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

All About Keyword Fields

ALEC MULVEY

Get relational-like data tracking performance by taking advantage of Q&A's powerful Keyword fields.

WHEN you create a new database, you design your form, then go to the Format Spec where you tell Q&A the type of data you want to store in each field. One of these field types is *Keyword*. I've often found that Q&A users don't know exactly what a Keyword field is for or how useful it can be. A common question is, "What's so special about Keyword fields? They don't seem to do anything different." Well, they do, and I'll show you in what ways!

What is a Keyword field?

In a nutshell, a Keyword field is a text field you can use to store multiple entries. What makes a Keyword field different from a regular text field is that each entry behaves as though it were the only entry in the field.

Q&A is a flat file database, not relational. Nonetheless, Q&A has a number of features, such as XLookup and XPost, designed to provide relational-like performance. Keyword fields is another one of them.

Let's look at a Keyword field in action. I'll use the example that Symantec provides in the documentation because it's a very good one.

Suppose you have a personnel or employee database. Each record stores details on one person. However, a person can have several hobbies—a "one-to-many" relationship in relational database parlance. When you have a "one-to-many" relationship like this, you can use a Keyword field to track the "many" part of it.

Each entry in a Keyword field must be separated by a semicolon. Spaces following or preceding each Keyword entry are optional. For example, in a Hobbies Keyword field, you might have these entries: *golf; tennis; chess*.

So much for entering the data. Now let's see what you can do with it.

Searching Keyword Fields

When searching a Keyword field, you don't need wildcards. Just type something like this into the Hobbies field at the Retrieve Spec:

tennis

This search will give you all the records that contain *tennis* in the Hobbies field, no matter what other hobbies the field might contain.

You can still use wildcards, though. For example, the following Retrieve Spec parameter, with two periods after the partial value, would retrieve all the records with *Wine* and *Windsurfing* in the Hobbies field:

win..

This is only one example of how Keyword fields can outperform regular text fields. It just gets better from here.

Using "OR" in a search

Searching a Keyword field for a couple of alternatives is simplicity itself. For example, the following retrieval parameters in the Hobbies field will turn up the records that include tennis or golf:

tennis; golf

Compare that to what you'd need to perform the same search on a regular text field named Hobbies:

..tennis.. ; ..golf..

Advanced Keyword field searches
AND searches

Keyword fields really shine when you need an AND-type search. Here's a Hobbies field example:

Eighth Q&A User Group Bash and Master Seminar Set for September

TOM MARCELLUS

MAKE plans now to attend the Eighth Annual Q&A User Group Bash and *Quick Answer* Masters Seminar, September 25-27, 1998. This year's three-day gala Q&A event will be held at the Days Inn Premier Convention Center at 1201 K St., Washington, D.C., just three blocks from the Metro and Convention Centers and a block from the White House.

Following tradition, the Q&A Masters Seminar, hosted by *Quick Answer* editor Tom Marcellus, will run from 8:30am to 5:30pm all day Friday (September 25). The National Q&A User Group's Bash will kick off Friday night with a "get aquainted party," and run all day Saturday and Sunday. Here's your perfect opportunity to see the Q&A "gurus" demonstrate a broad range of Q&A tricks and techniques, get all your nagging Q&A questions answered, and participate in a variety of Q&A problem-solving workshops.

For more information on the Friday Masters Seminar, watch for the insert in next month's issue or visit *The Quick Answer's* Web site at www.quickanswer.com. For details on the Bash, including lodging and transportation (for both events), and registration fees, write to Gale Platt at galep@ix.netcom.net (9233 SW 8th St. #423, Boca Raton, FL 33428) or visit the user group's Web site at www.qaug.com.

Birthday Tricks

Application in ONLINE Edition

A client needed to know a person's age in years and months as of the *last* birthday, and also in years as of the *nearest* birthday (to accommodate insurance actuarial tables used to calculate rates.) To find the last birthday in years and months, I used these three fields:

Date of Birth (DOB)	The person's birth date
Age in Years (AgeYr)	The resulting age in years only
Months (AgMo)	The remainder of age in years calculated by month

I then used the following program in the DOB field:

```
>If @Date>= @TD(@Str(@Year(@Date)) + "/" +
@Mid(DOB,6,5))
    Then AgeYr = @Year(@date) - @Year(DoB)
    Else AgeYr = @Year(@date) - @Year(DoB)-1;
If @Month(@Date) < @Month(DoB)
    Then AgMo = 12 + (@Month(@Date) - @Month(DoB));
If @Month(@Date) = @Month(DoB) and @DOM(@Date) <
@DOM(DoB)
    Then AgMo = "11";
If @Month(@Date) = @Month(DoB) and @DOM(@Date)>=
@DOM(DoB)
    Then AgMo="0";
If @Month(@Date) > @Month(DoB)
    Then AgMo = @Month(@Date) - @Month(DoB)
```

If you need to see the result as a single number, add an age field named AgeD to show decimal results. Format it N2 for numbers with two decimal places, then add this to the program:

```
;AgeD =
@TN(@Str(AgeYr)) +
@TN(@Str(AgMo/12))
```

If you need to know a person's age relative to the *nearest* birthday, you can use a statement like this:

```
Age = @Round(((@date-
DOB)*5)/1827,0)
```

Here, I multiply the number of days since the person was born by 5 (to offset leap years) and divide the result by 1,827 (the number of days in any five year period, except for those spanning the odd century) to get the number of years as a decimal result. @Round rounds the result to the nearest whole number.

