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

The Monthly Newsletter for Sesame Database Manager
http://www.insidesesame.com
Read a Free Issue of Inside Sesame!

AND

Makers of Sesame Database Manager
Compatible with Symantec Q&A™
http://www.lantica.com

Working Outside the Box with @Shell

ERIKA YOXALL

Lately, more and more Q&A users have been thinking outside the box—that is, how to exploit the wealth of tools out there to extend Q&A's functionality.

SEVERAL recent articles in *The Quick Answer* and some of the presentations at last October's Q&A Masters Seminar and National Q&A Users Group Bash featured examples of how many of us are using other programs such as Microsoft Word to extend Q&A's capabilities. Most of these techniques involve @Shell. But what does the @Shell command really do?

Take a test drive

@Shell is one of the nifty new features included in Q&A 5.0 for DOS. For those of you who haven't tried it, @Shell allows you to run a DOS command or launch another program without having to exit Q&A. When you quit the outside application, you return to Q&A automatically.

For example, the following program starts another copy of Q&A when you enter field #10.

```
<#10:  #10 = @Shell("C:\QADOS50\QA.COM")
```

When you exit the other copy of Q&A, you return to where you were in the original copy.

Being able to launch external programs is nice, but @Shell can do a whole lot more. Create a quick test database called Shell.dtf, and let's look at some of the things you can do with @Shell. Shell.dtf needs the following fields:

```
T1  Text, Program Spec field ID #10
T2  Text, multiline (five lines), Program Spec field ID #20
```

In the following examples, we'll use @Shell to make Q&A interact with Microsoft Word for Windows. That's right, @Shell can even launch Windows 3.1 and Windows 95 programs.

Launching an external program

The following program launches Word for Windows when you enter field #20 (your path to Word might be different):

Windows 95

```
<#20: #20 = @Shell("C:\Progra~1\Micros~1\Office\Winword.exe")
```

Windows 3.1

```
<#20: #20 = @Shell("Win C:\Office\Winword.exe")
```

Remember to always use the eight-character DOS version of any long Windows 95 directory or filename. If you're running Q&A in DOS, calls to Windows programs must be preceded by the *Win* command to start Windows first.

The Quick Answer
The Independent Guide to Q&A Expertise

January 1998
Volume 9, Number 1

1 **Working Outside the Box with @Shell**
Erika Yoxall

2 ***Tip:*** **Cheap Insurance for Your Databases**

4 **Index Reports Made Easy**
Tom Marcellus

7 **@Help**
Edited by T.J. Shuflin
- *Access Documents from a Database*
- *Checking a Field's Contents*
- *Customizing Table View Columns*
- *Move to the End of a Field's Contents*

9 **Refine Your Selection Lists**
Erika Yoxall

10 **Code for Updating Area Codes**
Jeff Nitka

11 ***The Quick Answer*** **Online Advantage**

MARBLE PUBLICATIONS

Adding a startup parameter

You can make Word start and open a particular document by typing a space and the document name after *Winword.exe* in the @Shell command. Or, you can have Q&A pass the document filename parameter to @Shell. If I keep my documents in a directory named C:\Hammer, I can type *Mydoc.doc* into field #10 and use @Shell to open it in Word:

Windows 95

```
<#20: #20 =
Shell("C:\Progra~1\Micors~1\Office\
Winword.exe C:\Hammer\" + #10)
```

Continues on page 3

Catch Us On the Web, Feb. 10!

http://www.QuickAnswer.com

Tom Marcellus

By February 10, *The Quick Answer's* Web site will be up and running. All you need to visit us there is an Internet connection and a browser such as Microsoft Internet Explorer 3.0 (or later) or Netscape Navigator 3.0 (or later).

Lots of things to see and do at the site. You can view a cumulative topical index of *The Quick Answer* that goes all the way back to the June 1990 premier issue. *(See page 4 in this issue to find out how we create it in Q&A.)* You can peruse a complete back issue index as well. You'll find a treasure trove of free utilities and the like to download, a Q&A Tip-of-the-Month, an area for Q&A-related announcements, and even specialty databases you can purchase online.

You can use the Web site to communicate with us, renew your subscription, purchase back issues and specialty databases, and even upgrade your mail subscription to the new **Online Edition** of *The Quick Answer* at no cost. *(See the inside back cover.)* So check us out at **http://www.QuickAnswer.com** after February 10, then drop us an email letting us know what you think of the place.

This month, **Erika Yoxall** shows some of the ways you can *hot-dog* with Q&A 5.0's amazing @Shell command to run DOS commands and interact with other programs—including Windows applications. If you've been looking for ways to extend Q&A's functionality, here's your ticket. In a second article, she shows how to make selection lists even more selective, like when you have a database of items you purchase from various suppliers, and you want a picklist of just the items from a *specific* supplier.

These days the phone companies are splitting area codes faster than you can shake a stick. Although I don't know of a painless way to update affected phone numbers across an entire database, I was struck by the simplicity of **Jeff Nitka's** solution to this thorny problem.

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, $69. Single copy price: $10; outside the U.S., $12.50. All funds must be in U.S. currency. Back issues are available upon request, for the same price as a single copy.

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

Copyright © 1997 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 714-722-9127. Fax 714-722-9127, mailbox@QuickAnswer.com
http://www.QuickAnswer.com

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

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

This publication is intended as a general guide. It covers a highly technical and complex subject and should not be used for making decisions concerning specific products or applications. This publication is sold as is, without warranty of any kind, either express or implied, 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.

How to Reach Us

Phone 800-780-5474 / 714-722-9127
Fax 714-722-9127
Email mailbox@quickanswer.com
Web http://www.quickanswer.com
Mail Marble Publications
The Quick Answer
1927A Harbor Blvd.
Costa Mesa, CA 92627

Quick Tip

Cheap Insurance for Your Databases

DOS

Whenever I create a macro that performs an extensive Mass Update, Post, or other such operation, I precede that section of the macro with commands to make a backup of the affected file(s). Here's an example of the part of the macro that creates the backup. It begins at Q&A's Main menu:

```
FUB<home>Database path and name<enter><home>
C:\QA\BACKUPS\DATABACK<enter>y
```

The *FUB* stands for *File / Utilities / Backup database*, and the macro simply overwrites the previous backup file. This way, if the procedure is run unintentionally, or if something unexpected happens in the middle of it, I can easily restore the database to its previous condition.

