I am not certain this is exactly how Q&A splits nodes, but you get the picture.)

Do Speedy fields make saving existing records slower?

Yes, but only if you change the value of one or more of the Speedy fields.

Q&A seems to realize whether a Speedy field has been changed (either by keying in a value or as the result of programming) and is smart enough to not do anything in the IDX file if it doesn't need to.

What this means is that you can have 30 keys in a record and pay a price when the record is added to the database but then not pay any price to maintain the record if you don't change the key values.

If you want to find customers using driver's license, social security number, zip code, last name, first name, your own customer ID number, telephone numbers (home, work, mobile, fax), and mother's maiden name, then I say make them all Speedy. Unless you change these fields, the fact that they're all Speedy won't affect performance.

What is the impact of duplicate keys?

Keys that are duplicated many times are to be avoided because they can hurt Q&A's performance.

If a key values appears several times, that is not a problem. For example, if you make last name a Speedy field, even though certain names are quite popular and will certainly be repeated, that should not make any

difference in performance. You could also make ZIP code Speedy even though hundreds of customers will have the same ZIP code.

If a key value only has a small number of values (for example, a Customer Gender field is M or F), thousands and thousands of records will have the same key value. That will hurt performance.

A blank key field can result in duplicate values and should be avoided. That is, if a field is Speedy, make sure it's rarely left blank.

I suspect when people are having problems with Speedy fields, the number one culprit is many or even most of the records have blanks in a Speedy field.

For example, suppose you have a field in your customer database for a mobile telephone number and you make it Speedy. Suppose that initially only 50% of your customers actually have a mobile phone number. If you have 50,000 customers, 25,000 have the same mobile phone number (blank). That will hurt Q&A's performance when you add a new record that also has a blank mobile phone number.

Here's something you can do to avoid that problem. Assuming that each record has a unique customer ID number, put the customer ID number into the mobile phone field with on-record-exit programming if the user has left the field blank. If customer numbers in telephone number fields might be confusing, put them in like this "(12345)" and train your operators to ignore parenthesized values in key fields. You might also use @Date or @Time as a way of generating a value that's reasonably unique.

What difference does it make if a field is SU?

When a field is Speedy-Unique, there's no difference in performance when adding records; no difference retrieving records. There is no difference in the way the index is built. When you attempt to add the value, Q&A checks to see if that value is already in the database, but otherwise there is no difference.

How many Speedy fields should a database contain?

That's for you to decide based on what works for you. There's no intrinsic limitation that I can detect. It's all a matter of performance and trade-offs.

Some, perhaps most, people think that if you have more than some small number of Speedy fields that Q&A's performance will suffer terribly.

In reviewing a discussion thread from 1998 in which someone complained about database speed, I found people saying 3, 4, and 6 were the critical number of Speedy fields beyond which performance would die.

I do not agree that there is any particular number beyond which Q&A's behavior drastically changes. I see no reason why there should be a threshold.

I have also done experiments and don't find a point at which performance suddenly degrades.

Continues on page 8



@Macro Doesn't Work



No matter what I try, I can't get an @Macro command to execute even in a simple database. And yes, it's the last statement in the program. There's no "Goto" or anything else following it in the Navigation or Program Spec. The program is in field #1 which contains the Customer ID:

```
>#1: @Macro ("alt6")
```

The alt6 macro in the automac.asc file looks like this:

```
<begdef><alt6><name>"<caps,>alt6<caps.>"<vidoff><altf9>5<enddef>
```

I've made sure to load the macro file (automac.asc) before starting the process. The "5" in the macro is the 5th entry on the "External Programs" menu, which looks like this:

```
5 - Call CB List d:\qa\qa.com -alcblist.mac -m1 %g
```

I'm using the %g variable to pass the existing field value of field #1 (the Customer ID) to the second copy of Q&A in order to generate a pick list of vehicles for that customer from another database's Restrict Spec. Any suggestions?

