‘Look Up’ Data In Text Files

Happy Holidays!

Tom Marcellus

DOS

YOU probably know how to use an XLookup to retrieve information from another Q&A database, or a Lookup command to get data from a Lookup Table. But did you know that you can program a Q&A database to lookup information from a plain text or ASCII file?

You can. And though you might not have an immediate need to do so, the technique extends Q&A’s functionality and your capabilities as a developer as well. It might even come in handy some day!

Bypassing the import process

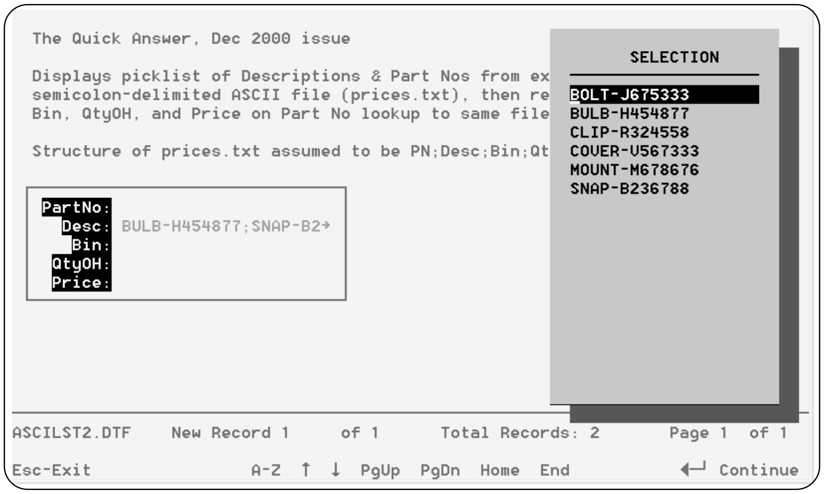
Normally, if you wanted to XLookup information originating in a text file, you’d first have to import the file into a Q&A database and make the Key field (the field that finds the right record) Speedy. This assumes, of course, that the data items—the “fields”—in the text file are separated by a Q&A-supported field delimiter such as a semicolon or commas and quotes, with each record ending in a carriage return.

The point is: you have to *do something* with the text file (import it) *before* you can access the data in it. You can automate the import with a macro. Just be sure you remember to run the macro each time.

But suppose the original file is an updated list of part numbers, descriptions, and prices, as it was in my case. Importing the file into a database is one thing, but having it just add new records in with all the *old* ones, compounds the problem—you have no way of knowing *which* record (new or old) an XLookup will find as a match. To handle that you’d have to first remove all the old records from the database before importing the new ones.

In the application I’ll show you, we were able to avoid all this simply by looking up the information directly from the text file. In this case, the company receives an updated price list (in a text file) by email attachment several times each week They have it set up so that the incoming new file overwrites the old one of the same name. This way, there’s always just the one valid price list file with a known filename.

With this setup, when they enter an order in their Q&A database, the programming first parses (splits up) the prices text file and displays each item’s description and part number on a convenient pick-list. (See Figure 1.) Then, when they select an item, programming finds the matching part number in the same text file and retrieves the rest of the information, such as the bin number, quantity on hand and price. They don’t need or want a separate external database of part numbers. As far as they’re concerned, the regularly updated prices.txt file *is* that database. They don’t work in it; their headquarters office does.



```
The Quick Answer, Dec 2000 issue                       SELECTION
                                                    ______________
Displays picklist of Descriptions & Part Nos from ex BOLT-J675333
semicolon-delimited ASCII file (prices.txt), then re BULB-H454877
Bin, QtyOH, and Price on Part No lookup to same file CLIP-R324558
                                                     COVER-V567333
Structure of prices.txt assumed to be PN;Desc;Bin;Qt MOUNT-M678676
                                                     SNAP-B236788
 PartNo:
   Desc: BULB-H454877;SNAP-B2→
    Bin:
  QtyOH:
  Price:

ASCILST2.DTF   New Record 1    of 1    Total Records: 2    Page 1  of 1
Esc-Exit              A-Z  ↑  ↓  PgUp  PgDn  Home  End        ↵ Continue
```

Figure 1. This database reads in a formatted text file and displays its “key” values on a handy pick-list, then retrieves the rest of the data from the same text file.

Continues on page 3

The Quick Answer

The Do-It-Yourself Guide to Q&A

December 2000

Volume 11, Number 12

MARBLE PUBLICATIONS

Time's Running Out...

Make your arrangements *today* to attend
The INTERNATIONAL Q&A USERS CONFERENCE 2001
January 12-14 in beautiful Irvine, California!

