

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>

Key In Those Keywords —Quick!

JON McLAUGHLIN



Make your Keyword fields work for you, instead of the other way around.

Application in
ONLINE Edition

KEYWORD fields are a powerful Q&A feature. They allow you to enter multiple values in a single field, yet Q&A will handle them individually. They're particularly valuable in reports, where Q&A sorts Keyword field entries individually in a column, the same way it sorts single entry fields. (For more on Keyword fields, see "All About Keyword Fields" in the May 1998 issue.)

The problem is that since Keyword fields often contain multiple entries, typing them in them can be time consuming. Also, inaccurate entries occur when the user fails to separate each keyword with the necessary semicolon. Another problem is caused by entering different descriptions of the same thing in different records, such as "Men" instead of "Males" or "Internet" rather than "Web" or "World Wide Web." Discrepancies like these—plus common data entry typos—produce inaccurate Keyword reports and searches.

The best way to avoid all these potential trouble spots—and to really simplify entering keywords—is to create and enforce the use of a list of predetermined Keyword values.

Basic structure

Let's say you want to use a Keyword field to create a report that shows what your computer customers are using their computers for. Such a Keyword field might contain one or more entries like the following:

Email
Business
Word processing
Accounting
Internet
Games

Using a Lookup Table list, you can turn quick-entry abbreviations into Keyword values separated by semicolons. For example, you can make it so that simply entering "a,e,g" in a field causes Q&A to transform these abbreviations into *Accounting;Email;Games*.

There are several ways to go about this, but I've found the technique I'll show you to work quite well using just a few fields and some inventive @Instr and @Mid programming. This solution allows for fast Keyword field entry and ensures that all entries are accurate. It also makes your list of Keyword field values easily editable in the database's Lookup Table.

You'll need the following four fields in your database:

Field #1 Keyword Abbrv—This is the field where Keyword abbreviations are entered, designed to totally minimize keystrokes. Each abbreviation is a key for the Lookup Table. I've chosen the comma to separate each abbreviation. Any separator character will work, as long as it is never used in an abbreviation.

The Quick Answer

The Do-It-Yourself Guide to Q&A

March 2000

Volume 11, Number 3

- 1 **Key In Those Keywords—Quick**
Jon McLaughlin
- 2 **QuickTip—Prevent Entry of 20th Century Dates**
- 2 **QuickTip—Sort in Last-Month-On-Top Order**
- 4 **Aspects of a Booking System**
Alec Mulvey
- 6 **@Help**
Edited by Bill Halpern & Gordon Meigs
 - Pass Field Values to DOS Text File
 - Easier Date Mass Updates
 - Q&A 4.0 DOS and Large Hard Drives
 - Multi-User Peer-to-Peer Networks
 - Q&A and NT Networks
- 8 **Database Form Design Basics**
Alec Mulvey

MARBLE
PUBLICATIONS

Field #2 Message—This could be an invisible field that, when a lookup to the table fails, displays a clear message telling the user to correct the entries in Field #1.

Field #3 Computer Uses—This is the actual Keyword field, where the full and correct values from the Lookup Table are placed, each separated by the necessary semicolon. It can be a Read-Only field.

Field #4 Count—This field is needed to track entries between commas in Field #1.

Continues on page 3

Quick Tip

Prevent Entry of 20th Century Dates

Even with your database's Global Date Format set to display dates with the full four-digit year, harried users might still revert to old habits and type only the last two digits of the year when entering a date. If they do, Q&A will treat the date as a 1900's date and it might be missed. To prevent the user from continuing without correcting the mistake, you can use a simple program like this in an optionally-filled **Exp Date** field. Adjust the cutoff year to your needs:

```
> If Exp Date <>
  "" Then If
  @Year(Exp Date) <
  1990 Then
  {
  @Msg("Date is
  earlier than 1990.
  Please re-enter");
  Goto Exp Date
  }
```

Quick Tip

Sort in Last-Month-On-Top Order

If you collect data such as sales activity in monthly recap records, it can be useful in reports to have the most recent month's data appear at the top, followed by the data for the previous months in descending chronological order.

One problem with a month sort, however, is that month 01 (January), always sorts on top in an ascending sort (AS) or at the bottom in a descending sort (DS). Another problem is the year change when the period you're analyzing spans two different years.

Fortunately, you can correct for these with a special field and programming that will cause the most recent month to sort on top, and December of the *previous* year to appear on the line *below* January of the current year.

We have a record in our marketing database for each month, and we keep 12 months of history. Our **Month** field is Field #1 and the **Sort** field is Field #2. We use the following programming to auto-fill the Sort field:

```
Month: #1
Sort: #2: #2 = #1 - (@Month(@Date));
      If #2 >= 1 then #2 = #2 - 12
```

In our reports, we sort on the **Sort** field but we make it invisible so it doesn't appear in the report. The Column/Sort Spec looks like this for the two fields in question:

```
Month: 10
Sort: 5,DS,I
```

This way, the month appears in the first (Month) column, sorted by the invisible Sort field.

The sample report below shows the result. This one was run in early February.