Jim Wilson

The answer is right in the macro file you mention. If you look at it carefully, realizing that the name of the Macro is *everything* between the quotes, you'll see that the actual macro name is "<caps,>alt6<caps.>". In a macro file, Q&A uses certain abbreviations enclosed in angle brackets to delineate shifted or special characters. For example, the up arrow key is <up> in a macro file, the down arrow is <dn>, the space bar is <sp>, the enter key is <enter>, the Shift-Tab key-combination is <capstab>, and so forth. Whenever you see a <caps> in a macro, Q&A is telling you that the next keystroke is the one you get when you press that key in combination with the Shift key. Therefore, <caps,> is Shift-comma, and <caps.> is Shift-period. This means that the *real* name of the macro is not alt6 (that's just the keyboard combination) but <alt6>, so your programming statement should be:

```
>#1:@Macro ("<alt6>")
```

One way to prevent problems such as yours is to never let Q&A assign default names to macros. Always specify a name when you save the macro. For example, you could have named this macro "Vehicle Pick List."

Date Field Problems

I use date fields heavily in my Q&A 5.0 databases. The problem I have is when some of my users enter future dates by typing "04/05/04" and Q&A puts "04/05/1904" in the date field. We can take care of this by typing the full four digits of the year, but some of my data entry people forget to do this. Is there some setting I'm missing?

There's more. If someone enters an invalid date such as "September 2002" in a date field, Q&A warns that "*This date can't be reformatted...*" But if the user presses Enter again, the message goes away and the incorrect information stays in the field. This can't be the way Q&A is supposed to work! My data is always getting loused up this way. What can I do to take care of this?

Bart

Well, you've experienced first hand a prime example of the problems that Q&A's "loose" data typing can cause. A date field should allow only a valid date entry, just like a number field should only allow a number, a time field a time of day, a Yes/No field a "Y" or "N," a unique field a unique value, and so on. (This will be the case in the new Q&A-compatible database under development by Lantica Software.) But, Q&A is "forgiving" and, in most cases, lets you override its warnings about incorrect data types.

In the case of date fields, there's a solution to both your problems. Add the following program to the date field(s) and not only will Q&A interpret two-digit years correctly but, if an incorrectly formatted date is entered, it will clear the field when you press Enter after the warning message. Here's a sample program for a date field named Date1. (It executes when you leave the field.):

```
>If Date1 < "1/1/1944" Then  
Date1 = @Td(@Str(@Lt(Date1, 4) + 100) + "/" +  
@Str(@Mid(Date1, 6, 2)) + @Str(@Rt(Date1, 3)))
```



Stumped?

Send your Q&A questions to @Help, The Quick Answer, Marble Publications, Inc., 1927A Harbor Blvd, Costa Mesa, CA 92627 or email to mailbox@quickanswer.com. Include your name, address, phone, and your

Q&A version number (and whether DOS or Windows) and a detailed description of the problem. We'll publish those questions we feel are of general reader interest; individual responses aren't possible.

Two things happen here. First, if a valid date is entered with a two-digit year, Q&A will interpret it as year "19" plus the two digits. If the result is earlier than 1944, the program adds 100 years to the date and plugs in the right century. On the other hand, if a non-date value is entered, it will be interpreted as earlier than 1944. In this case, the program will try to convert it into a valid date using the @Td (to date) function. Since the result will still not be a valid date, Q&A will clear the field.

Deleted Macros Resurfacing



I have a mysterious problem. One of my people created a macro and inadvertently assigned it to the Up-Arrow key. Thereafter, anytime anyone tried to navigate in a form using the arrow keys, all kinds of strange things would happen because the macro was pretty complex. I saw this went into the qamacro.asc file and deleted the offending macro. I then saved the file and thought everything would be fine. To my dismay, the Up-Arrow key macro keeps reappearing. I've deleted it at all seven users' stations but it still comes back! How do I get rid of it for good?

A second problem we are having is that we often have to reboot the server to get back our 7-user count. With everyone out of Q&A, the first user often gets the message that "of the seven network users, seven are already in use." Do you have any idea what is causing this?

Vince

Well, you almost had us stumped with this one. But, the second part of your message gave us the clue. We spoke on the phone and confirmed our suspicions and fixed the problem.

You're using Q&A on a network with two Netpacks, which gives you seven users. But (1) you are not using personal paths at your workstations, (2) your network is not set up properly or is unstable, and (3) your people are not exiting Q&A properly.