Look into **booking your flight** *right away*. (30 days advance booking usually gets the best domestic rates.)

Call the **Atrium Hotel** in Irvine and reserve your room at the low $80 group rate (800-854-3012 or 949-833-2770).

And go to **www.quickanswer.com/bash5.htm** to register if you haven't yet done so.

We'd **really** love to see you next month at the grandest Q&A gathering ever!

—Tom Marcellus, Your Conference Host

Quick Tip

Staying Out of Trouble With Macros

DOS

Q&A's keystroke recorder macros are a wonderful way to automate repetitive tasks. However, since they are "dumb" macros, meaning that they cannot adjust to unexpected situations, you can get into trouble if your system changes. Here are some guidelines to help stabilize your macros:

- Never hard code full paths into your macros. Always try to use paths relative to the Global paths you set in your Set Global Options screen. When your macro encounters the File Name prompt, press Shift-F4 to clear the existing path, then type in your relative path. This way, if you move your files, as long as you change your Global Options, your macros should still work.

- Know what your Automatic Execution setting is (the Set Global Options screen). If you create macros that expect Automatic Execution to be turned on, they won't work if Automatic Execution is turned off.

- Sometimes, your macro will cause a Yes/No dialog box to appear. For example, when backing up a file, you may get a prompt asking if you want to overwrite the existing file. If there is any possibility that your macro will encounter this, record the correct answer (Y or N) into your macro. Q&A menus never use Y or N for a choice, so this should do no harm if the dialog box does not appear.

- Never use the arrow keys to make selections from lists or dialog boxes. If your macro chooses from a list of reports, type out the report name instead of arrowing down to it. If you use the arrows to select from your list, your macro will break if the order changes.

- Be aware that macros that perform Mass Updates, Posts, and the like cannot do their jobs if someone has the database open. They will, however, continue to pound out their keystrokes, which can have unpredictable and even damaging results. Alec Mulvey demonstrated a technique for checking for this condition in advance in the March 1998 issue ("When to Check for Open Databases").

Stable macros are very important to system integrity. By being aware of how a macro will react to a particular situation, you manage many problems before they occur.

Erika Yoxall, Hammer Data Systems, LLC

The Quick Answer™

Editor / Publisher
Tom Marcellus

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

Cost of domestic subscriptions: 12 issues, $109; Outside the U.S., $129; Online edition, $79. Single copy price: $11; outside the U.S., $12.50. All funds must be in U.S. currency. Back issues are available upon request for the same price as a single copy.

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

Address editorial correspondence, @HELP questions, or requests for special permission to: Marble Publications, Inc., The Quick Answer, 1927A Harbor Blvd., Costa Mesa, CA 92627 Phone 800-780-5474 or 949-722-9127.
Fax 949-722-9127, mailbox@quickanswer.com

On the Web at www.quickanswer.com

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

Reach Us

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

Lookup Text Files...cont'd from page 1

How it works

What do you need to lookup and retrieve data from a plain text file? Not much, really. Just a formatted text file with a unique key value in each row. In this case, it was a part number. Each row in the text file is one record. The only practical limitation is that the text file can't exceed 32K, which is something on the order of 500-800 lines or rows, depending on the average length of the lines. (You can work around this file size limitation. See **Notes** at the end of the article.)

You can adapt the technique to suit most any kind of formatted text file lookup, as long as each row in the file contains a unique key value. (The sample database I'll show along with its associated text file are in the download file for Online Edition subscribers.)

Figure 1 on page 1 shows the sample database containing the five fields. (They're all text fields in this case.) Following is the Q&A 5.0 program (in the **PartNo** and **Desc** fields) that displays the pick-list from the data in the text file, then retrieves the information from it based on the selection made. The programming assumes the prices.txt file is in *part number;description;bin;quantity on hand;price* order, like this:

```
H454877;Bulb;47A;77;12.44
B23678;Snap;44B;102;4.55
R324558;Clip;67A;3244;2.22
etc.
```

PartNo:

```
> If @Add and PartNo = “” Then {

  Price = @Insert(“d:\qa\data\prices.txt”);
  GOSUB Desc;

  PartNo = @Userselect(Desc);
  PartNo = @Mid(PartNo, @Instr(PartNo,“-”)+1, 10);
  Clear(Desc);

  Price = @Insert(“d:\qa\data\prices.txt”);
  QtyOH =  @Instr(Price, PartNo);
  Desc = @Mid(Price, QtyOH, 100);
  Price = @Left(Desc, @Instr(Desc, “
“) -1);
  Price = @Del(Price, 1, @Instr(Price, “;”));
  Desc = @Left(Price, @Instr(Price, “;”) -1);
  Price = @Del(Price, 1, @Instr(Price, “;”));
  Bin = @Left(Price, @Instr(Price, “;”) -1);
  Price = @Del(Price, 1, @Instr(Price, “;”));
  QtyOH = @Left(Price, @Instr(Price, “;”) -1);
  Price = @Mid(Price, @Instr(Price, “;”) + 1, 10);
  }
```