Erika Yoxall, Hammer Data Systems

@Shell ... *continued from page 1*

Windows 3.1

```
<#20: #20 = @Shell("Win C:\Office\Winword.exe
C:\Hammer\" + #10)
```

Using the DOS DIR Command

But what if you can't remember the filename of the document to open? I use @Shell to place the contents of my *C:\Hammer* documents directory in a text file Q&A can use to generate a Userselect list. (I'm going to show only the Windows 95 syntax from this point on. The only difference is in the parameter that starts Word.):

```
<#10:
#10 = @Shell("dir /b C:\Hammer > Hammer.txt");
#20 = @Insert("Hammer.txt");
#20 = @Replace(#20," ← Carriage Return after quote
",";");
#20 = @Left(#20,@Len(#20)-1);
#10 = @Userselect(#20)

<#20: #20 =
@Shell("C:\Progra~1\Micros~1\Office\Winword.exe
C:\Hammer\" + #10)
```

The program in field #10 shells out to DOS and prints the contents of the C:\Hammer directory to a file named Hammer.txt. The */b* switch tells DOS to include only the filenames. The @Insert command retrieves Hammer.txt into field #20. Then, the carriage return after each filename is replaced with a semicolon, the trailing semicolon is stripped out, and the contents of #20 now make a handy Userselect picklist. You simply select a document file from the list, press Enter, and the program in field #20 opens Word to the selected document.

With @Shell, always use the "eight-dot-three" DOS filenaming convention to avoid long-filename problems.

Besides launching external programs and running DOS commands (see the sidebar below, and the one at the bottom of the next page), @Shell can do much more—all without having to leave Q&A. So the next time you have a knotty Q&A problem, remember that you can always go "outside the box" with @Shell for a solution.

Erika Yoxall owns Hammer Data Systems in Garrettsville, Ohio, specializing in Q&A, Microsoft Access, and your occasional C program. Phone/Fax 330-527-4018, **hammer@apk.net.**

For Advanced Users Only

In light of John Dow's article ("How Your Database Grows"—September 1997) on what adding fields does to the size of your database, I was searching for a way to streamline the number of "programming-only" fields often required in a database to perform complex calculations. In my case, I was dealing with a check writing routine.

The January 1995 *Quick Answer* includes a wonderful article on designing a check-writing database that converts dollar amounts to their longhand text equivalents for check printing. (For example, turning $23.55 into *Twenty Three and 55/100.*) The technique does the job, but requires 27 Lookup Table entries, 11 programming-only fields, and quite a bit of code. Though I was able to cut it down to just three fields, I didn't want the overhead they would add to the database.

I complained about this to a programmer friend, and he wanted to know why I didn't just write a function to return the longhand text. I started to explain to him how Q&A didn't support that kind of capability, when I realized it actually does! I could write the function in C, compile it as an .exe (executable) file, and have Q&A run it with @Shell.

The C language can work with variables in ways Q&A just can't. The only way to store and manipulate variables in Q&A is to have database fields for them. It works, but once your calculations are complete, you're stuck with fields you wouldn't otherwise need. Using C, you can write a function that picks up a value from Q&A, manipulates it in any way you want, and sends back the result. The function is in an executable file, and you simply use @Shell to execute it.

I replaced the whole kit and caboodle with two programming statements and the one small executable file—Moneytxt.exe, which takes the dollar value entered in an Amount field, and returns its longhand text equivalent. If the Amount field is #10, and the Longhand field is #20, the programming statement looks like this:

```
>#10:
#20=@Shell("c:\qa\moneytxt.exe " + @Str(#10) +
@Txt(@Instr(@Left(#10,3),".")=0,"."));
#20=@Insert("c\qa\temp.txt")
```

The program makes sure that the value in #10 has a decimal point, passes it to Moneytxt.exe, then inserts the text file created by the utility program. That's it! No bulky code, no extra fields, and almost no overhead. All I had to do was place Moneytxt.exe in my C:\QA directory!

Just about any routine that requires "programming-only" fields is a candidate for this type of elegant and efficient solution. It even works in Mass Updates.

Most people don't speak C or happen to have a C compiler laying around. But there are members of the Q&A community, including me, who do. Consider approaching one of us for this type of solution the next time you start adding those "programming-only" fields to your database.

Special thanks to Cow House Productions at http://www.cowhouse.com for C programming assistance.

[You'll find Moneytxt.exe available for download free of charge at ***http://www.QuickAnswer.com*** *when our new Web site opens for business on February 10. —Ed.]*

Index Reports Made Easy

How We Publish *The Quick Answer's* Cumulative Topic Index on the Web

TOM MARCELLUS

Turn a database with a Keyword field into a pre-formatted topic index suitable for publication

DOS

SINCE the early years of *The Quick Answer*, T.J. Shuflin has maintained a Q&A database that serves as a topic index to all the articles, tips and @Help answers (which I'll refer to collectively as *articles*) that have ever appeared in the newsletter. The database contains a record for each article, and includes fields for the title, author, month and year of publication, page number and other information, along with a special Keywords field. (Figure 1 shows a sample record.)

In most cases, the **Major Topic** field contains the title of the article. Below that is the **Keywords** field—and a very useful field it is, as it turns out.

In the sample record, the major topic of Jeff Nitka's article *is Reports from a Multi-Lineitem Database*. The keywords assigned to it are *Report*, *Multi-Line Item*, *Export*, *Import*, *Archive*, and *Macro* because these are the elements (or Q&A features) that substantially contribute to the overall solution presented in the article.