BILL HALPERN

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

& tennis ; golf

This will find only those people with both hobbies. To perform this kind of search in a regular text field, you'd need to specify your retrieval parameters in a convoluted manner, like this:

..tennis..golf.. ; ..golf..tennis..

And this is with just two hobbies: If you wanted to find persons with three particular hobbies, you'd have to type the three hobbies in a different order *six times*!

NOT searches
For a simple NOT search in a Hobbies field, use the forward slash ("/"), like this:

/ tennis   all hobbies except tennis
/ tennis ; golf   all hobbies except tennis and golf

Note that the NOT operator applies to all entries that follow the slash. (NOT tennis and NOT golf in the last example.)

The last two search examples I'll show use special constructions.

One but not another

& golf ;]/.. tennis ..   golf but not tennis

One entry alone
Finding records with a single entry in a text field is easy. *Golf* in the Retrieve Spec of a regular text field will show only the records where the Hobbies field contains the four characters *g-o-l-f* all by themselves.

In a Keyword field, of course, the same Retrieve Spec will find the records containing golf and other entries. To find just golf, you have a couple of options. At the 1996 Bash in Colorado, I demonstrated the following relatively complex construction for finding the records with golf all by itself in the Hobbies field:

& golf ;]/.. \ ; ..

This is an ampersand followed by golf, a semicolon, a right angle bracket, a forward slash, two periods, a backslash, a semicolon, then two more periods with no spaces anywhere after golf. This construction assumes that there are no semicolons in the target record's Keyword field—that is, no occurrences of *golf ;* in the field. (See the Sidebar, "Compiling a Search Field.")

However, David Flaks showed me an easier way to achieve the same result by using a programming expression in *any* field:

```
{Hobbies = "golf"}
```

Examples of using Keyword fields
Why type your Keyword field entries when you can have Q&A enter them for you via programming? Let's take a look at this option.

Example 1. Compiling a Search field
The classic example of compiling a search field is where you have a number of similar fields—invoice line items, for example—and you want a way to search *all* of them at the same time.

Let me give an example. Suppose you have an orders database containing fields for up to six ordered items, and each line includes a Product Code field. You can create a Search field formatted for Keywords in a convenient location on the form. If the product code fields are named PC1, PC2, and so forth (at the Field Names Spec), you can program your Search field in the following manner.

```
Search :  #130 = PC1+ ";" + PC2+ ";" + PC3+ ";" +
PC4+ ";" + PC5+ ";" +PC6
```

This will place all the product codes from any order in a single field, each code separated from the others by a semicolon. It's a valid Keyword structure, and will allow you to search the field for any combination of products ordered, or run a keyword report, as you'll see later.

This might not be efficient from a programming point of view, though. If your programming execution is set to *Automatic*, the contents of this Search field will be recalculated whenever you change any field on the form. If the record is unlikely to be changed once created, it

Compiling a Search Field

When using a program to fill a Keyword Search field with multiple values, you can wind up with extra semicolons if one or more of the originating fields is blank. The extra semicolons don't do any harm, but you can remove them with a program like this:

```
#1: #1 = @Replace(#1, ";;", ";");
    If @Left(#1, 1) = ";" then #1 =
    @Replfir(#1, ";" , "");
    If @Right(#1, 1) = ";" then #1 =
    @Repllas(#1, ";" , "")
```

The program replaces double semicolons with single ones and removes any leading and trailing semicolons. The program can be used in the Program Spec, but if you find it adversely affects performance, you can always use it—or a similar program—in a periodic Mass Update instead.

won't matter. In any case, there are a number of things you can do to control program execution.

You can have the program execute only when adding a new record:

```
Search :  If @Add then #130 = PC1 + ";" + PC2 +
";" + PC3 + ";" + PC4 + ";" + PC5 + ";" + PC6
```

Another way is to make the statement execute only when the record is saved (on-record-exit.) You do this by pressing F8 at the Program Spec and specifying the field ID number of the on-record-exit field. This has the advantage of faster navigation through the form, as there's only one calculation to perform when you save the record. On-record-exit programming, though, can slow Q&A down when moving between records.

To restrict the recalculation even more, you can use a combination of on-record-exit, on-field-exit, and conditional programming. Here's an example:

```
Search : >#130 : If @Add then  #130 = PC1 + ";"
+PC2 . . . and so forth
```

The problem with this program is that it won't recalculate if an item is later changed or added to the order. To accommodate the possibility in Q&A 5.0, simply change the conditional part of your statement to this:

```
Search :
>#130 : If @Add or @Modified then . . . etc.
```

An on-field-exit (or on-field-entry) statement like this won't execute unless the user navigates through the field in question, so take care!

Just one field to search avoids the nightmarish prospect of having to enter search criteria into each line item field *and* change Q&A's search logic. (Assuming you can remember how to do that!)

For example, suppose the first three line items for a particular sale are these:

```
PC1: VCR206
PC2: RADIO45
PC3: MWAVE88
```

Your program will fill the Search field with the following Keywords:

```
VCR206; RADIO45; MWAVE88
```

This gives you a way to retrieve all orders that include, for example, *both* MWAVE88 and RADIO45.

If the fields being concatenated in the Search field are more free-form—a product description, for example—then you'll probably be performing wildcard searches. In such situations, there might be little or no benefit in making it a Keyword field; it could be a plain old text field.