Month	Trffc Gross	Trffc Qual	Appl Taken	Lease Appvd	Closing Ratio on Qual	Trffc Qual/Gross	Lease/Qual Appl
02	4	3	1	1	0.33	0.75	1.00
01	60	47	28	9	0.60	0.78	0.32
12	35	19	8	13	0.42	0.54	1.63
11	55	33	8	12	0.24	0.60	1.50
10	15	9	3	11	0.33	0.60	3.67
-----	-----	-----	-----	-----	-----	-----	-----
Total:	169	111	48	46			
Average:	34	22	10	9	0.38	0.65	1.62

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, \$89; Outside the U.S., \$109; Online edition, \$59. 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 © 2000 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

Doug Gardner,
Gardner Companies,
St. Louis, Missouri
DBGIHG@aol.com

Key In Keywords... cont'd from page 1

The programming

First, place all the possible Keyword field values in the Lookup Table (File / Design / Program / Edit Lookup Table), each with an appropriate abbreviation as the Key. (In other words, place the abbreviations in the Key column, and the full Keywords in the corresponding Column 1.) The program for Field #1 is simply this:

```
<#1: @Help(#1)
```

This displays the custom help screen for Field #1 to aid the user in typing the appropriate abbreviations. In designing this help screen, remind the user to place a comma between each abbreviation. The screen should list all the values and their abbreviations found in the Lookup Table. I've found that a list of 18 to 20 abbreviations is normally possible in one custom help screen. The Help Spec (File / Design / Customize / Define Custom Help) needs to be set to *Concurrent* so the user can refer to it until all abbreviations have been entered. This set of abbreviations in conjunction with the help screen creates a quasi "pick-list" format that I've found to be just as quick and convenient as any point-and-click or scrollable pick-list. The only drawback is the need to use multiple custom help screens for a large number of values. By using several fields to place abbreviations in, and stringing these together, this drawback can be easily overcome.

Here's the program for Field #2:

```
<#2: If @Instr(#1,",") > 0 then {#1 = @Del(#1,
@Instr(#1,",",1); Goto #2};
If @Left(#1,1)="/" then #1=@Del(#1,1,1);
If @Right(#1,1) = "," then #1 = @Del(#1,@Len(#1),1);
#4 = 1; #2 = ""; #3 = ""; Goto #3
```

The @Del programming in this field deletes all unnecessary commas accidentally entered in Field #1. (The lookup programming in Field #3 will reject all other invalid characters.) This program also sets (or resets) the counter used in Field #4 to track commas in Field #1.

The programming for Field #3 is in two basic parts. First, part 1:

```
<#3: If @Instr(@Mid(#1,#4,100),",") = 0 then {#2=@Lookup(
@Mid(#1,#4,100),1);
If #2="" then {#2= "" + @Mid(#1,#4,100) +"" " + "Is An
Invalid Entry - Please Correct";
#3 = ""; #4 = ""; Goto #1} else{#3 = #3 + #2; #1 = "," +
#1 + ","; #4 = 1; Goto #4 }};
```

Now, part 2:

```
#2 = @Lookup(@Mid(#1, #4, @Instr(@Mid(#1, #4, 100),",") -
1), 1);
If #2 = "" then {#2 = "" + @Mid(#1, #4, @Instr(@Mid(#1,
#4, 100), ",", 1) - 1)
+ "" " + "Is An Invalid Entry - Please Correct";
#3 = ""; #4 = ""; Goto #1} else {#3 = #3 + #2 + ","; #4 =
#4 + @Instr(@Mid(#1, #4, 100), ",", 1); Goto #3
```

This programming loops. Each loop through, a new

Lookup is performed on each consecutive abbreviation, from left to right. I wanted to do this without deleting abbreviations in Field #1 as the programming progressed. This way, you won't have to start over from scratch if you type an invalid abbreviation.

The first part of the program recognizes when it's time for the loop to stop. More on this in a moment. First, let's look at the second part. Field #2 has set Field #4 equal to the number 1. Therefore, the first Key value for the Lookup Table is the characters from the beginning of Field #1 to the first comma. If the lookup fails, (that is, if #2 = "") then the program displays the invalid abbreviation in the *Please Correct* message, then returns you to Field #1 to make the correction. Otherwise, the correct Lookup value is added to Field #3 and followed by the necessary semicolon keyword field separator.

What follows next is the crux of the programming. If the first pass is good, this expression: #4 = #4 + @Instr(@Mid(#1,#4,100),","); Goto #3 gives Field #4 the integer position of the *next* comma, then loops back to the top of Field #3 for another pass. Now, a Lookup can be performed on the *next* abbreviation. The loop back takes us to the first basic part of the programming. It begins with a check to see if there are any commas left, which would be the case for the last abbreviation entered or if only one abbreviation was entered. If this is true, the lookup is performed. If the lookup fails, the programming displays this invalid abbreviation in the *Please Correct* message. Otherwise, the lookup value is added to Field #3. Since this is the last Lookup, no semicolon is added. Finally, with all lookups completed, a comma is placed at the beginning and end of Field #1, in order to accurately check for any duplicate abbreviations. This check is accomplished in Field #4, also in two basic parts. Part 1:

```
<# 4: If @Instr(@Mid(#1, #4, 100),",") = 0 then{#1 =
@Del(#1,
1, 1); #1=@Del(#1, @Len(#1), 1); #2 = ""; Goto #1};
```

Part 2

```
#2 = (@Mid(#1, #4, (@Instr(@Mid(#1, #4 + 1, 100),",")
+1)));
If @Instr(@Mid(#1, #4 + 1, 100), #2) > 0 then{#2 = "" +
@Mid(#1, #4 + 1, (@Instr(@Mid(#1, #4 + 1, 100), ",", -1))
+ "" " + "Is A Duplicate Entry - Please Correct";
#1=@Del(#1, 1, 1); #1=@Del(#1, @Len(#1),1); #3=""; goto
#1};
#4 = #4 + (@Instr(@Mid(#1, #4, 100), ",", 1) + 1); Goto #4
```

Field #4 checks for duplicate entries in Field #1. Like Field #3, it also is a loop. Let's look at Part 2 first. Part 2 first extracts the first abbreviation, including the comma before and after it, and places this data in Field #2. In order to be accurate, the abbreviation is "nested" in commas. For example, without this safeguard, the abbreviation "a" would return a match in an abbreviation like "ab." The leading and trailing commas avoid this ambiguity. Next comes this statement:

Concludes on page 11

Aspects of a Booking System

ALEC MULVEY



The Quick Answer

How Q&A is used at a summer camp to track just about everything from advance reservations to who gets to ride the horses

DOS

```
>If w1d3 <> "" Then w1d3 = "X"
```

BOOKING systems can be time-consuming to create. The devil is in the detail. I don't propose to describe all the elements in an entire booking system, but I'd like to show you some techniques you might find useful in applications involving grids of dates and the like.