Ultimately, each keyword becomes a topic entry or heading in *The Quick Answer* cumulative online index published on our Web site. And each major topic becomes a reference under that heading.

If you look at Figure 2 on the next page—a portion of a *Quick Answer* cumulative index page on the Web—you'll see that the entries are formatted in two columns with dot leaders.

On this sample Web page, a reference to Jeff's article appears in the index under the *Export* keyword heading. (The arrow points to it.) References to his article also

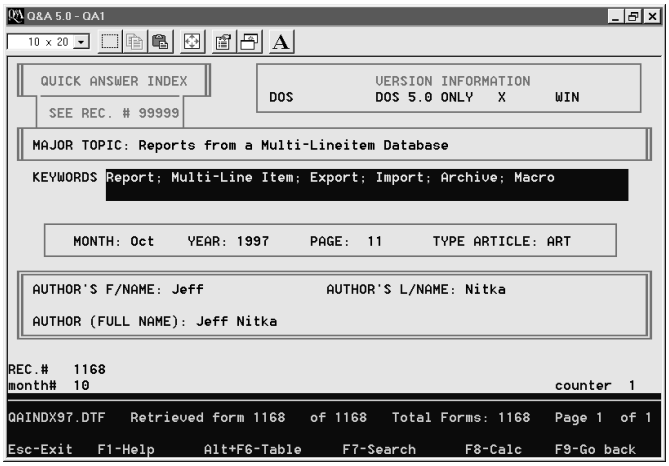


Figure 1. A sample record from The Quick Answer index database. The Major Topic field contains the title, and the Keywords field contains significant features or techniques presented the article.

More Nifty @Shell Tricks

- The December 1997 *Quick Answer* (p. 1) shows how to use @Shell to manipulate and pass Q&A data to Windows programs, then return to Q&A. Use the versatile DOS *Echo* command with @Shell to create an instant export file for Microsoft Excel or Microsoft Word. An example shows how @Shell can run Excel and import the file into a spreadsheet. Another example shows how @Shell can start Word, and run a Word mail merge. It's the way to extend Q&A's functionality by integrating it with Windows applications.

- The October 1997 issue (p. 2) shows how to use @Shell with the DOS *If Exists* command to find out if a file exists before using it in a macro or other procedure.

- The November 1997 issue (p. 7) shows how to use @Shell with the DOS *Echo* >> command to "post" Q&A fields on-the-fly to a document file, appending each "posted" line to the end of the file. Use this trick to track changes to a sensitive field, to give you an ongoing record of who changed the field, when, and what its original value was.

- The July 1997 issue (p. 12) shows how to use @Shell to instantly change any file's attribute to *hidden* or *read-only* (and change it back again)—a trick you can use to make a database or document disappear from the file system or make a document read-only.

—T.M.

appear under the other five keyword headings mentioned earlier.

So, you have what in relational database jargon is called a one-to-many relationship—one keyword to many references or records. It isn't the same as having multiple child records in one database linked to a single parent record in another (as in a true relational structure), but you can think of it in those terms.

That's how the index database is designed. But how do we get from there to the topic index format shown in Figure 2, suitable for publishing on the Web site? I'll take you through the process a step at a time, then show you how we automate it. You'll learn some tricks and techniques you might be able to use to coax custom output like this from your own applications.

Turning a Keyword report into an export file

The cumulative index, as it appears on the Web site, is in two columns. The first column's Keyword is combined with a dot leader (a series of periods), guiding the eye to the second column containing the title, issue date and page number. Because a Keyword field (a special Q&A field type) can't be manipulated in the report derived columns needed for the kind of custom output we want, we start with a Keyword report that we print to disk. Here's the Column/Sort Spec and derived columns for the Keyword report:

```
Major Topic: 20,AS,I
Keywords: 10,AS,K,H(\ ),HS(\ )
Month: 30,I
Year: 40,I
Page: 50,I

Heading: \
Formula: ";" + #20
Column/Sort Spec: 60,HS(\ )

Heading: \
Formula: ";" + #30
Column/Sort Spec: 70,HS(\ )

Heading: \
Formula: ";" + @Right(#40,2)
Column/Sort Spec: 80,HS(\ )

Heading: \
Formula: ";" + #50
Column/Sort Spec: 90,HS(\ )
```

Notice that the Keywords field is the lowest column number (#10), and includes *K* and *AS* codes. These tell Q&A to place the keywords in the report's leftmost column in alphabetical order, and follow each one (in subsequent columns) with the information from all the records in which it appears in the database. A single keyword might occur in just one record, or it might occur in 20, in which case the next keyword in the sort order wouldn't print until the information from all 20 records filled the righthand column(s).

Printing this report to disk (with *Default to Repeating Values* set to Yes) gives us an excellent Standard ASCII file (without column headings) in the following format:

```
Export ;Reports from a Multi-Lineitem Database ;OCT ;97 ;11
```

First is the Keywords field entry, followed by the Major Topic, the month, year, and finally the page number, each separated by a semicolon. Since each record in T.J.'s database (about 1,200 in all) contains an average of five keywords, the export file winds up containing some 6,000 such lines (records). It's too big to display in Write, but we don't need to.

Comes now a supplementary database

Next, we turn around and import this Standard ASCII file (using the semicolon as the field delimiter) into another Q&A database that contains these fields and Merge Spec:

```
Keyword: 1
   NoAt:
  Title: 2
  Month: 3
   Year: 4
   Page: 5
```

We now have one record in the new database for every Keyword occurrence in T.J.'s original database.