When everyone is running the same copy of Q&A via Netpacks without personal paths, they are all using the same Q&A program files, the same QA.cfg (configuration) file and the same macro file. During our discussion, we confirmed that the network still showed that both qa.dis and qa1.exe were open on multiple PCs even though nobody was in Q&A. This indicates that your people are not exiting Q&A from the Main menu and are therefore not releasing themselves as users. (They are probably "closing" Q&A by clicking the 'X' button in the top right corner of the Q&A window and overriding Windows' warning; or worse, shutting down their PCs while Q&A is still running. Either of these bad habits will do more harm than just keeping the user count high, they will damage the qa.cfg file that everyone depends on. QA.cfg is the file that tells Q&A what it is connected to, what the default path is, what printers are installed, and so forth. Damage to this file can actually corrupt your databases.

As far as the macro file is concerned, since you are all

using the same one, the last one to quit Q&A is the one that controls the macro file. (Q&A stores the macro file in memory once it starts. It does not read the macros from disk.) Even if you have deleted a macro, another user can make it "reappear" by just saving the macros at their station at a later time. Q&A writes their macros back to disk. So anyone creating a new macro or just pressing Shift-F2 / Save macros, will make your offending macro reappear. You must have everyone exit Q&A *properly*, make sure that the user count is at 1, and then delete the bad macro. This will get rid of it forever.

In addition, you *must* institute a policy of personal paths at the workstation. Since you're running Q&A 5.0 with *Lesspace*, you'll need to create multiple startqa.bat files. There should be one for each user (startqa1.bat, startqa2.bat, startqa3.bat, and so forth.) All the batch files can be in the same folder (you told me that Q&A was installed on a network drive mapped as I: on each of the workstations), but each must be used to call Q&A from only a single workstation. The batch files should look like these:

Startqa1.bat

```
@echo off
lesspace
qa.com -I:\qa5\user1 -aI:\qa5\qamacro.asc
lesspace /u
exit
```

Startqa2.bat

```
@echo off
lesspace
qa.com -I:\qa5\user2 -aI:\qa5\qamacro.asc
lesspace /u
exit
```

Startqa3.bat

```
@echo off
lesspace
qa.com -I:\qa5\user3 -aI:\qa5\qamacro.asc
lesspace /u
exit
```

And so on through startqa7.bat.

You will then have to create the folders *user1*, *user2*, *user3*, *user4*, and so on under the QA5 folder and do a one-time copy of qa.cfg into each of them. (This will save you from having to reset all of the preferences on each computer.)

Bring in a good network person to go over your network wiring, hubs, and protocols.

Betcha this solves all your problems!

Goto Commands in Mass Updates?



I need to be able to run a Mass Update on my database to perform a bubble-sort on records. The problem is that the bubble-sort contains multiple *goto* commands and Q&A won't let me save the programming as a valid Update Spec. I've tried to change the programming to use *gosubs* instead

Continues on page 12

To the best of my knowledge, there is only a cumulative effect as you add more Speedy fields. I don't see a discontinuity.

You need to balance the cost of adding a new record against the cost of retrieving it. You only add the record once, so you pay that cost once. Typically, records are retrieved much more than one time, so you pay that cost many times. Therefore, you should be more concerned with the cost to retrieve rather than the cost to add.

If you need to do rapid searches on 10 fields, by all means make all 10 of them Speedy.

Time how long it takes to recover your database (always copy it to a local drive so that you have a backup and aren't recovering on the network drive). Since a recover rebuilds the IDX file, the more Speedy fields, the longer it will take. You might find that your threshold is reached when you do a recover and the database has 10 Speedy fields.

I just conducted a test on a 19MB database with 31,000 records and found that removing the Speedy fields before recovering, then adding them back one by one, required ½ the time that recovering with the Speedy fields in place required. By the way, I found no difference whether the database Sharing Mode was set to Automatic, Allow or Disallow. So to save time, the recommendation is to remove the Speedy fields, run the recovery, then add them back.

Removing and adding back the Speedy fields entails a certain risk and an inconvenience. If you add them back incorrectly, performance will suffer. XLookups might not work. Therefore, I have just added a new option to DTF SPECS so that before you remove the Speedy fields, you can use it to build a macro that will add them all back once you have done the recover. You can start the macro running and forget it until they're all done. (If you have a recent version of DTF SPECS (2.5x), email me and I'll send you the updated version at no charge.)

When can Q&A not use the IDX file?

There are several conditions which force Q&A to scan the entire DTF file to find the record(s) you want.