The application I'll use to illustrate was designed for a school summer camp that runs for just a few weeks each year. The people running the camp needed a way to easily record advance booking details, as well as print invoices and produce various reports—all performed by a relatively untrained staff.

Because the range of dates that can be booked is known in advance and is confined to a few weeks, I created a database form in a standard grid format, like a calendar. This is the most user-friendly way to show booked dates. To aid data entry for the keyboard-only operator, I used programming that converted any character entered to an "X" for a clear visual indication of the booked dates. (See Figure 1.)

Each "cell" in the booking grid is five characters wide, with a two-character text field defined by the < > field delimiter characters within it. At the Field Names Spec, I named the week 1 fields **W1D1** (for week 1, day 1), **W1D2**, and so forth. The Total field in the right-most column is named **WEEK1**. Naming fields this way isn't essential, but it makes programming easier and lets you use Q&A's @Sum function to sum groups of fields by name.

To simplify input, I permitted the entry of any character into the grid using the following programming in each grid field:

This program tells Q&A to replace any character entered with an "X" but otherwise leave the field empty.

Using non-numerical entries (in this case X's) in the grid looks good, but makes totals harder to compute. I used the following clumsy but effective technique to get weekly totals:

```
#119: Week1 = @len(w1d1) + @len(w1d2) + @len(w1d3) +
@len(w1d4) + @len(w1d5)+ @len(w1d6) + @len(w1d7) ;
If Week1 = 0 then clear(Week1)
```

Using Yes/No fields for the grid would have allowed for simpler programming for the totals (for example, Week1= @Sum(W1D1..W1D7) but I preferred the visually more appealing X's.

Of course, some folks prefer to use the mouse, and this is in fact the kind of application where pointing and clicking would prove a very easy way to fill the pertinent grid fields. I'll briefly describe how you can do this.

Use a Temp Macro to Speed Repetitive Programming Tasks

Macros can help you construct applications like this. Whenever you need to, say, assign field names to a succession of fields or specify navigation programming, consider recording a quick temporary macro to do it for you. An example would be to assign the field names W1D1, W1D2, and so on to successive fields. I would record a macro that closes the field editor, tabs to the next field, opens the field editor, clears the existing name, types "w1d" and then stops. You then type the next number and activate the macro again with the keyboard shortcut. The disincentive to using macros in this way might be the fear of creating lots of redundant macros that use up key identifiers (hot-key combinations) and lay in wait ready to ambush you if accidentally activated, or the difficulty of remembering which macro does what. I solve this by always using the key identifier Ctrl-Q, and naming the macro "DeleteMe." This is, after all, a temporary macro to be overwritten with the next one in 10 minutes time, and to be deleted whenever found.

Copy	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Σ
WEEK 1					X	X		2
WEEK 2		X	X	X	X	X		5
WEEK 3		X	X	X	X	X		5
N/A								

Figure 1. The Booking page

The fields are set up in the same way. However, at the Navigation Spec you need programming that causes the cursor to “skip over” all the booking fields. This programming is placed in the last field before the booking grid and instructs Q&A on leaving that field to go to the field beyond the grid. This prevents the user from unintentionally making bookings.

The programming in each field now becomes on-field-entry:

```
< If W1D1 = "" Then W1D1 = "Y" Else W1D1 = ""
```

This way, the status of the field is toggled. Click on the date to register a booking, click again to cancel it. The downside of this is that the total does not automatically recalculate with the changes made. You need to change another field or press F8 for the weekly totals to recalculate, regardless of the Calc Mode.

Other pages in the database form are used to record contact and personal information, financial details of the booking, and so forth.

Another interesting feature of this application is that the birthday of the camp attendee is shown in the Birthday field only if it occurs during the person’s stay. (I describe how to do this in my article, “The Dating Game,” in the April 1998 issue of *The Quick Answer*.)

Invoicing

Invoices are printed as mail merge documents. The financial page of the database is fairly straightforward and totals all the extras. (See Figure 2.) One feature worth mentioning is the invoice number. There is provision for two payments and an outstanding balance. This typically consists of a deposit paid at the time the booking is made, a subsequent payment, and a final payment for extras taken during the camp stay or to adjust for errors. The mail merge document quotes an invoice number which is the student’s account number with a suffix added to indicate the invoice number. This suffix is /1 for the first invoice, /2 for the second, and /3 for the third. These are calculated by Q&A in field named Inv#. The various fields for recording the payments are shown in Figure 2.

Figure 2. The Financial details.

The Invoice number suffix is calculated using the following programming statement:

```
Inv#: #500 = 1 + ( Deposit > 0 ) + ( Payment1 > 0 )
```

The default suffix (before any payment) is 1. Once the deposit is recorded, Deposit is greater than zero. So, *Deposit > 0* is True, or 1. Therefore, *Inv# = 1 + 1 = 2*. Likewise for the next payment.