On the other hand, no penalty accrues in using a Keyword field instead of a text field.

Example 2: Tracking merge letters sent

A Keyword field is an ideal tool for keeping track of merge letters sent to customers or clients. Examples would include a series of introductory letters sent to new clients, or mailings sent periodically to customers or donors to a charity. To track information in this manner, you'd need a way of managing the mail-merge process, and a way to record who was sent which letter.

With the addition of one or two fields, it's easy.

First, add a new field named MailMerges to the database. Make it a Keyword field. Also add a text field labeled Print?, but name it *Print* (without the question mark) at the Field Names Spec.

Before you do the mailing, decide what to call it and write it down. Make the name as short as possible, such as "Spring 98", "New Offer" or the like.

Make sure the Print field is empty in all records, then decide who will receive the letter and mark their records either manually or by performing a Mass Update. You can put any character in the Print field, as long as you're consistent. (Editable Table View is ideal for this.)

When you've marked all the records you want to include in the mailing, create and save a Retrieve Spec that includes them. When automating a task like this, it helps to assign a standardized name to it, such as "Current Print." Then you can do this:

- Run your mail-merge Using your Saved Retrieve Spec.
- Sort by any field you like. (*Always* sort a mail-merge.)
- Mass Update the database using the same Retrieve Spec to do the following:
  - Add the name of the mailing to the MailMerges field. (*MailMerges: #1= "Spring 98; " + #1,* for example.)
  - Clear the Print field.

You can use a macro to automate the entire process, from printing the letters to running the final Mass Update to clear the Print field. When the macro has finished, you'll be able to quickly spot who has received which letter because the MailMerges field in each record will contain a history of letters sent. And you can, of course, make additional use of the field by retrieving lists of customers who've been sent various combinations of letters—that is, some letters but not others.

Keyword Reports

Keyword reports are one of the most-overlooked glories of Q&A. They are simple to create, yet brilliant in the wealth of information they can provide. Consider the following sample records:

Name	Hobbies
Harry	golf; swimming; wine
Ann	swimming; tennis
Marion	chess; wine

To produce a Keyword report on Hobbies, just type

1,K in the hobbies field in the report's Column/Sort Spec, then include any other fields you want. This way, Q&A will sort by the various entries in the Hobbies field, grouping the records by hobby. Such a report might look like the following:

```
Hobbies    Name
--------   ------
Chess      Marion

Golf       Harry

Swimming   Ann
           Harry

Tennis     Ann

Wine       Harry
           Marion
```

Here's this report's Column/Sort Spec:

```
Hobbies:   1,K
Name:      10, AS
```

Really simple, isn't it? Notice that although there are only three records, there are five groups—one for each hobby; and seven rows in the report—one for each hobby for each person. You have to admit that the benefit-to-effort ratio couldn't be much better!

One point worth mentioning is that if you restrict the records in the Retrieve Spec of a Keyword report, you'll still get groupings on ALL entries for the records identified by the Retrieve Spec. Using the same example, if your Retrieve Spec were *swimming; wine*, the resulting report would look like this:

```
Hobbies    Name
--------   ------
Chess      Marion

Golf       Harry

Swimming   Ann
           Harry

Tennis     Ann

Wine       Harry
           Marion
```

This report is identical to the earlier example and includes chess, golf and tennis as well! This is because swimming or wine occurs in the Hobbies field of all three records. When a *record* is included in a Keyword report, Q&A compiles groupings for all the entries for that record. (Bill Halpern shows a way around this in his article, "Tweaking Keyword Reports for the Output you Need" in the December 1996 *Quick Answer*.)

When Keyword fields aren't so compelling
I'd like to be able to say that Keyword reports are the single compelling reason to use Keyword fields—but I can't. The reason is that you can create Keyword reports based on ordinary text fields. Multiple entries are grouped as you would expect as long as the entries are separated by semicolons. (Try it yourself. Create a new report in your own database and just put *1K* in the City field.) On the other hand, Keyword fields obviate the need to use wildcard characters in record searches, and let you easily search using all permutations of and, or, either, not, and so forth.

What you can't do with Keyword fields
Keyword fields are fine for tracking one-to-many relationships where the many side (for example, several hobbies for one person) is a single field. If, however you need more than one field of information on the many side (product code, price, quantity, or the like), then alternative strategies, probably involving two or more linked databases, are required.

Alec Mulvey is a Symantec Approved Consultant and Microsoft Certified Trainer. He runs Keyword Training & Consultancy based in Ascot, near London, England. Fax +44 1344-884-111. alec@keywordtraining.com.

How to Post to a Keyword Field

If you need to *add* an entry to the existing contents of a Keyword field in an external database, you've got a problem because Q&A's XPost command will overwrite the entire target field with the posted value. You'll need a special program like the following to work around the problem:

```
< If @Instr(@XLookup( "Customer", Acct No, "Acct No",
          "Colors" ), Color ) = 0 then {

  XPost("Customer", Acct No, "Acct No",
  @XLookup("Customer", Acct No, "Acct No",
  "Colors") + "; " + Color, "Colors");
  @Msg(Color + " added to record " +
      Acct No + " Colors field.") }

  Else @Msgbox("Colors field in Account " + Acct No,
          "already contains " + Color + ".",
          "Duplicate Keyword not posted.")
```

This sample program conditionally posts a **Color** field's value (*Blue*, for example) in the current record to the **Colors** Keyword field in the target record (on a match between the **Acct No** fields). If the target Keyword field already contains *Blue*, a message advises that the duplicate keyword won't be posted since it's already there. Otherwise, the **Colors** Keyword group is retrieved (XLookup) from the pertinent record, the new color, preceded by a semicolon and space is added to the end of it, and the modified group is then posted back to the Keyword field.