Making the index pages

We want Keywords such as @XLookup and XLookup to appear one right after the other under the X index heading. (Normally, terms beginning with @ would sort before any letters of the alphabet.) So we Mass Update the second database using the following Update Spec:

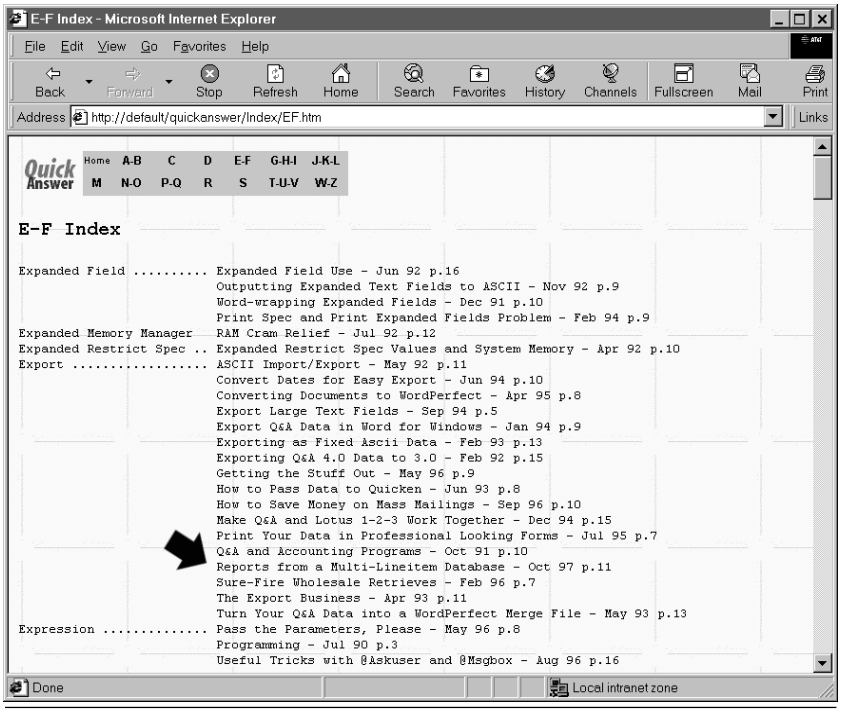


Figure 2. Portion of a Quick Answer index page (E-F) on the World Wide Web.

```
Keyword: #1
   NoAt: #2

If @Left(#1, 1) = "@" Then #2 = @Mid(#1, 2, 25)
Else #2 = #1
```

We can now sort on the NoAt (*no @ sign*) field so that all the @ functions will appear under their appropriate alphabetic headings.

One report for each index group

We have a series of index reports in this new database that are identical except for their Retrieve Specs. Here's the Column/Sort Spec, followed by the Derived Columns and Define Page settings for these reports:

```
Keyword:10,AS,H(20:),I
   NoAt: 5,AS,I
  Topic: 20,AS,H(65:),I
  Month: 30,I
   Year: 40,I
   Page: 50,I

Heading: \
Formula: #10 + " " + @Text(24 -@Len(#10), ".")
Column/Sort Spec: 60,HS(\ ),I

Heading: \
Formula: 2 #60
Column/Sort Spec: 100,HS(\ ),I

Heading: \
Formula: @Text(#60<>#100, #60)
Column/Sort Spec: 70,HS(\ )

Heading: \
Formula: #20 + " - " + #30 + " " + @Right(#40,2)
       + " p." + @Str(#50)
Column/Sort Spec: 90,HS(\ )
```

Define Page settings:

```
Page width.: 100  Page length..: 600
Left margin: 0    Right margin.: 100
Top margin.: 0    Bottom margin: 0
```

The derived columns format the report so that the first column contains the Keyword followed by however many dots (periods) it takes to fill the remaining column width. The derived columns also ensure that the Keyword (and its dot leaders) don't repeat for each reference. The last derived column formats each reference so the major topic is followed by the month, the last two digits of the year, and the page number.

Here's the Retrieve Spec expression for the *EF Index* report (the one for the *E* and *F* keywords.):

```
{@Left(Keyword,1) = "E"
or @Left(Keyword,1) = "F"
or @Left(Keyword, 2) = "@E"
or @Left(Keyword,2) = "@F"}
```

The same basic retrieval expression is used for all the other index reports. Only the letters and index ranges differ. The one exception is the index report that includes special characters, symbols, and the like. The retrieval expression for this one is:

```
{Keyword < "@" or Keyword > "ZZ"}
```

In the ASCII character table that Q&A uses to determine sort order, the @ symbol is one lower than the letter *A*. Accordingly, this expression includes all the characters (as keywords) before @ and after *Z*.

Using these specs, each report is printed to a disk file, winding up in the Web page format shown earlier.

A little cleanup

One final step is needed before the reports are ready to be dropped into their Web pages. We bring each one into Write and resave it to ASCII. This removes the extra space left by the 600-line page length. (These files vary in length, and we don't want page breaks in them.)

Automating it all

Except for adding the records to the original index database (a much-appreciated labor of love on T.J.'s part), the entire process I've describe is fully automated. A macro runs the Keyword report (creating the ASCII export file), removes all the records from the second database, imports the ASCII file into it to populate it with all new records, then runs the Mass Update.

The series of index reports, *AB Index*, *C Index*, *D Index*, *EF Index*, and so forth, are run via macros to create corresponding ASCII text files named *AB.txt*, *C.txt*, *D.txt*, *EF.txt*, and so on.

Here's the macro that prints the EF.txt report to disk, then cleans it up it in Write and resaves it to ASCII:

```
<begdef><nokey><name>"EF<sp>Index"<vidon><esc><esc>rp<capsf4>
d:\qa\data\qaindex.dtf<enter>EF<sp>Index<enter>ni<capsf4>
d:\qa\index\EF.txt<enter>y<esc>wg<capsf4>d:\qa\index\EF.txt
<enter><capsf4><capsf4><ctrlend><ctrlf8><enter><esc><esc><enddef>
```

Except for the report names and file names, the macros (13 of them) are identical. We simply run them one after another—bang, bang, bang. Takes just a minute or two.