Desc:

```
< If Price = “” Then {
  If @Right(Desc, 1) = “;” Then
  Desc = @Left(Desc, @Len(Desc) -1);
  Clear(Bin);
  RETURN };

  If PN = “” Then {
  Price = @Replfir(Price, “;”, “^”);
  Bin = @Left(Price, @Instr(Price, “;”) -1);
  Bin = @Mid(Bin, @Instr(Bin, “^”) +1, 10) + “-”
  + @Left(Bin, @Instr(Bin, “^”) -1);
  Desc = Desc + Bin + “;”;
  Price = @Mid(Price, @Instr(Price, “
“) + 1, 32000 );
  Goto Desc
 }
```

What the programming does

The program executes on-field-exit from the first field, **PartNo**. It could just as well execute on-field-entry.

The first thing it does is insert the prices.txt file in the **Price** field. The GOSUB then passes control to the **Desc** field where the pick-list will be constructed.

The first block of code in **Desc** executes after the "file" in **Prices** has been successfully parsed to make the pick-list. More on that later.

It's the second block of code in **Desc** that executes the first time through. Initially, the **Prices** field contains the entire semicolon-delimited prices.txt text file—one row per part number. The program replaces the first semicolon (@Replfir) with the caret ("^"). This way, it knows that the part number and description are between the first character in the field and the next semicolon and that they're separated by a "^".

The **Bin** field is then used as a temporary holding place for this first *part number^description* string. The program copies to **Bin** the description first, followed by a dash ("-"), then the part number. So, the first time through, **Bin** will contain something like *Clip-R324558*. This string, prefaced by a semicolon is then appended to whatever is already in the **Desc** field. Since we're constructing the pick-list at this point, the individual items destined for the list must be separated by a semicolon or comma.

Once the new string is appended to the **Desc** field, that row is deleted from the "file" in **Price**, so **Price** now contains the original text file with its initial first row deleted. The statement:

```
Price = @Mid(Price, @Instr(Price, “
“) + 1, 32000 );
```

does this by finding the first carriage return in the file and keeping everything *after* it. (The carriage return after the first quote mark is essential.)

The program in **Desc** now re-executes via the Goto command. (This is called a *loop*.) It continues to build the pick-list in the **Desc** field in the same manner until all the data in **Prices** has been exhausted. The RETURN statement then passes control back to where the GOSUB was called in the originating **PartNo** field.

With the pick-list in **Desc** now completed, the *@Userselect(Desc)* statement displays it. (See Figure 1.)

When you select an item (*description-part number*) from the list, the program parses it, placing just the part number in the **PartNo** field. The price list in **Prices** has been exhausted by the pick-list building operation, so the program re-inserts the prices.txt file.

Concludes on page 12

Q&A Basics

Printing Out Your Data Part 3—Using Print Specs

ALEC MULVEY

DOS

IN Parts 1 and 2 of this series (October and November 2000 issues) I described the five ways of printing from Q&A:

- F2 at a database record
- Mailmerge
- Report
- Mailing Labels
- Print Specs

So far, I've covered all but the last. Of these five printing methods, Print Specs are the least used. And it's a shame, as they certainly have a place and are often a better choice than a report.

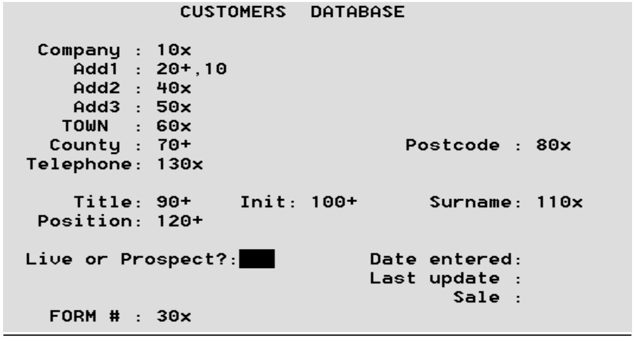
Nature of Print Specs

Print Specs are like a cross between mailing labels and reports. They're like mailing labels in that they print fields under each other rather than in columns and let you print two or three columns of data on a page. They're also like reports in that Q&A stores them in the database, they include saved Retrieve and Sort Specs, and can accommodate headers and footers.