One day at a time

Let’s have a look at another aspect of the booking system—a database designed specifically for the day camp where the students attend each day but have an option to board or sleep over. (A separate database is used for the full-time students who are invariably from overseas). There was a crucial need to record who attends each day and, for billing purposes, which extra services they take, such as extended day or horse riding. Even more importantly, the ones who stay over need to be recorded at least for safety and security purposes. All of these options can and do change hourly, and several times during the day, the staff (young and unfamiliar with the system) need to print out lists every time Arabella changes her mind and decides that she can, after all, bear to share a room with Sophie again!

The need was for a report that would print a “Sleeping Tonight” list without requiring any setup whatsoever. I achieved this by naming the fields for each day **Day1**, **Day2**, and so forth. (The “Sleep” column shown in Figure 3.)

I filled the database’s Lookup Table with the actual dates in the format YYYY/MM/DD, which is Q&A’s internal date format. (See Figure 4 on page 11.)

The purpose of all this is so that a Retrieve Spec can be created to find all those students who are sleeping that night, regardless of the date. The following expression in any field in the Retrieve Spec accomplishes the task:

```
{@(@Lookup(@Date, 1)) = "X"}
```

Concludes on page 11

WEEK 2	Attend	Sleep	Early	Late	Trans
SUN 25 Jul					
MON 26 Jul	X	X			
TUE 27 Jul	X	X			
WED 28 Jul	X	X			
Thu 29 Jul					
Fri 30 Jul					
Sat 31 Jul					

Figure 3. One of three Attendance and Options pages.

@Help

EDITED BY GORDON MEIGS
AND WILLIAM HALPERN



Passing Field Values to a DOS Text File

I am trying to pass database values to an ASCII file so I can then use the file as variable input later in other fields in Q&A 5.0 databases. As I understand it, the following program should create a plain ASCII text file named `c:\will.nos` containing nothing but the contents of field #2000:

```
@Shell("echo "+ #2000 +" >c:\will.nos")
```

When this program executes, however, the resulting `will.nos` file contains the value in field #2000 *plus* a backslash (\) or the text: `"\COMMAND,"` depending on which of two computers I use. This creates all sorts of problems when using that ASCII file for an action later on. I've enquired about this before. Any news? Is it something to do with a Windows setting?

Daniel Kirsh

Using `@Shell` with the DOS Echo command and redirection symbol ("`>`") is a great way to pass the contents of a field to a DOS text (ASCII) file for use elsewhere in Q&A or in an external program. First, you need a field to hold the "return value" ("`0`") produced by successful completion of the `@Shell` command. Let's call this field #99. Next, you need a field whose content you want to pass to the ASCII file. Let's call that one #10. Finally, let's call the ASCII file, `C:\Junk.txt`. Here's an entire sample programming statement:

```
>#10: #99 = @Shell("Echo " + #10  
+ " > C:\Junk.txt"); #99 = ""
```

If field #10 contains 112345, you should wind up with `Junk.txt` containing nothing but 112345. (The program also clears field #99.)

Regarding the unwanted `"\COMMAND"` that will sometimes be mysteriously added to an ASCII file created this way, Tom Marcellus reports that he has experienced this when running Q&A 5.0 under Windows 95 and 98, and that the fix is to simply add a space before the *final* double quote mark in the `@Shell` command, as follows:

```
... " >C:\Junk.txt ")
```

Tom first presented this technique for generating ASCII files on-the-fly in his tip in the November 1997 issue, page 7, where his example does include the space preceding the final double quote mark.

Easier Date Mass Updates

There must be a better way to update dates than the way I've been doing it. When I import data into my database, I need to assign to each new record (usually about 200 at a time) the date that it is entered. I do this with a Mass Update. I have my Program Spec configured to calculate two new dates based on that date. So, next, I display each record (with the empty date fields) and press F8 to perform the calculation. I then press F10 to get the next record, and repeat this 200 times. It's too much work. Is there a way I can use a single Mass Update to perform all these calculations at the same time?

Marlin Penton

There sure is. When setting up your Mass Update, press F8-Options at the Update Spec screen to display the Auto Program Recalc screen. Here, you can choose to execute on-record entry and/or on-record-exit statements, and optionally calculation statements as well during the Mass Update. Remember, Q&A will only execute programming statements that do not start with "`<`" or "`>`" (on-field-entry or on-field-exit) symbols. Remember also that Q&A performs calculations in field number order (field #5 will calculate before field #10) so take this into account when programming your database.

If you've set up your Mass Update correctly (remember to save your Spec so you won't have to re-create it next time), you can have your Update Spec program auto-fill the initial date field, then have the other date fields (whose dates are dependent on the first date) auto-filled via the regular calculation programming in your Program Spec.



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.

Q&A 4.0 for DOS and Large Hard Drives

I use Q&A 4.0 for DOS. Recently, I purchased a new computer with a Celeron processor. I tried to install Q&A from the DOS prompt with the Symantec-supplied disks, but when I insert disk #1 and type install, I get a "divide error" message and can't proceed. I called Symantec, but they no longer support Q&A. Can you tell me how to install Q&A 4.0 on my new computer?

D. Phillips via email

We've said it before, but it's worth repeating: Q&A 4.0 for DOS *will not install* in a hard drive partition that's larger than 2G. The drive simply *must* be partitioned for the install program to work. However, Q&A 4.0 program files *can be copied* to a partition that's larger than 2G from another computer via a network or cable connection, or from a zipped floppy copy of an already installed version. You can then run Q&A using the *Lesspace* utility available for download from virtually all the Q&A-related Web sites. (Without *Lesspace*, running Q&A in a partition greater than 2G is just asking for data corruption.)