Creating the Web pages

The ASCII text files are now ready for import into FrontPage. Our Web page .htm (hypertext markup) filenames correspond to the ASCII text files we created earlier. For example, we have a Web page named *EF.htm* for *EF.txt* (which contains index headings E through F). We open EF.htm, clear the old index, then choose *Insert / File / EF.txt*. FrontPage asks us how we want the file imported. We choose *Convert text to formatted paragraphs*, and FrontPage inserts the file with the preformatted (<pre>) HTML tag to retain its formatting.

We import the other .txt files into their corresponding Web pages the same way (a total of 13 pages in all) and we're done. The entire process takes about 20 minutes, most of which is spent importing the files into the Web pages since FrontPage, unlike our ever-so-humble Q&A, doesn't do macros.

Tom Marcellus is editor of *The Quick Answer*, and host of its Web site (scheduled to appear February 10) at **http://www.QuickAnswer.com**

@Help

EDITED BY T.J. SHUFLIN

Access to Documents from a Database Record

We use a Q&A 5.0 database to maintain client profile information. A client might have several associated Q&A Write documents that we catalog in another database (one record for each document). These records don't contain the actual documents, but simply track them by client name, document tracking number, and filename. When we're on the phone with a client and viewing the profile record, we'd like to be able to quickly display one or more of that client's documents to check information in it. Is there some way this can be done?

Sherry Willamette, Charlotte, North Carolina

FIRST, you'll need to display a selection list of the document filenames for the client whose profile record you're viewing. If you could store each client's associated document filenames right in the client profile database—in a Keyword field, for example—you could use a program like this to display a picklist of documents:

```
< Docs List = @Userselect(Keyword field name)
```

But since you're using a separate database to track documents, you'll need a different approach. I'll assume you want to view the documents in Write (rather than in the client's database record) and that your document tracking database (let's call it **Docs.dtf**) has at least these two fields named **Doc Filename** and **Client Name**

Design a report (name it *Client Document List*) for the Docs database with the following Column/Sort Spec and derived column:

```
Doc Filename: 1, AS, I

Heading: \
Formula No. 1  #1 + ";"
Column/Sort Spec: 2, HS(\ )
```

Leave the Retrieve Spec blank, set the report to print to disk, and set its top, bottom, and left page margins to 0 (zero). These specs will produce a list of document filenames with no column heading, and with each filename followed by a semicolon (for use in creating the Userselect list to come).

Next, add a field named **Docs List** to your client profile database (let's call it **Clients.dtf)**, and program it as follows:

```
< If Docs List = Client Then
  { Docs List = @Insert("Docs.lst");
    Docs List = @Replace(Docs List, "
", "");
    Docs List = @Repllas(Docs List, ";", "");
    Docs List = @Userselect(Docs List);
    If Docs List <> "" Then
    { Docs List = @Shell("QA " + Docs List);
    Clear(Docs List); Chome }
    Else Chome};

  If Docs List = "" Then
  If @Xlookup("Docs", Client, "Client", "Doc
             Filename") <> "" Then
  { Docs List = Client; @Macro("Docs List") }
```

Carriage Return after quote

Here's how the program works. When you move to the Docs List field, it will be empty. This will invoke the last part of the program, copying the client's name to the Docs List field, and running the *Docs List* macro, which copies the client's name to the Clipboard, runs the report you created earlier, then returns to the record.

Now that the Docs List field is no longer empty, the first part of the program executes. It inserts the report (disk file) with the client's document filenames, removes the blank line and final semicolon, then displays the filenames on an @Userselect list. The selected filename is appended to the @Shell command that starts a second copy of Q&A, and the document is displayed in Write.

To return to the client's record, you simply exit the second copy of Q&A. (An Alt-X macro that does the trick appears in the sidebar on page 10 of the September 1997 issue.) When you've returned to the client's record, you can select another document to view simply by moving to the Docs List field again.

Here's the *Docs List* macro:

```
<begdef><nokey><name>"Docs<sp>List"<vidoff><f11><capsf10>
<esc>rpdocs<enter>client<sp>document<sp>list<enter>y<f12><f10>
<f10><f10><enter>docs.lst<enter>y<esc>fsclients<enter><f12>
<f10><dn><enddef>
```

In the macro's last line, replace the <dn> keystroke with whatever keystrokes will move to the Docs List field.

Stumped? Send your Q&A questions to **@Help, The Quick Answer, Marble Publications, Inc., 1927A Harbor Blvd, Costa Mesa, CA 92627** or fax to **714-722-9127**. Please include your name, address, phone number, 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.

Checking a Field's Contents from Another Field

Sometimes, when working in a database record, I need to know if a word (such as "crib") is anywhere in a particular text field in the same record, so I can use that information to decide how to fill another field. But how do I test to see if the text is there? Here's an example:

```
Field A:  The crib can convert to a . . .
```

In Field B, I want to find out if Field A contains the word "crib" and, if so, then I want to be able to make field B contain the same text that's in Field A.

Al Werner, Pittsburgh, PA

YOU can use the @Instr (*in-string*) function to find an occurrence of any word or phrase in a field. You should be able to make use of the following example (or something similar to it) to place the contents of Field A in Field B when Field A contains the word " crib " (with a space before and after the word):

```
If @Instr(Field A, " crib ") then Field B = Field A
```

Without the spaces around the word, words such as *cribbage* (or any word that ends with the letters *crib*) would trigger the program.

Here's a different example:

```
If @Instr(Field A, " crib") then Field B = ""
```

This statement searches Field A for the word " crib" (with a space in front of it) and, if found, deletes the contents of Field B.

Customizing Table View Columns

In Q&A for Windows, how can I limit the number of fields that show in Table View?

Greg Harvey, via email

IN Q&A for Windows, Table View is called a *Spreadsheet Layout*. When viewing a database with lots of fields in a Spreadsheet Layout, there's no way to get rid of some of the columns. You must first design and save an input form that contains only the fields you wish to see. Once you've saved the input form, click on the spreadsheet icon to view the fields you selected. You can drag the column headings to resize the columns or change their order. When you've got the columns how you want them, click on *Records*, then *Save Spreadsheet Layout*. This will preserve your modifications.