Print Specs should be used when you want to list data that can include fields of variable length (such as addresses), or when you want to precisely position fields on a page, such as on a pre-printed form. They're not so hot when you need to print text other than field contents.

Two types of Print Specs

There are two distinct types of Print Specs—*Free-form* and *Co-ordinate*. Despite this, you use the same *Fields Spec* screen when designing them. The *codes* you enter in the Fields Spec determine the type of Print Spec it is.



```
                 CUSTOMERS  DATABASE

  Company : 10x
     Add1 : 20+,10
     Add2 : 40x
     Add3 : 50x
    TOWN  : 60x
   County : 70+                  Postcode : 80x
 Telephone: 130x

     Title: 90+    Init: 100+      Surname: 110x
  Position: 120+

 Live or Prospect?:███        Date entered:
                              Last update :
                                     Sale :
   FORM # : 30x
```

Figure 1. Fields Spec for simple name and address listing.

When you start to create your first Print Spec (*File / Print / [database name] / Design/Re-design a spec*) Q&A prompts you to name your Spec.

Q&A then takes you to the Retrieve Spec (where you can optionally press F8 for the Sort Spec) and then to the Fields Spec. As you'd expect, the Fields Spec is a copy of the database form and you type codes in the fields you want to include in the print job.

From the Fields Spec, you can press F1 for Help and see that the process is fairly straightforward. You type a number into each field you want included. We'll look first at Free-form Print Specs in detail. After that, I'll point out the differences in Co-ordinate Print Specs.

Free-form Print Spec

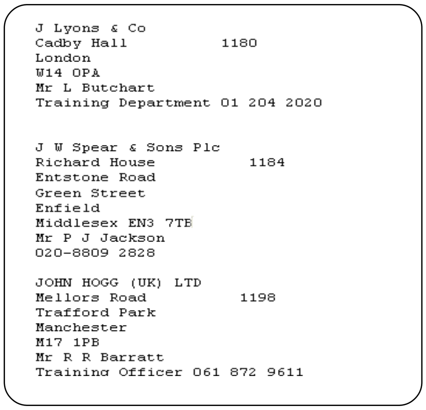
To start, simply type a number into each field you want to print in the order you want them printed. You can number the fields 1, 2, 3, and so forth, but it's smarter to number them by 10s, such as 10, 20, 30 and so on. This way, you can later include another field between two that are already numbered without having to renumber all the fields.

Follow each number with a "+" (plus sign) if you want the *next* field to print on the *same* line.

Follow each number with an "x" (the letter x) if you want the next field to print on the *next* line.

Figure 1 shows the Fields Spec for a simple address listing, and Figure 2 shows the result.

The first code is **Company : 10x.** This puts the company on the first line by itself ("x" means next field goes on a *new* line). The next code is **Add1 : 20+,10**. This means that address line 1 is the second field; the "+" means that the next field (whatever that is) will be on the same



```
J Lyons & Co
Cadby Hall          1180
London
W14 OPA
Mr L Butchart
Training Department 01 204 2020


J W Spear & Sons Plc
Richard House          1184
Entstone Road
Green Street
Enfield
Middlesex EN3 7TB
Mr P J Jackson
020-8809 2828

JOHN HOGG (UK) LTD
Mellors Road          1198
Trafford Park
Manchester
M17 1PB
Mr R R Barratt
Training Officer 061 872 9611
```

Figure 2. Output from the Figure 1 Print Spec.

line; and the ",10" means that there should be 10 spaces before the next field instead of the default one space.

Underneath this is **Add2**, but this has the number **40** in it. So it isn't the next field because **Form #** below it has the number 30 in it—**30x**, to be precise, which tells Q&A to start a new line after this field. The effect of this is that the **Form #** prints on the same line as the first address field, separated by 10 spaces. (See Figure 2.)

From this you can see that creating a Print Spec takes a little getting used to. It's counter-intuitive in a way because you place the codes to position a field in the field that *precedes* it—and that's not the previous field on the database form but the one with the next lowest number!

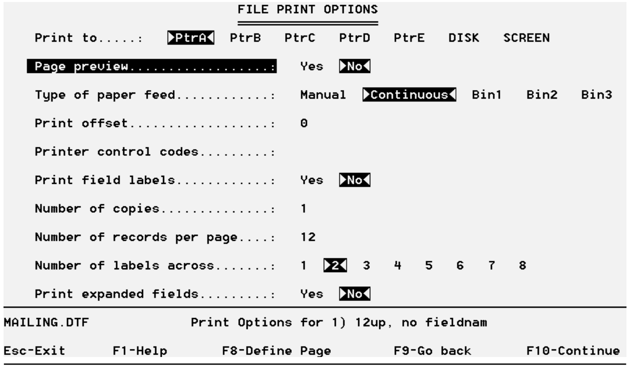
Another example of this is the last line, *Training Department 01 204 2020*, for example. This is comprised of **Position: 120+**, then **Telephone: 130x**. A little tricky, admittedly, but you soon get the hang of it. As in all such exercises, the first one takes an hour of trial-and-error to get right. But then that one is saved. The second one is an adaptation of the first one and so goes much quicker. After that, it takes you only a few minutes.