Q&A 5.0 for DOS *will* install to a partition that's greater than 2G, as will Q&A for Windows. However, both versions still need to work with data and temporary files on a partition that's less than 2G to avoid corrupting the data. With Q&A 5.0 you can use *Lesspace*, but with Q&A 4.0 for Windows, you have no choice but to run it in a 2G or smaller partition. Partition Magic from PowerQuest is a great tool for creating a small hard drive partition without destroying any existing data on the drive.

Multi-User on Peer-to-Peer Networks

We recently purchased a new Windows 98 computer for our office. We've been networked for years using LANtastic and Q&A 5.0 for DOS with no problems. But now, with the new computer set up as the server, the other computers cannot access Q&A (a constant beeping occurs) until I exit completely out of Q&A. The other computers can access all databases simultaneously as long as the Windows 98 (server) is not using Q&A. If I set one of the older Windows 3.11 computers as the server, all computers can access Q&A. The only problem is that the Windows 98 computer causes the server to get error messages, usually relating to index problems. With the older computer set as the server and only the older computers accessing the Q&A database, we don't have any problems. I called Artisoft and they suggested this is a DOS Share problem. Have you come across this situation? Is there a cure?

Kevin Swad

Artisoft (the company that makes LANtastic) is correct. You are now using Microsoft peer-to-peer networking and don't have DOS Share running. With the older computers and LANtastic, you actually used DOS Share as part of the setup. But you don't need Share to solve your current problem. Instead, you must set the Sharing Mode of *every*

database to *Allow*. (From the Main menu, select File / Design / Secure a File / Declare Sharing Mode, change the Sharing Mode to *Allow* and press F10 to save the change.) This way, all the workstations will be able to share the files no matter where they are run from. As for the beeping, we haven't seen this. Visit www.quickanswer.com and make sure you have Q&A set up properly to run on your Windows 98 PC. (Read "How to Set Up Q&A on a New PC.") Index corruption problems on newer systems are the result of running Q&A in too large a hard drive partition. Download and start using the *Lesspace* utility and these problems should also disappear.

NT Networks and Q&A

How can I get Q&A for DOS (4.0 or 5.0) to run on my company's new NT network? I keep getting lockups and network busy messages.

Joy Dee

We all know that NT is not all that it's advertised to be. It is extremely finicky about hardware, file sharing, timing, memory, network access, and so on. We have been struggling with these problems for months and have come up with what we feel is the best set of guidelines to date. These rules will not guarantee that you will have no problems, but our experience is that they have allowed us to run in many NT installations successfully. Remember, they only apply to DOS versions. (Q&A for Windows will *not* run in NT.)

- One of the major issues seems to be when a person leaves Q&A open in the background, on their task bar, and has the properties of the shortcut set to "always suspend." Make sure to uncheck this checkbox in the Q&A Shortcut properties on the Miscellaneous tab. Make sure people close the database if they will not be using it for long periods of time.
- Another major problem seems to occur when trying to XPost in Q&A 5.0. (When one user gets hung up, others do not necessarily lock up.) The real problem is that the server seems to lock the file and will not allow it to be accessed again until either the server is rebooted or the offending user is logged off *from the server*. Make sure that all of your database files are healthy and are Generation One. (See previous tips and articles on this topic. You can search The Quick Answer back issue indexes at www.quickanswer.com to find out which issues they're in.)
- Use only one networking protocol (for example, TCP/IP).
- All network cards should be the same speed (10Mbps or 100Mbps, but not mixed).

Concludes on page 12

Database Form Design Basics

ALEC MULVEY

It's easy to overlook the simple but significant elements when designing a new database.

DOS

THIS month I'd like to cover some of the basics of Q&A for DOS 4.0 and 5.0 database design—in particular, choosing the fields you'll need and how to place them on your database form.

You know how to design a Q&A database, right? You select *File / Design a file / Design a new file* or *Redesign a file*, type your field names on the screen, add a colon (:) at the end of each one, then press F10 to save the design. That's about all there is to it. One of the attractions of Q&A is the simplicity of both designing a database and subsequently modifying it.

To illustrate some database design principles, I'll use as an example a typical name-and-address database to store contact details on customers, patients, friends or whatever. Hopefully, you'll pick up a few tips you can use immediately to make your database more eye-pleasing, easier to use, and more functional.

At the database design screen

Let's dive straight in and create a new database called *People.dtf*. For the names, you'll need at least two fields—let's call them *Firstname* and *Surname*. (You can use *Lastname* if you prefer.) Type them on the screen and add the colon at the end of each one. Whatever you type here becomes the field name. It will show as the column heading in reports and be used as the field code in merge labels or merge documents. Because of this, you don't want to use field names that are very long, especially for fields that contain relatively short values, as that would cause problems in your reports. You can include spaces in your field names, and capitalization doesn't matter, so if you prefer to use *First Name*, then that's okay, too. The *First name* field can contain initials if that's all you have.

Now some address fields. For U.S. readers, two address fields, along with fields for the city, state and ZIP code normally suffice. Elsewhere, or for multi-national databases, these aren't nearly enough. Here in the U.K., I suggest five address lines plus Post Code (equivalent of the ZIP Code), plus a field for the country, like this:

U.S.	Add 1:	U.K.	Add 1:
	Add 2:		Add 2:
	City:		Add 3:
	State:		Town:
	Zip:		County:
			Post Code:
			Country:

If you're confident that you're only going to store U.S. contacts, then stick with the U.S. arrangement. But, if you think you might want to store details on even a few foreign contacts, then I suggest you use the "U.K." framework as shown above.