If you're repeatedly selecting the same criteria to view records, consider recording a script to save time, and place a script button on the Master Form. First create the script using the Scripting Assistant, and have it set the search criteria and sort order you want. Choose the input form you want to use (the one that shows only a limited number of fields). When the script tests out okay, create a button on the Master Form to run it. This way, when you click on the button, the script will run, and all you'll have to do is click on the Spreadsheet icon when the first record appears.

Table View in Q&A for DOS

In Q&A for DOS, you can display a record, then press Shift-F6 to display the Table View Spec. You can choose which fields to include in the table, and the column order, then save the Spec by pressing Shift-F8. (You can display a list of existing Table View Specs by pressing Alt-F8.)

From the Table View Spec, you can press F10 to display the records in table view. While in table view, you can highlight a record and press F10 to show it in form view. In Q&A for DOS table view, you can see only five columns at a time (you have to scroll to the right to see more), and you can't change column widths.

Move to the End of a Field's Contents

I use a Q&A 5.0 database with a field that I frequently have to update with additional text. The problem is, I have to add any new text to the *end* of what's already in the field. This means that I first have to press F6 to expand the field, then cursor down to the end of the text and press Enter to start a new line. Is there some way that I can make Q&A auto-expand the field when I move to it, and place me on a new line at the end of all the existing text?

Jim Bowling, Columbus, Ohio

SURE you can. All you need is a simple programming statement and short macro. You can place the following program in the Program or Navigation Spec. (Substitute your own field name for *Textfield*.):

```
< If Textfield <> "" then @Macro("End of text")
```

Following is what the *End of text* macro should look like. You can type it directly into your macro file, or record and save it using Q&A's macro recorder:

```
<begdef><nokey><name>"end<sp>of<sp>text"<vidoff>
<f6><end><end><end><enter><enddef>
```

When you move to the field, and it isn't empty, the program launches the *End of text* macro. The macro "presses" F6 to open the Field Editor, then performs three <end> commands to move to the end of the text in the field (no matter how much there is). The final<enter> moves the cursor to a new line, and the macro ends. Be sure to test your new macro on sample data before using it with live data.

T.J. Shuflin is a forensic chemist and crime lab director in Alexandria, Louisiana. He served as the National Q&A User Group's first president in 1991-92, and maintains the group's Web site at http://www.qaug.com.

The Quick Answer

Refine Your Selection Lists

ERIKA YOXALL

The database contains hundreds of items, but you want a picklist of only those from a certain supplier. So what's the problem?

Q&A 5.0's Userselect family of commands is a powerful tool for creating time-saving pop-up user selection lists. (See my article, "Meet the Userselect Family" in the October 1997 *Quick Answer*, for details on these commands.) Sometimes, though, the trick is to get the right values onto the list!

The XUserselectR and @XUserselectR commands let you specify a range of values to include on the selection list. But what do you do if your criteria for telling Q&A which values to include depends on a field *other* than the selection field?

Here's an example. Suppose you have purchase orders database. When you create a new purchase order, you choose the vendor by name from a pop-up selection list. Once you've selected the vendor, you then want a selection list that shows *only* the items you purchase from that vendor. Sounds perfectly reasonable, doesn't it? You want Q&A to scan the items database, isolate the records for *that* vendor, then give you list with just those items. The problem is, you want to give Q&A *two* distinct criteria for selecting records in the external database (not just a range based on one criteria) and the XUserselectR command won't let you do it.

I recently worked on an application that needed this kind of two-criteria capability. Originally, I tried concatenating the first three letters of the vendor name with the item description in a hidden field, and restricting the selection list to the first three letters of the vendor name. This only sort of solved the problem, and it ate up four of my precious 19 characters. (Userselect lists show only the first 19 characters of the selections.) There had to be a better way.

After some experimentation, I came up with a solution. In a nutshell, the technique searches the external database for records that meet your criteria, builds the selection list values into a string in the current record, and uses the string as the parameter for the selection list.

The databases

To see how it works, you'll need two databases, ITEMS and ORDERS.

ITEMS contains information on the items you purchase from vendors. It includes the following fields and specifics:

ItemCode	Stores a unique ID that identifies the item. The values in this field will appear on the item selection list. Text, Speedy/Unique, 19 characters maximum, #10
Description	Text, #20
Unit	Text, #30
Price/Unit	Money, JR, #40
Vendor	Text, Speedy, #50
RecordID	Number, read-only, Speedy, can be hidden, Initial Value = @Number, #60
KeyValue	Text, Speedy, read-only, can be hidden, #70

ITEMS requires only the following on-record exit statement:

```
#70 = @Left(#50,12) + @Str(#60)
```

This statement adds the leftmost 12 characters of Vendor to RecordID—the search string for ORDERS.

ORDERS is a multi-lineitem database for creating purchase orders for vendors. It contains the following fields and specifics:

PurchaseOrderNo	Number, read-only, Initial Value = @Number, #10
Vendor	Text, #20
ItemCode1	Text, #30
Description1	Text, read-only, #40
Quantity1	Number, JR, #45
Unit1	Text, read-only, #50
Price/Unit1	Money, JR, read-only, #60
Total1	Money, JR, read-only, #70
OrderTotal	Money, JR, read-only, #1000
ListValues	Text, read-only, can be hidden, #1010
Countdown	Number, read-only, can be hidden, #1020

You'll need a set of fields (ItemCode, Description, Quantity, Unit, Price/Unit, Total) for each lineitem in your purchase order. For the purpose of testing this technique, no more than three lineitems are necessary. Program ORDERS as follows. Start with this on-record-entry program that sets the Countdown field to the largest RecordID in the ITEMS database.

```
#1020 = @XLR("ITEMS",9999,"RecordID",#1020)
```

If the Vendor field is blank, the next program displays a list of vendors to choose from. Once you select a vendor, the program checks each record in ITEMS. If you purchase the item from the selected vendor, the item code is added to the string in ListValues, followed by a semicolon. When all the items are checked, the trailing semicolon is deleted and you're moved to the first ItemCode field:

Continues on page 12

The Quick Answer

The Program Spec

Code for Updating Area Codes

JEFF NITKA

When it come to database maintenance, how do you cope with new telephone area codes?