Once you start using Print Specs, you'll be amazed at just how versatile they are. And you can produce a variety of different layouts of the data on the printed page by altering the settings at the File Print Options screen shown in Figure 3.

The first five choices on the Print Options screen and the *Number of copies* setting should be familiar enough, but a few words should be said about the other options:

Print Field Labels (Yes/No)—If you set this to *Yes*, then Q&A will precede each printed field by its field label. Clearly, this changes the appearance of the output hugely. Sometimes, for example when listing technical information, you'll want field labels shown so users can make sense of the information. At other times you'll want no labels—for example, when listing names and addresses as in the earlier example. (There's also a *third* option—printing field labels on a *selective* basis—more on which in a moment.)

TIP: In these circumstances, Q&A usually prints field *names* (as at the Program / Set Field Names Spec) where they differ from the field labels. Here, though, Q&A is to be taken literally: labels *means* labels—that is, what you see on the database form.



```
                         FILE PRINT OPTIONS

  Print to.....:      [✓ PtrA]  PtrB   PtrC   PtrD   PtrE   DISK   SCREEN

  Page preview..................:    Yes  [✓ No]

  Type of paper feed............:    Manual  [✓ Continuous]  Bin1   Bin2   Bin3

  Print offset..................:    0

  Printer control codes.........:

  Print field labels............:    Yes  [✓ No]

  Number of copies..............:    1

  Number of records per page....:    12

  Number of labels across.......:    1  [✓ 2]  3   4   5   6   7   8

  Print expanded fields.........:    Yes  [✓ No]
MAILING.DTF              Print Options for 1) 12up, no fieldnam

Esc-Exit     F1-Help      F8-Define Page      F9-Go back      F10-Continue
```

Figure 3. The File Print Options screen.

TIP: When you don't print field labels, Q&A automatically suppresses blank lines caused by empty fields. When you print field labels, Q&A doesn't suppress anything—it prints all the labels even for blank fields. Compare the example of **Form # 1180** in Figure 2 with Figure 4.

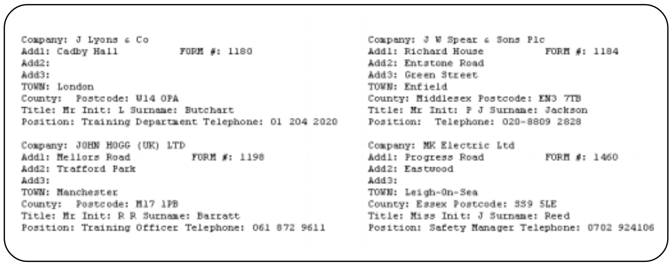
Number of Records per Page—This is where you tell Q&A how many records to *attempt to* print on a page. If set to 1, then you'll get one address per page as in the earlier example. This might be just what you want for printing out extensive information such as a datasheet or for placing information on a pre-printed form. Any higher number will print multiple records per page. For "mailing label"-style printouts, you'll want to set the highest number that will fit on a page. There's no harm in guessing high. If Q&A can only fit 12 records per page, you'll get the same result whether you specify 12, 14 or 30.

Number of Labels Across—This is self-explanatory. A setting of 1 across is useful if you want to print out information which might be quite lengthy, because you have the full page width. A setting of 2 across or even 3 across gives you a way of producing compact listings of names and addresses. Figure 4 shows the same information as in Figure 2, but printed with field labels and using 2 labels across.

Print Expanded Fields (Yes / No)—Set this to *Yes* to print the full text where a field's contents are greater than the field's width (a field where you have to press F6 to edit it). Otherwise, long entries are truncated.

TIP: Regardless of this setting, data will be truncated if *Number of labels across* is set to greater than one. For multi-column label-like output, Q&A preserves the position of each "label" on the page. This is clearly incompatible with printing all the data in a field regardless of its length. With *1 label across* and *Print Expanded Fields* set to *Yes*, Q&A will print each record in full, so you might get a different number of records on each page.