The trick is to place the @XLookup *inside* the XPost command as the "what to post" parameter. It works for Text fields as well. —*Tom Marcellus*

@Help

Exporting Data to Quicken

DOS

I have a client who recently purchased Quicken and needs to transfer data to it from a Q&A database. Where can I find information on how to do this?

Jimmy Hutton, via The Internet

Right here! Quicken can import ASCII text files as long as they're structured in the Quicken Interchange Format (QIF). The easiest way to create a QIF export file is to use a Q&A Write merge document designed for the purpose. The example that follows exports Q&A check records into a Quicken check register where they can be printed.

Here are the suggested Define Page settings for the Write document:

```
Left Margin: 0"     Right Margin: 0"
Top Margin: 0       Bottom Margin: 0
Page Width: 8.3"    Page Length: 10
```

The page length of 10 will create a file with 10 lines per record.

Here are the 10 Quicken check register codes (shown in boldface), along with examples of the text and merge fields to add to the Write document:

D Type Date to be paid here in MM/DD/YY format
T-*Commissions Paid*
N***
P*OWNER NAME*
MPay Phone Commissions
A*ADDRESS*
A*ADDRESS1*
A*CITY,ST,ZIP*
LCOMMIS-OWNER
^

In this example, the minus sign following the **T** on the second line is the amount of the transaction. Since it's a check, it reduces the checking account balance.

Here are what the other Quicken codes signify:

N—The asterisks indicate an unprinted check. Quicken will assign a check number during check printing.
P indicates the Payee
M is for an optional Memo line
A is for the payee's address (up to three lines)
L is for the Category/Transfer/Class

The caret (^) on the last line tells Quicken it's the end of the record.

To transfer the Q&A data to Quicken, print the merge document to disk to a filename such as *Checks.qif*. When the merge document is printed, the "Type Date to be paid here…" prompt reminds you to replace this text with a date in MM/DD/YY format, which will become the date on the Quicken check. At the Retrieve Spec, select the database records you want to transfer.

When the merge is complete, exit Q&A, start Quicken, and import the file. The specific steps to do this depend on the version of Quicken you're using.

You can get more information on how Quicken's transaction files are structured by visiting Intuit's Web site at www.intuit.com/support. Good Luck!

[See also "How to Pass Data to Quicken" in the June 1993 issue. —Ed.]

Q&A 5.0 Date Patch File Missing

5.0

I have Bob Buckland's QP503 date patch utility for Q&A 5.0. The utility makes reference to an accompanying file named QP503PAT.TXT that was not in the ZIP file. Where can I obtain this file?

Paul Smith via The Internet

The German Q&A 5.0 developers changed date formats #7, #9 and #10 to accommodate European date styles. After applying Bob Buckland's QA 5.0 date patch, dates in these formats will appear in Q&A 5.0 in the same format as Q&A 4.0 and the screen samples and manual.

QP503PAT.TXT was from an earlier version of the patch. The latest version (QA503C.ZIP dated 11/20/96) can be downloaded from the *Quick Answer's* web site at *http://www.quickanswer.com*. You'll need PKUNZIP.EXE (or a compatible ZIP decompression utility) to decompress the file. You'll wind up with two files: QP503C.COM and QP503C.TXT. The text file merely says to run QP503C.COM to view your options and to view or print the instructions. Before running the patch, be sure to make a make a backup copy of your QA.DIS file.

If you're using the original single user copy of QA 5.0, place QP503C.COM in the same directory (or "folder") as QA.DIS, and enter *qp503c PATCH* at the DOS

WinClip—Assembling the Ideal Address Block

ALEC MULVEY

WINCLIP, the latest utility from *The Quick Answer*, is an easy-to-use yet tremendously productive add-on that benefits anyone who uses a Windows word processor in conjunction with Q&A for DOS. (WinClip was reviewed by Erika Yoxall in last month's issue.) WinClip copies the contents of a Q&A database field to the Windows 95 Clipboard, and can be configured to fire up a Windows program where you can then paste the data manually or have a macro in that program paste it for you. In this article, I'll confine myself to what you can do in Q&A *before* running WinClip.

What to copy?

What kind of data would you want to copy to a Windows program this way? Well, it could be anything, and to any program—not necessarily just a word processor. The most obvious use, though, is copying a name and address to paste into a mailing label, letter or fax created in a Windows word processor.

"But," I hear you cry. "I thought WinClip can copy only *one* field at a time." True, but there's nothing to stop you from assembling as much information as you like into that one field—even an entire letter or fax compiled by conditional programming right there in the database record. For my example, I'd like to show you how I assemble an ideal mailing label address block.