It doesn't matter if some fields remain empty. In fact, it's unavoidable. For example, a short address might leave the Add 2 and Add 3 fields blank, but that won't present a problem at all. In a mail merge, Q&A will automatically suppress any blank lines caused by empty fields.

It's always smart to have a dedicated *Country* field. If the person is in your country, just leave the field blank. This way, you don't need a field such as *Foreign?* or *Overseas?* to signify people in other countries. Contacts in your own country can be isolated by using the simple retrieval parameter */=* (slash-equals) in the *Country* field.

The Format Spec

When you save or resave a database design, Q&A displays the Format Spec. Q&A treats all fields as plain text fields by default. It's tempting to change these to *TU* (Text Upper Case or all caps) or *TI* (Text Initial Caps), but think twice before you do so. It is generally accepted that text in all caps is harder to read (it's certainly more tiring to read) and is like shouting. Though *TI* may work for first names, there are certain surnames that will cause problems—names such as McDonald, O'Shea or Hamilton-Smith will wind up as *Mcdonald*, *O'shea* and *Hamilton-smith*. Take it from me—people with such names take great exception to having them "spelled incorrectly!" Likewise for company names. You'll wind up with *Coca-cola*, *Bbc*, *Ibm* and *Hewlett-packard*, and this is not how you want to address a letter to someone at one of these companies. Even address lines are vulnerable. With an address field formatted *TI*, you'll get entries like *Po Box 7161* and *Sw 8th St*. Rather than use *Ti* for some fields and leave users confused about when they need the shift key, I just ignore *Ti* altogether.

State and Country are obviously not candidates for *TI* formatting. They are, however, candidates for *TU*—all caps (as with the Post Code field for UK and Canada) as these are always shown in capitals. If yours is a U.S. / International database, make your ZIP Code field *TU* and at least 10 characters long so you'll have enough room for any ZIP + 4 extended ZIP Codes. The field length will then be suitable for British and Canadian postal codes and those of most other countries.

Title and salutation fields

The fields you've added to your database might seem adequate for storing people's names—but then again they might *not* be. It's not unusual when writing to someone to

address the letter to *Mr Paul Revere* and begin the letter: *Dear Mr Revere*. (In the U.S. one usually adds a period after *Mr.* or *Ms.* or *Mrs.*) So don't fall into the trap of using one field for the whole name. We've all received mailings where the letter starts "*Dear Mr Alec Mulvey*" (well, I have!) and it looks *dreadful*. You need a *Title* field to contain *Mr* or *Mr.* for a man, which is easily enough done—but then you're immediately on dangerous ground. These days it is not acceptable—and quite rightly so—to ask a woman's marital status to find out whether to address her as *Miss* or *Mrs.* Many people persist though, and risk alienating customers in the process. You might like to use just *Ms* unless the status is volunteered by the person on a form or in a letter. (Using *Ms*, you still risk alienating some women who don't like being referred to by this modernistic creation.) Or, you can do what I do—I just omit it altogether. So, it's *Mr Paul Revere* for the man, but just *Lucy Smith* for the woman.

Accordingly, in a mail merge envelope template, the first line will look like this:

Title* *First Name* *Surname

And the letter will start like this:

Dear ***Title* *Surname***

(This obviously presents a problem with "*Lucy Smith*" who will be addressed as just "*Smith*"—but read on.)

These three fields are all you need when writing to strangers. But they aren't really adequate for writing to your best customer. It's too formal. What's required in addition is a *Salutation* field. This would store *Mr (or Mr.) Revere* for a relative stranger, or *Paul* (or some nickname) if he were well known to you. Using programming, such a field can be automatically filled with the title and surname, but will store whatever you change it to. Programming, however, is beyond the scope of this article.

(You may need still more fields if you have many contacts who are academics, in government, the judiciary, the diplomatic service or members of the nobility or royalty; especially if a *couple* needs to be addressed. I have several clients where this is relevant.)

The salutation now becomes:

Dear ***Salutation***

The remaining problem is if you are addressing a woman of high rank, where it might be too informal to start a letter with "*Dear Lucy*." In this case, phone her secretary and ask how she prefers to be addressed.

Other prime fields

Generally, you will have a field named *Company*, which is self-explanatory. Of course, you will need a telephone number field which should be formatted for Text, not Numbers. This way, you can add spaces, hyphens, and additional text such as (Ext. 235) or (evenings) or cell

phone. Make sure the field is long enough for these notations. These days, one phone number field is not enough. Have at least two such fields, plus fax and mobile (cell phone) fields—or a third telephone field. Name these fields Tel 1, Tel 2, Fax, and so forth. Keep the names brief.

You should have an email address field at least 40 characters wide. You might also want a Web address field named *Web*. These fields are the only examples I can think of where *TL* (Text Lowercase) is the appropriate format, as this is the convention for email addresses and URL's (Web addresses). Always add the *www* where necessary, as you can have web sites that are *not* part of the World-Wide Web. You can safely omit the *http://*, however.

You might like to store the job title of your business contacts. You could name such a field *Job Title* or *Position*. Once again, though, this can be information that isn't readily forthcoming. Some people rejoice in their job titles, but these days titles are becoming increasingly irrelevant and forcing their use can be seen as old-fashioned. However, a CEO or Managing Director would surely expect to have his or her status on correspondence.

Field layout

The worst possible database design is to have all the fields placed right under each other down the left margin. There's no need for this. You can place several fields on one line, and lay them out on the form to be more visually appealing. I never start on the extreme left, but indent fields at least one space and try to leave a margin of at least one character on the right, too. This helps ensure that all the data prints when you press F2 and F10 to print the record while viewing it.