Continues on page 9



```
Company: J Lyons & Co                                    Company: J W Spear & Sons Plc
Add1: Cadby Hall          FORM #: 1180                   Add1: Richard House          FORM #: 1184
Add2:                                                    Add2: Entstone Road
Add3:                                                    Add3: Green Street
TOWN: London                                             TOWN: Enfield
County:  Postcode: W14 0PA                               County: Middlesex Postcode: EN3 7TB
Title: Mr Init: L Surname: Butchart                      Title: Mr Init: P J Surname: Jackson
Position: Training Department Telephone: 01 204 2020     Position:  Telephone: 020-8809 2828

Company: JOHN HOGG (UK) LTD                              Company: MK Electric Ltd
Add1: Mellors Road          FORM #: 1198                 Add1: Progress Road          FORM #: 1460
Add2: Trafford Park                                      Add2: Eastwood
Add3:                                                    Add3:
TOWN: Manchester                                         TOWN: Leigh-On-Sea
County:  Postcode: M17 1PB                               County: Essex Postcode: SS9 5LE
Title: Mr Init: R R Surname: Barratt                     Title: Miss Init: J Surname: Reed
Position: Training Officer Telephone: 061 872 9611       Position: Safety Manager Telephone: 0702 924106
```

Figure 4. *Two labels across* output with field labels.

@Help

EDITED BY WILLIAM HALPERN

Retrieve Records by Irregular Date Ranges

I have a special need related to a database I use to track retirement annuities for past employees. The records have one date field—their date of retirement. Every so often, I need to find all the records for people who retired before a certain year, in a certain month, during a specific range of days in that month. For example, right now I need to find all records for people who retired in any year before 1999 between October 15 and October 31. I can't seem to come up with a way to do this.

Rick

This is easy enough using Q&A's date functions along with Retrieve Spec programming expressions. Try typing the following at the Retrieve Spec screen:

In the Date Field: `#1`

In a second field (any other field):

```
#2: ={@Str(@Year(#1)) < 1999}
```

In a third field (any other field):

```
#3: ={@Str(@Month(#1)) = 10}
```

In a fourth field (any other field):

```
#4: ={@Str(@Dom(#1)) > 14}
```

Press F10 and Q&A will find all the records you're looking for. Here's what makes it work.

When you place a record retrieval restriction in multiple fields, Q&A's default is that a record must meet the restriction in *all* those fields to qualify. When you use a number beginning with the "#" character, Q&A interprets it as a programming expression rather than a literal retrieval restriction. Accordingly, what we typed in the date field doesn't affect what records will be found, but does allow us to reference this field in other fields. (This technique, in conjunction with Q&A's date functions, lets you perform all sorts of complex date range searches). In this case, the second field says: "*Get all the records where the year of the date is before 1999.*" The third fields says: "*Also make sure that the month of the date field is October (month 10).*" And the last one says: "*And ensure that the day of the month is greater than 14 (15th through 31st).*" By changing the *1999*, *10*, and *14*, you should be able to find any date range of records. You can use this kind of retrieval method in reports, Print Specs and the like as well.

Changing the Default Table View

This is driving me batty! In Q&A 5.0, I saved several Table View Specs. The trouble is that I can't get Q&A to remember the last one I used. It lets me load any of the saved Specs via the Alt-F8 key. But the next time I open the database the default Table View is something else, not the last one I used. I seem to remember that Q&A 4.0 remembered the last Spec I used. How do I make Q&A 5.0 do the same?

Shaun

This is one of those little quirks that can drive you crazy. First, though, let's get something straight. In this respect, Q&A 4.0 and 5.0 behave in exactly the same manner. What's odd is that Q&A remembers the last Table View Spec that you *type in at a blank Table View Spec screen* (when you press Shift-F6 from the Table View) not the last one you *used*. This seems to negate the usefulness of having saved Specs, but it's just one of those quirks we have to live with.

Q&A Doesn't Always Add Correctly?

I have two databases that are essentially the same. They both contain five fields where users type in numbers. Let's say the numbers are 1, 2, 3, 4, and 5. and the fields are #1, #2, #3, #4 and #5, respectively. They also have another field that sums these values. The programming in this field is: *#6 = #1 + #2 + #3 + #4 + #5*. In one database I get the correct value of "15" in field #6, in the other one I get an incorrect value of "12345" What am I doing wrong?

Erik

You've discovered Q&A's "loose" field formatting for different types of data. Though this product design philosophy can prove useful in many situations, at other

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.

times, as you've found out, it can be misleading. Q&A will let you enter improperly formatted values into certain types of fields. For instance, it will let you type non-dates into fields that are formatted with a "D" for dates. It will also let you perform arithmetic calculations in programming on text formatted fields. This gives you a great deal of latitude, but can cause problems, too.