DOS WIN

Q&A's programming language provides the common If-Then-Else structure. What Q&A does *not* provide, though, is the CASE structure, commonly used when a program must account for many possibilities for a field's value. Without CASE support, you might think, when having to consider many different field values, that you're stuck with setting up an If-Then for each and every possibility. Fortunately, Q&A's Lookup Table feature provides a way around having to use multiple If-Then statements.

One application of this method involves a situation that has hit many of us during the last few years—the phone company changes or splits one or more local area codes. New area codes are needed to accommodate the increasing demand for pagers, cellular phones, and extra lines for modems in home and office computers.

Recently, my state implemented two new area codes affecting some 250 exchanges, requiring massive updates to computer systems. (The *exchange* is the first three digits of the local phone number.) Using a Mass Update spec, I could have written a separate If-Then-Else statement like this for each affected exchange:

```
if @Len(PhoneNo) = 12
then if  @Left(PhoneNo,3) = "908"
     then if @Mid(PhoneNo,5,3) = "232"
          then PhoneNo = "732" + @Mid(PhoneNo,4,9)
          else
          if @Mid(PhoneNo,5,3) = "233"
          then PhoneNo = "732" + @Mid(PhoneNo,4,9)
          else ...
```

Alternatively, I could have used a lot of OR's, like this:

```
if @Len(PhoneNo) = 12
then if  @Left(PhoneNo,3) = "908"
     then if   @Mid(PhoneNo,5,3) = "232" or
               @Mid(PhoneNo,5,3) = "233" or
               @Mid(PhoneNo,5,3) = "244" or
               @Mid(PhoneNo,5,3) = "258" or
               blah , blah, blah ..
          then PhoneNo = "732" + @Mid(PhoneNo,4,9)
```

Instead, I accomplished the task this way:

1. First, I obtained a list of all the affected exchanges from my local phone company.
2. From this list, I entered all the affected exchanges in the *Key* column of my customer database's lookup table. In column 1 (for each exchange), I entered the number 1.
3. I ran a Mass Update on the database using the following Update Spec:

```
#1: Rem "In this example, the old area code is
908. The new area code is 732. The code only works
on phone numbers in the following format: ###-###-
####, where the dashes are manually typed.";

if @Len(PhoneNo) = 12 then
if @Left(PhoneNo,3) = "908"
then if @Lu(@Mid(PhoneNo,5,3),1) = "1"
    then PhoneNo = "732" + @Mid(PhoneNo,4,9)
    else Rem "Exchange not listed -
              Phone Number remains unchanged."
else Rem  "Area code in phone number does not
           match the area code being changed."
else Rem "Inappropriate format for this update."
```

Considerations

- If you use a field template (###-###-####, for example), the first line of code should read *if @Len(PhoneNo) = 10*, since templates only provide appearance—the template characters themselves aren't stored as part of the value.
- You should make a backup of your database after you enter the exchanges in the lookup table but before running the mass update. This way, if you make a mistake, the backup will have the original data and you won't have to re-enter all the exchanges in the table.
- This routine is optimal for large customer databases. Though entering the exchanges in the lookup table will take some time (the number of affected exchanges in my area code was about 250), it's still more efficient than manual updates. Be careful when entering the exchanges in the lookup table; the update accuracy depends on it.
- If you want to update more than one phone number in your customer record (like a fax number, an alternate number, or the like), copy and paste the above code for each PhoneNo field, replacing every occurrence of PhoneNo with the different phone number field.

Jeff Nitka works for a chemical manufacturer and develops Q&A applications part-time for Epoch Software, 732-874-3989. Jeff is the author of the Program Evaluator (a Q&A program debugging utility), SurfDrve (a Q&A disk drive reader), and FaxMan (a Q&A faxing database), all available from Marble Publications.

The Online Advantage

http://www.QuickAnswer.com

*Upgrade your **Quick Answer** subscription to the **Online Edition**, and we'll toss in a **free** issue!*
Plus you'll save $20 on your next renewal

ON February 10, *The Quick Answer* will open its Web site at **www.QuickAnswer.com**. This will let us deliver **The Quick Answer** "electronically" at *lower subscription rates*.

No, we're *not* discontinuing the regular printed edition of **The Quick Answer**. But we want you to consider upgrading your current subscription to the online edition as soon as possible. And there are some very good reasons to do so.

Our subscription price hasn't increased in five years, even though our two big expenses—printing and mailing—have.

Beginning next month, the renewal rate for the printed **Quick Answer** edition will increase from $79 to $89 per year ($109 outside the U.S.) We'll offer an optional *six-month* subscription for $49 ($69 outside the U.S.).

But the price of the online edition will be $20 less—just $59 per year ($39 for six months) wherever you live!

How the Online Edition adds value to your subscription:

- *Quickly copy programming code and macros from any article, tip or @Help column to the Windows Clipboard, and paste them right into your database or macro file!*
- *Search your **Quick Answers** (like word processor documents) to find occurrences of words and phrases!*
- *The download files for many issues will contain the actual Q&A databases featured in that issue. Use these pre-programmed databases as they are, or tinker with them and modify them to suit your needs!*
- *Post your name and email address, along with a brief description of your Q&A interests or expertise (even hawk your Q&A wares) in **QuickConnection**—an area of the site reserved for online subscribers only. Network with and get help from fellow Q&A enthusiasts around the world!*
- *Fast, easy online access to **The Quick Answer** topical and back issue indexes (see page 4 in this issue) that cover all the articles and tips published in the newsletter since the June 1990 premier issue!*
- *You'll receive other online subscriber benefits, such as free downloadable databases and utilities designed by Q&A developers.*

With the online edition, you'll enjoy the same familiar Quick Answer. There will be no changes to the newsletter's appeareance or content. The online edition will look just like the regular printed edition.