The actual field (the information blank where the data goes) usually starts with a colon, like this:

Add 1:

The field then extends to whichever of the following occurs first: A field termination character ("*>*"), another field label, a line draw character (line or box), the right edge of the screen.

Fields don't have to begin with a colon, though. You can use the ("*<*") field start character, which will appear on the design screen but won't appear in the form during use. Moreover, fields are *not* required to have a label. If need be, you can actually have rows and columns of line-item fields using only the "*<*" and "*>*" characters, with no field labels in view. Fields with no labels are assigned default field names by Q&A, such as *F0001*, *F0002*, and so on. Never use these arcane field names in programming or mail merges. Instead, go to the Field Names spec (*Design / Program / Set Field Names*) and give them appropriate internal field names by which you (and Q&A) can recognize and specify them when needed.

Multi-line fields

Fields in Q&A normally end at the right margin, if not before. But you can also create multi-line fields. This can

only be done using the Field Terminator (">") character. The shape of the field will be irregular if it begins with a colon, and a regular rectangle if it begins with the "<" character. The latter is much easier on the eye when the field contains a lot of text. (See Figure 1.)

Multi-line fields are useful as *comments* fields. Almost every database should contain at least one comments field for general free-form text. From the above rules, you will have already worked out that a multi-line field cannot: (a) have another field after it on the same line and (b) cannot have a box drawn around it.

Figure 2 shows how our database looks so far. Notice the following about the Figure 2 design:

- I've added a heading with a box around it. This costs nothing yet identifies the database and makes the form look better.
- The fields' data areas are aligned. This is much more important than the labels lining up. Users don't look at the labels, they look at the data, and it's far easier to read the data if it lines up. The field labels can be left-aligned with spaces after them (as Add1 and County), or right-aligned (as Postcode). You can place a space between the label and the colon, or not, as you wish.
- You can vary the capitalization of the field labels as desired—it makes no difference to Q&A.

Other fields you didn't know you needed

In my view, our database is still short of several fields I would consider the following fields essential:

Key field—Every database should have a field that uniquely identifies every record in it. Even if you don't feel you need a Key field, add it at the outset anyway. Name it *Key, Unique, ID, Ref* or *Record No.* If you don't want to type a unique identifier in each record, go to the *Customize / Set Initial Values Spec* and type *@Number* in this field, then make it read-only by going to the *Program / Read-only Fields Spec* and typing an **R** in the field. This way, Q&A will automatically give each new record a unique sequential record number. At least it will tell you the order in which the records were added to the database and if any have been deleted.

Date Added—You will likely need to know, at the very minimum, when a record was added to the database. Add a field named *Date Added*, make it

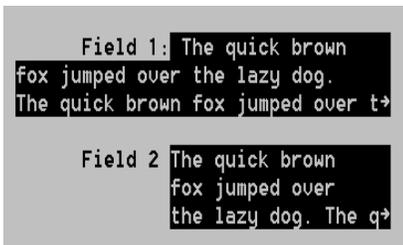


Figure 1. Variable shape of multi-line fields. Field 2 starts with a "<" on the design screen.

a Date field, make it read-only, and give it an initial value of *@Date* at the Initial Values Spec. Ideally, you would also record the *User ID* of the person who added a record, and also the date and User ID when the record was last modified. For some applications, it can be useful to record the time as well as the date. Details on how to do this appear in my article in the February 1997 issue.

Sacrificial Field—As a matter of course, I always add a small field at the top left of first screen (page) of the database. This field, which I name *TopLeft*, can have a variety of uses.

When a user leaves a database open, the cursor is usually in the first field, which is often the most important field containing, for example, the unique record reference number (the *Key* field). *This* is the field that usually gets corrupted when the user inadvertently overtypes it. With the sacrificial field, it doesn't matter.

The field can be used in the future as a means of dialogue between the user and Q&A. A user can get to this field from anywhere in the form with a few presses of the Home key, and type something into it which can instruct Q&A to do something (such as run a macro in Q&A 5.0). Anything else can simply be cleared from the field.

Without this sacrificial field, the *Key* field is likely to be the first field on page 1. As this is a read-only field, users would be unable to press PageUp to move from page 2 to page 1.

Other Text to Add to the Form

It's not just fields I would suggest adding. If your form isn't crowded (which it shouldn't be), then why not use some of the space to add helpful tips for users? This can take the form of simple instructions—but don't overdo it. A safe and effective option is to show reminders of important keystrokes, such as those shown in Figure 3 on the following page.

Conclusion

As you can see, our simple name and address database of just seven original fields has "matured" into one with three name fields, a salutation field, seven address fields,

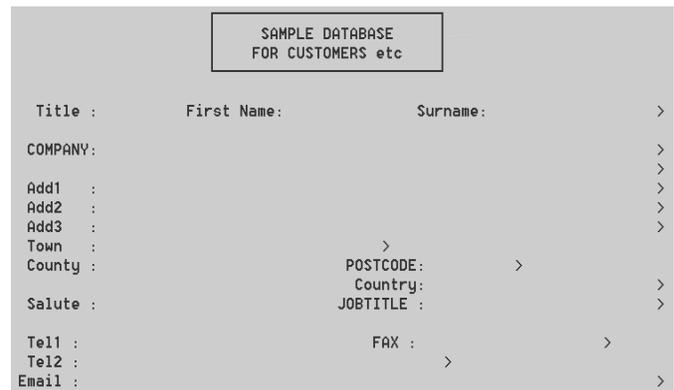


Figure 2. Fields for a basic contact database

Key In Keywords... cont'd from page 3

```
If @Instr(@Mid(#1, #4 + 1, 100), #2) > 0 ...
```

Here, Q&A compares the extracted text string in Field #2 to all the text in Field #1 that is to the right of this text string. A match will return a value greater than zero. When this happens, the programming displays the duplicate abbreviation in the *Please Correct* message. It then deletes the leading and trailing commas in Field #1 and returns the user to Field #1 to make corrections.