If you specify a retrieval that doesn't use any Speedy fields, Q&A doesn't use the IDX file.

However, even if you're searching on one or more Speedy fields, there are several conditions which Q&A wasn't designed to handle using the index.

- Avoid starting your value with "...". If you type ".Smith." rather than "Smith." or "Smith", Q&A won't use the index. If this is the only Speedy field you're searching, Q&A will read through the entire DTF file.
- Try to avoid starting your value with "/" (meaning "not"). If this is the only Speedy field you're searching on, Q&A will have to read through the entire DTF file.

Sometimes you can use a range (rather than using "/" and get better performance. I just did a test on a database I had with 37,000 records. One of the fields is a phone number and only 100 records have data in this field. I made the field Speedy, which took several minutes. I then tested it across a network (to slow it down a tad). First I did a search on a non-Speedy field for a value that doesn't exist to see how long it took to scan through the entire DTF file. Result: five seconds. That's the baseline.

Then I tried the Retrieve Spec "/=" , which came up instantly with the first record having a phone number. But when I pressed Ctrl-End it took five seconds. (It scanned the entire DTF.) The Retrieve Spec ">=" (" came up instantly and pressing Ctrl-End only took ½ second. This difference between ½ second and five seconds would be magnified somewhat on a busier network than my two computers, so you can see that this makes a big difference.

Why did I search for "greater than or equal to left parenthesis"? Because some of the phone numbers begin with a left parenthesis. I had first tried ">=0" but it missed some records—those that started with a left parenthesis.

If you are careful and check out the Retrieve Spec syntax you're using, Q&A can take advantage of the Speedy fields that exist. If you're not careful, Q&A may end up doing a full scan of the DTF file.

Does Retrieve Spec programming make Q&A slower?

Yes. Using a retrieval expression enclosed in curly braces (" {...}") makes retrieval time take much longer. However, the more you can use Speedy fields to restrict the set of records to be evaluated, the faster the search will go.

Some empirical results

I just did a few experiments with a database that is typical in size. It has 28,000 records. The DTF and IDX files are 70 and 40 MB respectively. The database has seven Speedy fields, all with fairly unique key values. (I say "fairly unique" because one field is a first name and there were 800 records with "John" in the first name field, which is Speedy.) I added records (including another "John") and did various searches.

Searching a non-Speedy field took 30 seconds. That's the time it took to read through the DTF file. (From a networked computer it took 90 seconds.)

Searching Speedy fields was either instantaneous or on the order of one second for the worst cases. It even handled entering a record with a seriously duplicated key (all but one of the 28,000 records were blank) without taking more than a second.

I then made more fields Speedy so that there was a total of 18 of them. (The IDX file expanded to 74 MB.) According to conventional wisdom, 18 Speedy fields are way too many. Yet I added records and noticed no difference in performance.

Finally, I added records from the networked computer while the database had 18 Speedy fields. Adding a record with values in the Speedy fields took one to three seconds.

Adding a record that was virtually all blank (so that for a number of the Speedy fields it would be a duplicate key value) took seven seconds.

These times show that accessing across a network slows it down. If you're using a database all by yourself on a local drive, you could have far more Speedy fields with far better performance than someone with a database on a server being hit by many users. My time of one to three seconds to add a record to a database with 18 Speedy fields with only one person accessing the database across a network could take five times longer with several dozen users in the system and potentially accessing the same database. (In practice, many users will not actually be using the database during any particular three-second window because they will be writing something, reading something, talking on the phone to someone, or they may be accessing another database.)

I did another experiment. Using a database with 28,000 records, I removed all the Speedy fields then added a record from a networked computer. It was too fast to clock any time with a sweep second hand. I then made a field Speedy and tried again. I repeated this until 10 fields were Speedy. The result? Adding a record with up to four Speedy fields was under a second, adding a record with five to 10 Speedy fields was about a second. There simply is no threshold as far as I can tell.

Another interesting fact was the time to restructure the IDX file as I kept adding Speedy fields. The rule I observed was that the time to add a field did not vary depending on which field it was. However, the time to add did vary according to the *type* of field it was. For example, a unique field (customer ID number, for example) took 105 seconds (locally, not across a network). The database has several fields like this and they all took the same time. Indexing names was faster, varying between 80 and 90 seconds for four of these fields. I believe that the fact that there is some duplication and hence fewer unique index entries actually results in a smaller tree, which is faster to build (and search, too, I should think). The worst case was indexing fields with many blank values.