The "assembly" involves more than just combining the usual list of fields—first name, last name, company, and address fields. These may very well serve in the U.S. where addresses are usualy brief and standardized. In the U.K. and elsewhere, though, addresses are inconsistent and can be relatively long. Here's an actual U.K. address, changed only to protect the innocent:

```
Mr Tom Williams
Partner
Graham Plunger & Co
Unit 1
Westerham Industrial Park
Lower Road
Westerham
Farnborough
Kent
ME47 2JC
```

That's 10 lines *without* the country. On the other hand, a U.K. address might be as concise as this:

```
Mr Tony Blair
10 Downing Street
London
SW1
```

For convenience, I want to use the same Avery Label stock for my mailing labels and the same window envelope. I don't want oversized labels, or risk part of the address not showing in a window envelope. Also, I want to be able to send the name and address to my printer the same way every time, without having to manually edit it to fit just prior to printing.

The way I go about this is to do my "preprocessing" in the Q&A database. I've settled on the following fields:

```
Title     Init      Surname
Position
Company
Add1
Add2
Add3
Town
County
Postcode
Country
```

I also use a Salutation field because I might want to address a letter to the "Right Honourable Anthony Blair," yet in the letter itself refer to him as "Tony."

I tried out various Avery labels, fonts and point sizes, and experimented with window envelopes until I settled on an address block format that would work reliably. This allowed me a total of six lines for everything. You've probably spotted the snag—some addresses might be too long. But this is where the "preprocessing" comes in.

I added this six-line **FullAddress** field to the database and made it read-only:

```
FullAddress <




                                    >
```

The left angle bracket (instead of the usual colon) gives a rectangular field that looks like a mailing label or window on a window envelope.

The purpose of the field is twofold. It assembles the information for mailing labels and letters conveniently into one field, ideal for WinClip. It also acts as a "previewer," so I can see how the name and address will look when printed. Here's the programming in the FullAddress field (copied into this document using WinClip, of course!):

```
#77: FullAddress =
 (Title + " ") + (Init +" ") + Surname + "
" + Company + "
" + Add1 + "
" + Add2 + "
" + Add3 + "
```

```
" + Town + "
" + @Txt(Not (County = "" and Postcode = "" ),County + @Txt
(County <> "","    ") + Postcode + "
") + Country;
FullAddress = @Replace(FullAddress, "  "," ");
FullAddress = @Replace(FullAddress, "

","
");
FullAddress = @Replace(FullAddress, "

","
")
```

The program (set to Auto Recalc) executes whenever the record is modified. If I have to change the address at some later time, I want the FullAddress field to reflect the change immediately, where I can see it.

The first line concatenates the Title (Mr, Miss, or the like), Init (initials or first name) and Surname. The first @Replace command replaces any occurrence of two consecutive spaces with one.

The lines down through the Town line add the data from the pertinent field, followed by a hard carriage return inside quotes. (This is the open quote mark at the end of a line followed by its companion closing quote mark at the beginning of the next line, with a hard carriage return between the two.)

The line after Town inserts the county; if not the county, then the postcode and a hard return, but only if neither are blank *(Not(County = "" and Postcode...)*. If there's both a county and postcode, then the postcode is placed on the same line as the county, separated by a few spaces, the accepted pattern in the U.K. All addresses have a postcode, but some people don't use it. About half my postal addresses include a county.

The final two @Replace commands remove any blank lines caused by empty fields. They can handle up to three consecutive empty fields. If there might be four, you'd need another @Replace command.

This address construction suits my needs very well. I can see how the data I've added to the record will print on a name and address label or letter. And if I don't like how it looks, I can change it then and there while working in the record.

Occasionally, an address has so many lines that it won't fit in the visible (unexpanded) portion of the

If WinClip Won't Work

Even though you've installed WinClip properly, there's a setting in the Windows 95 Properties for your Q&A shortcut that can prevent it from working.

Right-click on your Q&A shortcut, and select Properties from the popup menu. Click on the Program tab, then click on the Advanced button. The checkbox labeled *Prevent MS-DOS-based programs from detecting Windows* must be unchecked (empty). If it's checked, WinClip will create the Winclip.txt file, but won't be able to run the Windows side of the process (copying the data to Windows' Clipboard).

FullAddress field. I can easily spot this by the small right arrow Q&A inserts at the bottom of the field. This lets me know I need to edit the address somehow, either by abbreviating it, or by combining two fields into one, with the elements separated by a comma, like this:

```
Unit 1, Westerham Ind. Pk.
```

I can make the change, and the FullAddress field will immediately show the revised name and address block.

Adding sophistication

If you'd like a more direct warning that an address might be too long, you can add a simple routine like this:

```
<If @Add and Company<>"" and Add1<>""
and Add2<>"" and Add3<>"" and Town<>"" and County<>"" Then
{ @Msgbox("Address is Too Long","","Please Edit");
GoTo Add1}
```

Automating with a macro

WinClip copies the contents of the field where the cursor is. The field must be unexpanded when you press Ctrl-C (or whatever hot key you use) to invoke WinClip. If you're using Q&A 5.0, it's easy to create a macro to move to the target field and run WinClip. Here's how.

Add a new read-only text field named **WinClip** to the page where you enter the name and address information. Make it three characters wide, assign it a color that will make it stand out from the form's background and other fields. Assuming your WinClip-invoking macro is named *WinClip*, enter the following in the Navigation Spec:

```
WinClip: < @Macro("WinClip"); GoTo FullAddress
```

Now, when you click on this field, WinClip will spring into life and copy the address block to Windows' Clipboard, ready for you to paste wherever you need.

Conclusion

Like many businesses, I send an introductory letter with some materials to each new sales contact. The first thing I do is add their details to my Q&A 5.0 database. WinClip, in conjunction with my FullAddress field, makes it a snap to copy a perfectly-formatted address block to the Windows Clipboard, where it goes straight into my ready-made form letter with my company's logo.