If no match is found, the programming moves to this statement:

```
#4 = #4 + (@Instr(@Mid(#1, #4, 100), ",") + 1); Goto #4
```

Similar to what happens in Field #3, Field #4 becomes the integer position of the *next* abbreviation in Field #1, and we loop back for another pass. As in Field #3, the beginning of this loop needs programming to know when the last (or only) abbreviation has been reached. When that occurs, all corrections have been made, so the leading and trailing commas of Field #1 are deleted and we're done!

Good database design ensures that data entry is as consistent and accurate as possible. The technique I've presented here for Keyword fields does that—and more. It also makes data entry fast and easy. And that's what this game is all about, isn't it?

Jon McLaughlin is a professional musician and piano teacher living in Vancouver, Washington. He designs Q&A applications for himself and others. He can be reached at (360) 694-6869 or via email at hbark@juno.com.

Booking System... cont'd from page 5

This expression assumes that any entry in the field is converted to an "X." If this is not the case, then the more general {@(@Lookup(@Date, 1)) <> ""} will do the trick.

Here's how it works. Suppose the current date is 28 July. Q&A's @Date function will then return 28 July. The Lookup Table is constructed so that this equates to Day 11. (The program tells Q&A to lookup the Key column value 28 July and return the corresponding value from Column 1, which is Day11). Indirection is used (the @ sign all by itself) to identify this as the field to be matched with "X." Thus, *Day11* = "X".

The report is then designed in the usual way and run from a macro and custom menu to make the process, as they say in the U.S., a "no-brainer."

Conclusion

Designing a booking system has its challenges. Nevertheless, you can take steps to make such an

KEY	1	2
Date	Day number	
====	=====	
1999/07/18	Day1	
1999/07/19	Day2	
1999/07/20	Day3	
1999/07/21	Day4	
1999/07/22	Day5	
1999/07/23	Day6	
1999/07/24	Day7	
1999/07/25	Day8	
1999/07/26	Day9	
1999/07/27	Day10	
1999/07/28	Day11	
1999/07/29	Day12	
1999/07/30	Day13	
1999/07/31	Day14	
1999/08/01	Day15	

DAYCAMP.DTF Lookup Table

application intuitively easy to use and highly productive, even with untrained users. I hope you'll find some use for the tricks and techniques I've shown.

Figure 4. The dates are the Key values in the Lookup Table

Alec Mulvey owns Keyword Training & Consultancy in Ascot, near London, England, and has been building Q&A applications and training clients for 10 years. Keyword Training is the UK distributor for the International English edition of Q&A and John Dow's Utilities. Fax +44-1344-884-111, alec@keywordtraining.com, www.keywordtraining.com.

Form Design... cont'd from page 10

five telephone number fields, Internet-related fields, and between four and nine additional fields, for a total of 25 fields! That's quite a bit of information to have on a contact, but I think you'll find it will serve you much better now and in the months and years to come.

Alec Mulvey owns Keyword Training & Consultancy in Ascot, near London, England, and has been building Q&A applications and training clients for 10 years. Keyword Training is the UK distributor for the International English edition of Q&A and John Dow's Utilities. Fax +44-1344-884-111, alec@keywordtraining.com, www.keywordtraining.com.

SHIFT F7 - Reset Record

USEFUL KEYSTROKES

Ctrl F5	-- Today's date
F5	-- Copy field value from last record added
Ctrl F6	-- Save & go to Add Data mode
F7	-- Save and go to retrieve spec
Shift F7	-- Undo changes to this record
Home *	-- Start of field / page / record *
End *	-- End of field / page / record *
	* when repeated

Figure 3. Two examples of using screen "real estate" to assist users.

@Helps...cont'd from page 7

- Run Q&A from *local* drives (rather than from the server with the Network Pack).
- Set your NT virtual memory to at least three times the recommended size.
- Set each database's Sharing Mode to *Allow* (instead of *Automatic*).
- Avoid the use of screen savers.
- Share only the data folders rather than the entire drive
- Keep the databases on a FAT rather than an NTFS partition.
- Never perform database Recovery across the network from a workstation. Copy the database to a workstation, perform the Recovery there, then copy the database back to the network drive.

A Y2K Bug For Real!

I've come across a bug in QA 5.01. If you design a Crosstab Report that covers a period of time extending from 1999 to 2000, and use the Date field as, say, the ROW, and @Month for the Group, then if you print the report to the Screen or Preview it, Q&A works out all the data correctly. But try and print it to a printer and Q&A will *not* show any data for JAN00, FEB00, and so on. If you restrict the date range to after 1/1/2000, it works fine. And if it's any other report type, it also seems to work okay. Have you had similar problems? I've downloaded and applied your date fix patches and apart from the above Crosstab report problem, Q&A seems to run fine.

Allan

Yup, you appear to have found a hairy one. Fortunately, the fix is easy. All you have to do is make sure that your ROW Spec reads *ROW, F(D16)* instead of just *ROW*, and your printer will print all the values, even across years. Without telling Q&A to use the long date format in the Spec, it doesn't seem to be able to send the correct format to the printer and the calculations for the summary fields get zeroed.

Bill Halpern and Gordon Meigs own Professional Computer Technology Associates in Newtown, Pennsylvania, 215-598-8440, www.pcta-usa.com, email pcta@comcat.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 Q&A Year-2000 Issues
- 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. PMB 389
Costa Mesa, CA 92627

March 2000 Issue