For example, typing "March '98" into a date field will prevent Q&A from finding that record when you search for or sort on it by date. What you are seeing is similar. I'll bet that in the database that's giving you the correct result, at least one of the fields is formatted as a number (N) field, and in the other database, all the fields are formatted as text (T) fields. Q&A is smart enough to know that if you're adding (or performing any arithmetic calculation) on a group of fields, one of which is formatted for numbers, that it should look at the other text fields in the calculation and convert all of them that it can to numbers. However, if they're all text fields, even if they contain numbers, Q&A will treat *all* the values as strings and simply concatenate them (combine them into a single string value). Chances are, in the database that's giving you the "12345" result, all the fields are formatted for text values and are simply being strung together. In the one where the result is "15," at least one of the fields is formatted as a number field. What you must do is review your databases and make sure that you're defining your fields properly for the type of data they store. From the Main Menu, select *File* / *Design a file* / *Customize a file* / *Format values* and assign the proper field types to match your data. Of course you could simply change your programming approach and use the @Tonumber function this way:

```
#6 = @Tonumber(#1) + @Tonumber(#2) + @Tonumber(#3)...
```

My recommendation, though, is to do it right from the start by specifying the proper data type for each field.

New PC Blues (or 'I Can't Get No Expanded Memory')

For all of you Q&A 5.0 aficionados out there, be aware of an important issue with some new PCs appearing on the market. As you should know, *Expanded Memory* is critical to the proper operation of Q&A 5.0 for DOS. Without it, complex reports, large mail merges and even some searches and sorts just won't work. In the past, the solution often simply required the addition of the statement, *Device=C:\Windows\Emm386.exe RAM* in the config.sys file or a variation of this. But this doesn't work on some new PCs and there is *no* workaround. For example, Dell is currently shipping their Inspiron 7500 and 8000 series notebooks with only 56K of available paging memory. To successfully setup Expanded memory, you need a *minimum* of 64K of paging memory. Gateway is shipping certain models with "Promise ATA" hard disk controllers that usurp all the paging memory. Q&A 5.0 will *not* work properly in either of these environments. Certain HP and Compaq models have similar issues. What's the solution? Before purchasing a new computer, make sure it will support Expanded memory. If the vendor can't or won't tell you, see if you can get your hands on the model you're interested in and type *mem/c/p* at a DOS or command prompt. If you see a line that says something like this:

```
Free Expanded (EMS)  16,777,216  (16M)
```

you'll be okay running Q&A 5.0 on that machine. But if this line is missing when the config.sys file contains the proper Expanded memory device line (as the example shown earlier), then don't buy that computer if you plan to run Q&A 5.0 on it. This also applies to a lesser extent to Q&A 4.0 for DOS. Although Q&A 4.0 can perform some tasks more efficiently when Expanded memory is available, it doesn't *require* Expanded memory.

Can't Copy Programming Statements in Online Edition of *The Quick Answer*

I'm an Online Edition subscriber and I download my *Quick Answer* each month from www.quickanswer.com. I like having my *Quick Answers* in Acrobat Reader files because I can copy programming statements and other key information from them then paste the data anywhere I want, even into Q&A. The problem, though, is that when I try to select a programming statement to copy, Acrobat Reader selects the text across *both* columns of the newsletter, not just the text in the column that contains the programming I want. This means that when I paste the text, I have to do a lot of cleanup work to wind up with just the programming. How do I force Acrobat Reader to let me select/copy text from just one column or the other?

Nancy

By default, Acrobat Reader 4.0 activates the "grab" tool (the "hand") on a displayed document. This lets you scroll the document up and down simply by holding the left mouse button down and dragging the page. To *select* text, click on the "T" tool on the toolbar. This changes the "hand" to a vertical "I" bar. Then, to select text in just one of the columns, hold the Ctrl key down while "drawing" a box around the text you want to copy. When you release the Ctrl key and mouse, Acrobat Reader will confine your selection to just the text inside the box. You can now pull down the *Edit* menu and click *Copy* to copy the selected text (or programming statements) to the Windows Clipboard. You can then paste your selection wherever you want, including into a database's Program Spec if you have Q&A running in a window. —T.M.

Bill Halpern owns Professional Computer Technology Associates (PCTA) in Newtown, Pennsylvania, 215-598-8440, www.pcta-usa.com, email pcta@comcat.com.

Create Dynamic Record Navigation Menus

Cliff Sobin

Making a database easy to use is an essential part of its design. It isn't enough to have records stuffed with a myriad of interesting data and calculations. If the person *using* the database can't quickly and easily navigate to, understand and update information as soon as the record displays, then any attempt at efficiency is lost. For this reason, just as in Web site design, you have to take into consideration what specific information a user might be after, enabling them to go directly to that spot, and displaying the information so it can be immediately grasped.