Alec Mulvey is a Symantec Approved Consultant and Microsoft Certified Trainer. He runs Keyword Training & Consultancy based in Ascot, near London, England. Alec@KeywordTraining.com, Fax: +44-1344-884-111.

For more on WinClip, visit www.quickanswer.com. In Europe, order WinClip from Alec's Keyword Training (£29.95 + £4.95 p&p + VAT). WinClip is also available from Marble Publications for $49 postpaid. Specify for Q&A 5.0 or Q&A 4.0, and Windows 95 or Windows 3.1

Use Point & Click Automation to Simplify Complex Tasks

ERIKA YOXALL

When you need to automate a strict sequence of steps that require variable user input along the way, a "menu" database can provide an effective solution.

I F you've been reading my articles in past issues of *The Quick Answer*, you've probably noticed that I like to use one-record utility databases designed to perform specialized functions. Well, here's another one!

I use macros to automate complex procedures for clients. Sometimes, though, a procedure requires more than one macro and user input at various stages in such a way that simple <wait> commands in the macro won't suffice. In order to ensure that the necessary macros are run in the correct order and have the user input they need, I employ a one-record "menu" database.

As an example, I'll show you a database I developed—Chkmenu.dtf—to walk a client through his weekly bill-paying procedure. (See Figures 1 and 2.) The process consists of five steps. The client selects which bills he wants to pay based on the vendor's statement or invoice date. He then creates a check for each vendor to be paid, prints the checks, prints a series of reports, and finally posts the check number and date to each paid bill.

Because the procedure is macro-driven, setup involves creating and saving the necessary files and Specs, then recording the macros that run each section of the procedure. You should record each macro starting at the File Menu, and begin it with a Shift-F10 keypress. This is because each macro will have to "press" Shift-F10

to save and exit the menu database record to the File Menu.

Once you have these pieces in place, you can create the menu database that guides the user through each step of the procedure.

Each step in the process requires at least two fields in the menu database:

- A one-character text field defined with < > so it's invisible on the form. This field will be checked off as each step is completed.

- A four-character text field, justified center, set to a contrasting color at the Change Palette Spec, and with an Initial Value of *GO*. This field serves as a *GO* button you click on to launch each step in the procedure.

I also like to include instructions for each step that tell the client what task will be performed, if anything will print, and what kind of paper or forms should be in the printer.

Each page of the menu database should include an invisible one-character text field in the upper left corner for the cursor to "rest" in.

Chkmenu.dtf also needs to include some fields to hold parameters. For example, *Select Bills To Pay* needs a starting and ending date to select which bills to pay, and *Print Checks* needs a Check Date and a Starting Check Number.

The trick to making a menu database like this work is to force the user to perform the steps in the correct order.

Figure 1. Page 1 of the Chkmenu.dtf data entry form. Here, the date range of bills to pay has been specified, the payments have been combined for each vendor (in a separate database), and the user is about to print the checks.

Figure 2. Page 2 of the Chkmenu.dtf data entry form, where reports are printed and payments posted.

For example, you can't let someone launch the *Create Checks* step before selecting which bills to pay. This is where programming comes in. Each time the user clicks a *GO* button associated with a particular step, the program needs to check to make sure that all the previous steps have been performed and that all the required fields for the current step are filled in. If all is well, the current step is checked-off and the macro is launched. Otherwise, a message box appears informing the user what needs to be done first.

Following is the program for the *Print Checks* step (step 3 in the Chkmenu application). The program makes sure the first two steps are done by verifying that they're checked-off. If they are, it then makes sure the current step has *not* been performed. If it has, a message displays. If not, it makes sure that a *Check Date* and *Starting Check* have been entered. If everything is okay, the program checks-off the step and launches the Print Checks macro.

```
<GoPrint:
If Check1 <> "X" Then
{@Msgbox("You must select bills to pay.","","");
Goto Home};

If Check2 <> "X" Then
   {@Msgbox("You must create checks.","","");
   Goto Home };

If Check3 <> "X" Then
   {If CheckDate = "" Then
   {@Msgbox("You must enter a Check Date.","","");
   Goto CheckDate};
   If StartingCheckNumber = "" Then
 {@Msgbox("You must enter a ","Starting Check
Number.","");
   Goto StartingCheckNumber };
   Check3 = "X";
   @Macro("Print Checks")}

Else
   {@Msgbox("This process has already","been
   performed for", "this check run.");
   Goto Home }
```

Each step in the process is controlled by a similar program. You just retrieve the single menu database record, fill in the parameters, and click *GO*. Each step returns you to the same menu record for the next step.

This kind of automation can be applied to most multi-step procedures. My clients like not having to keep track of the order in which they're supposed to perform the steps, and they always know exactly where they left off if distracted in the middle of a procedure.

Notes

- Use on-record-entry programming or Initial Values to set all GO buttons to *GO*.