You can't extrapolate times from my results to your own situation because there are so many variables with workstation performance, network cards and configuration, and server performance. However, for the record, my workstation was a Win 98 SE system running a 3xx Celeron processor and the so-called server is my laptop, running Win 95 with a Pentium 166. I don't know the specs on the hard drive.

Conclusions

- Speedy fields can yield tremendous improvements in retrieval performance.
- Having lots of Speedy fields doesn't hurt retrievals.
- Speedy fields cost during operations such as adding a

record or recovering the database, but there's no threshold number of fields. The number of Speedy fields you should use depends on your needs, the characteristics of your database and the system environment. My point is: If you think you need more Speedy fields than you have, don't be afraid to try adding more.

- Speedy fields with some duplications are not a problem. Because the IDX tree is smaller, they may actually improve performance. Examples are last names and postal codes.
- Speedy fields with massive duplication are a problem and hurt performance. Also, remember that 10,000 records with a blank field are just as bad as 10,000 records that all have "X" in the field. Either way, it's a duplicate field and bad. You may be able to substitute a value that's reasonably unique and can be recognized as not a valid value.
- Finally, keep careful notes and experiment. Use a stopwatch or sweep second hand to get actual times for doing various operations. For example, adding Speedy fields will slow down a recovery or the time it takes to import data into the database. (Remove Speedy fields before recovering, then add them later. This will save time. DTFSPCECS can be used to build a macro to restore all the Speedy fields automatically.) You might decide after making more fields Speedy that it's had an unacceptably adverse effect on something other than just retrieving records.

John Dow is the creator of a host of valuable Q&A utilities, including DTFDATA, DTFDOCTR and DTFWIN, and he developed the patch that enables Q&A for Windows to run in Windows 2000. He is also a principal of Lantica Software, LLC. www.johntdow.com, 412-973-9473.

[If Q&A's "Intelligent Assistant" has been "taught" for a database, there might have far more Speedy fields in the database than you need, slowing performance when adding and updating records. A trip to the Speed-Up Spec (File / Customize / Speed-Up Searches) will tell the story. —Ed.]

Need Help? John's consulting services are available to help you solve your problems, whether with Q&A or with other databases and languages. He has worked with a wide variety of computer systems and has even created programming languages and complete database systems from scratch when required for special purposes. He excels at problem solving and tool building. In addition to his technical credentials, he communicates in plain English and can understand your business needs. The end result is that you have a new and/or improved system that works and works well. For further information, e-mail john@johntdow.com or call 412-973-9473.

Quick Tip

Auto-Add XPostable New Records to a History Database



Suppose you need to auto-fill a new record in a history database for each payment entered in your invoice or statement database. This makes it easier to track payments by date received, by customer—or whatever—and generate payment history analysis reports. Here's one way to do it:

1. Create the history database with all the required fields and programming. The database *must* include a Speedy Key field. Let's say it's named **Key** and is field ID #1 in the Program Spec.
2. Create an ASCII file that contains, say, 100 lines of "X"s so that if you imported this file into a database, you'd get 100 new records with an "X" in the first field. Name this file HISTORY.XXX and store in the default database directory.
3. Create an On-Record-Entry field named "Lookup" in the invoice database. Let's say it's field ID is #481.
4. Program field #481 (Lookup) as follows:

```
< #481: XLu("History", "X", "X#1", "X#1", #481);  
if @Error then @Macro("<altx>"); #481 = ""
```

5. Add the following Alt-X macro to your macro file:

```
<begdef><altx><name>"<caps ,>altx<caps .>"<vidoff><f7>  
<capsf8>lu<enter>y<esc><esc><esc>fuis<capsf4>HISTORY.XXX  
<enter><capsf4><enter>history<enter><f10><f10><esc><esc>  
fs<capsf4><enter>INVOICE<enter><altf8>lu<enter><f10>  
<enddef>
```

This way, when you open the invoice database, the Lookup field's program looks for a history record with an "X" in the key (#1) field. If it doesn't find one, it runs the Alt-X macro that imports the HISTORY.XXX file into the history database, creating 100 essentially empty new records.

It's now a simple matter to run an Xpost from within the invoice database to auto-post any field or fields to the history database when a payment is received.