How it will work

New issues of **The Quick Answer** will be posted on our Web site on the 5th of each month *(that's two to three weeks earlier than a regular mail subscription is normally delivered.)* You can visit the site whenever it's convenient, and enter your Subscriber ID (which we'll send you) to download that month's issue. It'll take just a few minutes, and you can then read **The Quick Answer** at your leisure, and optionally print it out on your printer.

What you'll need

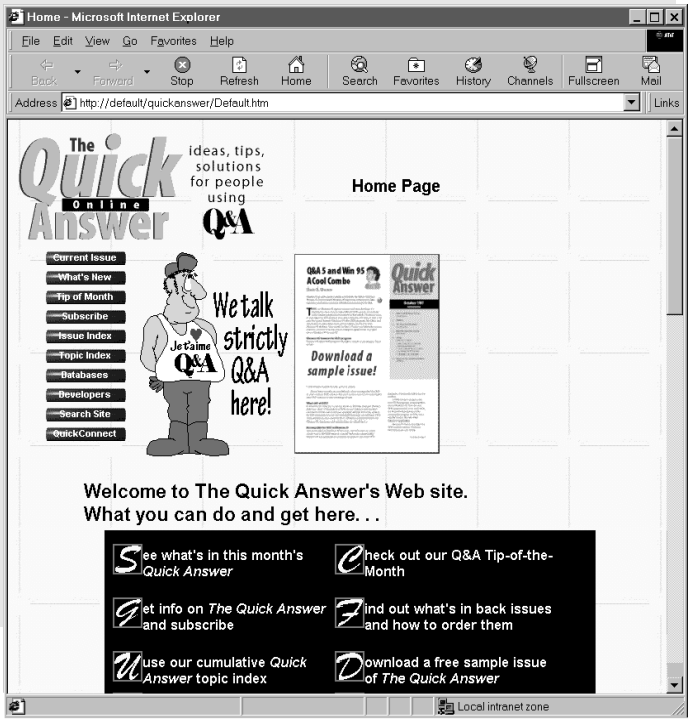
You'll need Windows 95 or Windows 3.1, Internet access, and Microsoft Internet Explorer 3.0 (or later) or Netscape Navigator 3.0 (or later). The online edition will be published in **Adobe Acrobat Reader** *Portable Document Format*. You can download Acrobat Reader free of charge at **http://www.adobe.com** or at our Web site after February 10. Installation is a snap. Once you've downloaded your issue of **The Quick Answer**, simply start Acrobat Reader, then open the issue to read or print.

You can start right away

To upgrade your subscription, send us an email message today at **mailbox@quickanswer.com**. Include your name and address, and say "Upgrade Me" to an online subscription. We'll update your record to extend your subscription by one issue, and send your Subscriber ID number to you by return email.

It's that easy. No more mail delays. No more copies mangled in the postal system. No more retyping programming code and macros. Plus free databases, utilities, and more.

Upgrade to an online subscription today!



Refine Your Lists...Continued from page 9

```
<#20:
If #20 = "" Then #20 = @XUserselect
("ITEMS","Vendor");
#1010 = #1010 + @XLu("ITEMS",@Left(#20,12) +
@Str(#1020),
"KeyValue", "ItemCode");
If Not @Error Then #1010 = #1010 + ";";
#1020 = #1020 - 1;
If #1020 <> 0 Then Goto #20
Else   {#1010 = @Left(#1010, @Len(#1010) - 1);
Goto #30}
```

In the following program, if the ItemCode field is blank, you're offered a list of items based on the ListValues string created when you selected the vendor. When you select an item, the corresponding lineitem fields are filled by XLookup, and you move to the Quantity field:

```
<#30: If #30 = "" Then
{#30 = @Userselect(#1010);
XLu("ITEMS",#30,"ItemCode","Description",#40,"Unit",
#50,"Price/Unit",#60); Goto #45}
```

After you enter a quantity, the Total for the line is calculated:

```
>#45: #70 = #60 * #45
```

All line item Totals are summed to get the OrderTotal. Be sure the @Sum function includes all the lineitem total fields:

```
#1000: #1000 = @Sum(#70)
```

Add a few dozen records to ITEMS for three or so vendors. Then, add a new ORDERS record. When you do, you'll get a list of vendors from ITEMS. Select one, and Q&A should display a list of items from that vendor.

Notes

- Both databases must be set for Automatic Calculation. (Press Shift-F8 from any record in Add Data or Search/Update mode.)
- The number of characters from the Vendor field used in the @Left statements in both databases depends on how many items you expect to have. Q&A will match only on the first 16 characters. I assumed 9999 items or less (four characters), so I used the first 12 characters of the Vendor field.
- For the sake of simplicity, I used ITEMS to build the Vendor Userselect list in ORDERS. You'll probably want a separate database for your Vendor information.
- The time it takes to assemble the items list increases with the number of records in the ITEMS database. My 200MHz Pentium PC scans a 1,124-record ITEMS database in seven seconds. The program has to assemble the list only once per record, though, and you can see its progress on the status line.
- If you format the ListValues field in ORDERS as a Keyword field, you'll then be able to run a simple Keyword search on it to find all the purchase orders that contain a particular ItemCode. All you need is an on-record-exit program that copies each ItemCode in the purchase order to ListValues, and separates each with a semicolon.

I've found this technique to be very useful when a simple range parameter just can't give me what I need.

Erika Yoxall owns Hammer Data Systems in Garrettsville, Ohio, specializing in Q&A and Microsoft Access. Phone/Fax 330-527-4018, hammer@apk.net

MARBLE PUBLICATIONS

1927A Harbor Blvd.
Costa Mesa, CA 92627

Bulk-Rate
U.S. Postage
PAID
Torrance CA
Permit No. 333