- Consider using @Color to "dim" the *GO* buttons after a step is complete. For example:

```
If Check3 = "X" Then @Color(GoPrint,8,0)
```

- Use an on-record-entry program like this to prevent records from being added to your menu database:

```
If @Add Then
{ @Msgbox("You cannot add records to this file!",
"This is a one-record interface file.",
"Use Search to Retrieve the record.");
@Exit }
```

- Each macro launched should return to the menu record for the next step. Use on-record-entry programming to determine when the last step is complete, display a message, clear all check and parameter fields, and exit the menu record:

```
If Check5 = "_" Then
{Clear(Check1, Check2, Check3, Check4, Check5,
StartDate, EndDate, StartingCheckNumber,
CheckDate);
@Msgbox("Check run complete!","",""); @Exit}
```

- If possible, it's a good idea to include an easy way to reverse the entire process and start fresh.

- All Check fields should be Read-only. Use Navigation programming to prevent a user from entering a *GO* button field by any means other than clicking on it.

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

Quick Tip No-No Macro Hot-Keys

Never use Alt-I as a macro hot-key. It's too useful in it's default role (Insert Mode). In a macro, it might be the only way to ensure that Q&A is in insert mode. The Insert key merely toggles between insert and overtype modes, whereas Alt-I *always* puts Q&A in insert mode. (Alt-I, Insert *always* puts Q&A in overtype mode.)

You can't use Ctrl-I as a macro hot-key, either, or your macro will run every time you press the Tab key!

ALEC MULVEY

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.

Programming Quirks

JEFF NITKA

THIS month, I'll depart from the usual, and offer a hodgepodge of little Q&A programming quirks and how to get around them.

Quirk #1—Never have a backslash ("\") in any field on which you might search. A search for any record in which a specific field contains a backslash (using a ..\.. search parameter) will result in no records found.

To prevent users from accidentally entering a backslash character, try one of the following programs. The first one replaces backslashes with nothing, and the second replaces them with forward slashes:

```
PO#: > PO# = @Replace(PO#,"\", "")
PO#: > PO# = @Replace(PO#,"\",  "/")
```

Quirk #2—On a selection list generated by the XUserselect command, Q&A 5.0 will display up to 19 characters of the external field. But XLookups are accurate to only 16 characters, and because they're often used on external fields that are also used for XUserselect lists, you should limit the data in these external fields to 16 characters.

To prevent users from entering more than 16 characters into such a field, you should use the Field Template Spec and type 16 @'s. To enforce the restriction with programming (thus adding an extra measure of security), try this in an on-record-exit program:

```
>#100: Account Number = @Left(Account Number,16)
```

Quirk #3—The XUserselect command can have a hard time generating a selection list from a large database that has become even slightly corrupted. (Recovering the external database usually solves the problem.)

If you can't afford for your application to be dependent on the stability of the external database, then it might be best to abandon the XUserselect strategy and, instead, prompt the user for the specific value and have an XLookup verify the existence of it in the external file. (XLookup will work better on large external databases where XUserselect isn't as reliable.) Instead of this:

```
> Name = @XUserselect("Accts","Name")
```

Use this:

```
> If   @XLookup("Accts",Name,"Name","Name") = ""
then { @Msg("Account doesn't exist."); Goto Name }
```

Quirk #4—Suppose a user presses the Esc key when a selection list generated by the @XUserselectR command is displayed. In this case, the value returned by @XUserselectR should be the empty string (""). However, when the "Start" and "End" parameters are the same as the field name that will store the value returned by @XUserselectR (a very common programming strategy), @XUserselectR will mistakenly return the value from the assignment statement (if any) which immediately preceded it. Here's an example:

```
Name:
> Help = "Choose from this list";
Name = @XUserselectR("Accts","ID","ID",Name,Name);
```

Notice that the Name field appears to the left of the assignment operator (=), and is also used as the "Start" and "End" parameters (the last two parameters in the command). In this case, if the user presses Esc, the Name field will contain "Choose from this list," which isn't what you want. To see if the user pressed the Esc key, you can check the value of Name, like this:

```
If Name = "" or Name = Help
Then { @Msg(""); Name = ""; Goto Name }
```

You could also do this:

```
> Help = "Choose from this list.";
Temp = "";
Name = @XUserselectR("Accts","ID","ID",Name,Name);
```

In this case, if the user presses Esc, the Name field will contain the value of the last assignment statement, which is the empty string.

And finally, you could do this:

```
> Help = "Choose from this list.";
Temp = @XUserselectR("Accts","ID","ID",Name,Name);
If   Temp <> "" then { Name = Temp; Temp = "" }
else Goto Name
```

In this case, Temp has replaced Name as the field to store the user's selection. However, unlike the case where Name appeared to the left of the assignment operator (=), Temp will contain the empty string should the user press Esc when the selection list is displayed.

Jeff Nitka develops Q&A and Microsoft Access applications for Epoch Software (E-Mail: 105020.2215@compuserve.com). 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).

prompt in that directory.

If you're running a network copy of QA 5.0 (that is, you've already run the Network Pack), your QA.DIS file won't have the original date/time stamp, and the patch installation just described won't work. In this case, enter *qp503c NOSAFETY* at the DOS prompt. (Bob added the SAFETY check to ensure the utility works only with QA 5.0 and not, for example, on a Q&A 4.0 QA.DIS file.)

Q&A Startup Problems on Network

We converted our Windows for Workgroups network by upgrading one PC to Windows NT Workstation acting as the system server, and eight client Win 95 workstations. We had to reinstall Q&A 4.0 and our databases. Now, we can't get Q&A to run properly. Sometimes it will run fine, but often, when we start Q&A, the workstation will hang, either at Q&A's opening "splash screen," or with a black screen and blinking cursor. We've tried everything, including changing all the Windows and PIF settings. Is there a compatibility problem between Q&A and system we're using?