Ken Charters, ken@pcpro.com.au

[We've warned that XPosting to a Speedy field, which this technique does, can corrupt the target database. So use this approach at your own risk. For a different technique that auto-creates records to XPost to, see "Automatically Create Records to Post To" in the March 1996 issue. —Ed.]

Quick Tip

On the Blink in Full-Screen Mode



A new issue that relates to running Q&A for DOS in full-screen mode has cropped up. As some folks are still unaware that you can run Q&A for DOS in one of two ways, here's a refresher.

Any well-behaved DOS program can be run under Windows 95 or later either as a windowed application—in a moveable and sizeable window where the Windows taskbar is still visible at the bottom of the screen—or in full-screen mode, when all you can see is the DOS application, just as though the program was running on a DOS-only PC.

To switch between these two modes, you press Alt-Enter. You can do this at any time, even while Q&A is performing a Mass Update or other task.

Up to now, I've said that this window vs. full-screen mode is a purely personal preference issue. Some people prefer full-screen. Others (including me) prefer to run Q&A in a window. But just last week I encountered a problem on a client's new **Sony Vaio** notebook PC that makes this preference issue purely academic.

On this PC, when Q&A is running full-screen, there are all

sorts of problems. Rather than go into details or symptoms, let me just say that you will see there is something wrong as soon as you try to use Q&A Write. The cursor flashes near the middle of the screen, not at the left margin where the text appears as you type! You'll see this cursor blinking in the wrong place in all operations.

I don't know the reason for this. It could be a video driver issue. There might be a fix for it—maybe a simple one like downloading an updated display driver.

It's notable too that this PC also suffers the "No Expanded memory" malady that's been mentioned time and again in these pages. (Q&A for DOS 4.0 can use Expanded memory. Q&A 5.0 for DOS demands it for all but the simplest operations.) But to me the solution is simpler: Don't run Q&A for DOS in full-screen mode if you encounter this phenomenon on your PC. Run Q&A in a window, and then you'll be able to select the size of the window (by changing the font), switch readily to other running programs, and copy and paste text between Q&A and Windows programs.

Alec Mulvey

Subsequent programming up to the block that starts with *If CallbackTemp = "Next weekday" Then...* is self-explanatory.

The "Next weekday..." block checks to see if the current date is a Friday, Saturday or Sunday, and returns the following Monday (as the scheduled date) if it is.

The next block of code handles the numeric intervals such as 5 days, 10 days, 14 days, and so forth. It converts the interim **CallbackTemp** value to a number (stripping out the "days" part of the value) and adds it to @Date (the current date) to get the future date.

Incidentally, it doesn't matter what order you type the items in the Userselect statement as they'll be sorted alphabetically (by ASCII value, actually) on the displayed list. The thing to remember is that if you have a single digit selection such as "5," and then a few two-digit selections such as "10" and "20," you'll need to put a space between the comma and the "5" in that particular Userselect parameter. Without it, the default Userselect sort will put the "5" selection after the "10" and "20" because "5" comes after them in the ASCII table.

The final two blocks of programming warn you if you've entered a callback date that's more than 30 days in the future or earlier than the current date. Again, you can modify these parameters to suit your needs.

Callback lists

A handy way to quickly find out who needs a callback today is to design a **Callback report** that prints to the screen. (You can always press F2 from the screen to print the list if you need to take it with you or have a hardcopy on your desk.)

When designing the report, at the Retrieve Spec, type { @Date } in the **CallbackDate** field. Then, at the Column/Sort Spec, specify the columns you want in the report, such as **Name**, **Company**, and **Phone**. When you get to the Print Options screen, set the report to Print to Screen.

If you want to see scheduled callbacks for today *and* tomorrow, use { @Date }; { @Date + 1 } (today *or*

tomorrow) as your Retrieve Spec parameter. In this case, also be sure to make the **CallbackDate** field **1,AS** at the Column Sort Spec so today's callbacks will appear ahead of tomorrow's in the report output.

The scheduler allows you to schedule callbacks for Saturdays and Sundays. The danger here is that if you don't run your callback report on those days, you might miss some scheduled calls. To prevent this from happening, you can use a Retrieve Spec that includes any scheduled weekend calls when you run the report on the following Monday. Here's a compound report retrieval expression that takes care of this for you:

```
{ @Text(@Dow$(@Date) = "Monday", @Date) };  
{ @Text(@Dow$(@Date) = "Monday", @Date-1) };  
{ @Text(@Dow$(@Date) = "Monday", @Date-2) }
```

The semicolon between each expression means *or*. So if it's Monday, you'll get the records scheduled for callbacks during the previous two days (Saturday and Sunday).

I'm sure you can figure out what to change if you need to see Saturday's and Sunday's callbacks when running the report on a Friday.

For a list of *all* future callbacks (including today's), use: >= { @Date } as your Retrieve Spec parameter.

For multiple users

If the database is shared and password protected, each user should have their own unique User ID. You can use that ID so that each person can print a callback report that shows only his or her scheduled callbacks. To do this, you'll need a field in the database that Q&A can fill with the User ID. A simple program like: #100 = @UserID—perhaps as an on-record-exit program with field #100 a Read-only field—will do the trick:

Then, in your callback report, as a *second* Retrieve Spec parameter, you can use { @UserID } in the **User ID** field. This way, each user will get their own custom callback list from just the one report.

Tom Marcellus is editor of *The Quick Answer*.

Quick Tip

Temporarily Disable a Block of Code

You might need to disable a block of programming for debugging purposes, but you don't want to remove it entirely. With Q&A 5.0 you can do this with the REM (Remark) command, but it can be tricky. An easier way, and one that's easier to reverse and also works in Q&A 4.0, is to make that block of programming conditional on a condition that will never be met. First, add line breaks above and below the block of code you want to disable. Then wrap the block in a conditional statement like this:

```
; IF AFieldName = "disabled code" THEN { ..block.. } ;
```

Here's an example of disabling a block of programming that adjusts dates for Y2K-related situations:

```
; IF Firstname = "Programming Disabled" THEN {  
IF @Add AND Date > @Date + 30 Then Date = @Replace(Date,  
@Yr(Date), @Yr(Date)-1) } ;
```

It's a simple matter then to remove the temporary conditional and reinstate the programming with confidence.

Alec Mulvey

of gosubs, but there are so many iterations that Q&A keeps giving me a "too many gosubs" error message. It's a real dilemma. Is there any solution?

Rick

You've hit the wall here. Being familiar with bubble-sort programming, I know that it's practically impossible, with the constraints imposed by Q&A, to do any significant sorting using gosubs. There are just too many iterations and Q&A won't let you perform more than 28 gosubs. Let me suggest the following workaround.

Add a Read-only field to the end of your database named SortRecord. Mass Update the database and put an "X" in this field in all the records that need to be bubble-sorted. Then, program the field with the same programming that starts the bubble-sort routine but with the addition of the initial statement:

```
< SortRecord: If SortRecord = "X" then
{
SortRecord = "";
Rem("Conditions that start the bubble sort here");
```

```
Rem("Goto the field where the bubble-sort starts")
}
```

Now you have to get this to fire. In the Navigation Spec, program the new field this way:

```
< If 999 = "X" then Goto SortRecord else Chome
```

Press F8 to define the on-record entry Navigation field and type "999."

Now all you have to do to bubble-sort your records is to Mass-Update them with "X" in the Read-only SortRecord field and then F10 through them in Search/Update mode. You could even define a macro to press 50 or more F10's for you to reduce your time and effort during the final update.

This is a work-around, but it will help you get the job done with minimal effort.

Bill Halpern is co-owner of Professional Computer Technology Associates (PCTA) in Newtown, Pennsylvania and is President and CEO of Lantica Software, LLC. 215-598-8440. bill@pcta-usa.com, www.pcta-usa.com.

The
**Quick
Answer**
online

Visit *The Quick Answer's* Web Site at
www.quickanswer.com

- Complete Back Issue Index 1990 to date
- Comprehensive Topic Index 1990 to date
- Info on Online Subscription Benefits

- Info on the Q&A successor product
- Q&A Tips and Ideas
- Searchable Solutions Database
- Dozens of Free Files to Download
- Full Database Applications
- Q&A Consultants Directory
- Links to Useful Web Sites & More!

Find out about savings and benefits when you switch to the Online Edition

MARBLE
PUBLICATIONS
1927A Harbor Blvd #389
Costa Mesa, CA 92627 USA

July 2001 Issue

First Class