Ron Stevens, Toronto, Ontario, Canada

Sometimes, we can get so caught up with suspected operating system (OS) incompatibilities that we overlook the basics. Your problem isn't with the OS, but with the setup and configuration of Q&A itself. You're running Q&A from the "server" under a Network Pack license. But because your workstations are older PC's upgraded to new configurations, they're like a bunch of different animals trying to run Q&A the same way. *The problem is with your video startup switches*.

The clue came from the "splash screen" (and blank screen) scenarios you reported. In these cases, Q&A has loaded and is running fine, you just can't see it because the video mode is wrong. The apparently random failures are simply due to the fact that each time a different workstation starts Q&A, Q&A sets the default video mode to *that* PC, and the rest be dammed.

Here are couple of solutions. First, install and run Q&A on each workstation. (Keep only the shared databases on the "server"). This will improve network performance. Q&A should recognize the system during installation, and should thereafter run fine on it. If not, try running Q&A with the command line *QA –SCC* or *QA –SMC*—the two most common video mode startup switches. (A third one, *–SMM,* is for strictly monochrome monitors not seen much these days.)

Bill Halpern and Gordon Meigs own Professional Computer Technology Associates (PCTA) in Newtown, Pennsylvania. PCTA specializes in data management and supporting systems and services. 215-598-8440, 71023.356@compuserve.com

Dated material—Please Rush

Copying Your Q&A Data to Windows Programs Just Got a Whole Lot Easier

Introducing **WinClip** for Q&A

IF you use—*or plan to use*—Windows programs along with Q&A, you need a way to copy data from your Q&A databases to whatever Windows programs you're using. *What does it take now?*
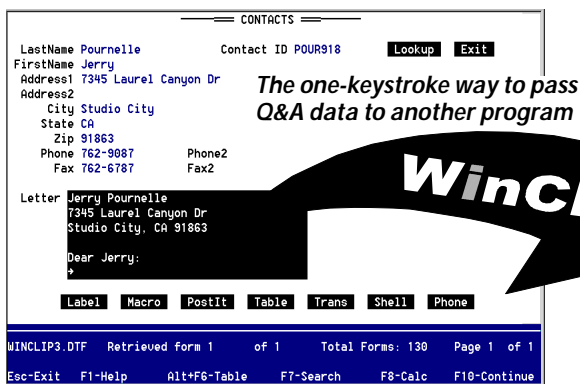
First, you have to put Q&A in a window.

Next, you have to click the Mark icon on the toolbar.

Then you have to carefully drag the mouse over just the data you want to copy, click the Copy icon to copy it to the Windows Clipboard, then press Alt-Enter again to return to full screen.

That's **5 steps**—*and you're still not done*! Now you have to start the other program where you want to *use* the data.

What if you could do all this with just one keystroke? Now you can!

WinClip—a new Q&A add-on from *The Quick Answer*—lets you *instantly* copy any database field to Windows' Clipboard!

You simply press *Ctrl-C* (or the hotkey of your choice) and *BINGO!*—the field's contents (a data value, a name and address, even an *entire multi-page letter*) are in the Windows 95 Clipboard, ready to paste wherever you need!

But that's not all. You can have **WinClip** then start your Windows program so you can immediately paste the data into it (*Ctrl-V* or *Edit / Paste*) and finish the task!

When you quit the external program, you're back in Q&A!

WinClip makes working between Q&A and Windows programs amazingly easy!

The one-keystroke way to pass Q&A data to another program
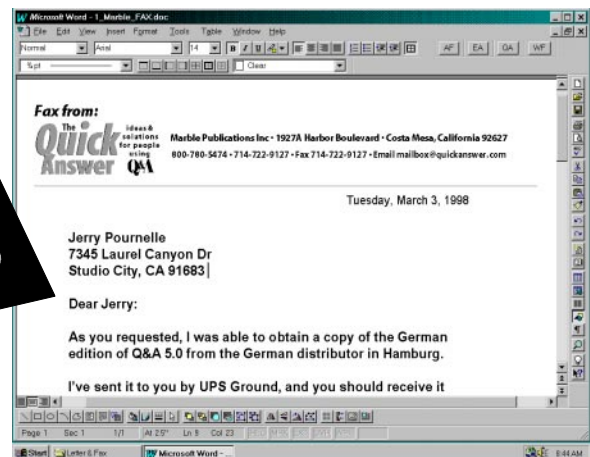
WinClip

- **WinClip** works with **Q&A 4.0** and **Q&A 5.0** running in full screen or window mode in Windows 95. **WinClip** works in any database as-is—it requires no extra fields or programming.

- **WinClip** is not a TSR—it takes no memory away from Q&A.

- **WinClip** handles more data than Q&A 5.0's own clipboard, and lets you effortly pass it to any Windows program or any windowed DOS program—including Q&A itself!

- **WinClip** creates a text file of the data it copies to the Clipboard. You can insert this file in any Q&A database field, display it in Write, or use it in any program that accepts ASCII files.

- **WinClip** retains the formatting of your Q&A data. What you copy is what you paste. (WYCIWYP)

- **WinClip** comes with a demonstration database you can use to try out various WinClip options.

- **WinClip** is a snap to install, and the documentation includes a variety of configuration and usage examples.

Get WinClip today, and find out how incredibly easy it is to share your Q&A data with your other programs!