A Q&A database can contain up to 10 screens and 2,000 fields. Because many designs use on-field-entry and on-field-exit Goto commands to move the user to specific fields in a record (*<#10: Goto #50*), for example), paging up and down the form is often not a viable option. Even in the absence of programmed navigation, it's usually just not the best way to navigate. If the record contains 10 pages of fields, how does the person know where the target field is located, if it even exists, and whether or not it contains information that's worth navigating to? Why not give users a "window" they can peer through to see what's in the record, along with the ability to get to the right field in a hurry! Q&A 5.0's Userselect command is the tool that can do it.

A Userselect "window"?

In my November 1999 article, "Have a Window on Your World with Userselect," I showed how the Userselect command can be adapted to allow a user to quickly and conveniently view information stored in an external database record. While this can be useful in many applications, it's often more useful to be able to see deep within the record you've retrieved. By analogy, while international concerns are important, it can be vital to know what's going on in your own backyard!

To show the technique, I created a little database named **Medlist.dtf**. (It's in the download file for Online Edition subscribers.) This greatly simplified version of a case management system we use in our law office is comprised of just two screen pages. The first contains the name of the client and a case memo field. The second contains fields that store the names of medical providers used by the client, memo fields for the treatments provided, and fields that store the amount of any bills to date. When the record displays, all the user sees is the information on page one until our Userselect "window" comes into play.

I call it a "window" because it lets users see, in an abbreviated fashion, certain key information that's already in the record. But it also serves as a handy navigation menu because it lets users jump directly to the field they're most likely to want to read and/or update. (See Figure 1.)

Navigating a record via a Userselect menu is easy to design into a database. Just follow these steps:

1. Add a field in which all navigation programming will be done. (I'll call this the *index* field).

2. Decide on the fields you want to provide the user with access to via the menu.

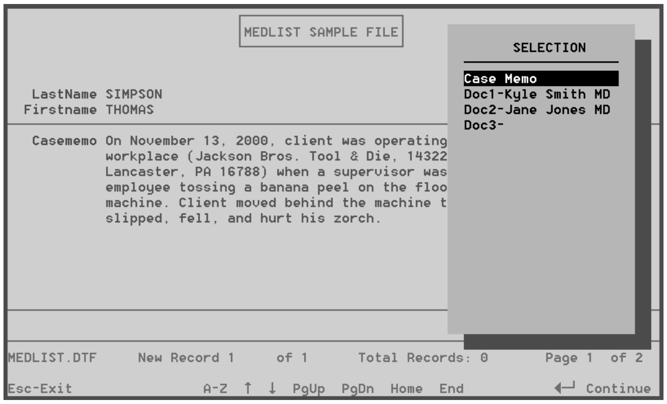
3. For each such field, add an on-field-exit Goto statement (you can use the Program or Navigation Spec for this) that will send the user back to the index field once they've finished with that field or combination of fields.

4. Create an on-field-entry Userselect statement for the index field using one of the following two methods:

```
<#75 = @Userselect("1-Last Name,2-CaseMemo,
3-Doc1,4-Doc2,5-Doc3")
```

```
<#75=@Userselect(@Insert("C:\qa5\work\
navigate"))
```

Concludes on page 11



```
                    MEDLIST SAMPLE FILE                        SELECTION
                                                         ─────────────────────
                                                         Case Memo
   LastName SIMPSON                                      Doc1-Kyle Smith MD
  Firstname THOMAS                                       Doc2-Jane Jones MD
                                                         Doc3-
   Casememo On November 13, 2000, client was operating
            workplace (Jackson Bros. Tool & Die, 14322
            Lancaster, PA 16788) when a supervisor was
            employee tossing a banana peel on the floo
            machine. Client moved behind the machine t
            slipped, fell, and hurt his zorch.

MEDLIST.DTF     New Record 1     of 1      Total Records: 0       Page 1  of 2
Esc-Exit                A-Z  ↑  ↓  PgUp  PgDn  Home  End          ↵ Continue
```

Figure 1. This Userselect list provides a "window" on what's already in the record and gives the user the option of going there with a mouse click.

Print Specs…

cont’d from page 5

Enhancing Field Codes

You’ve already seen that you can specify a number after the “+” or “x” code at the Fields Spec. The “+” adds the requisite number of spaces before the next field, and the “x” code specifies the number of additional lines to print after the field. For example, the following prints one blank line between the line with **Surname** and the line with **Company**.

```
Surname: 20x,2
Company: 30x
```

> ***TIP:*** Note that the default (“x” all by itself) adds *one* carriage return, whereas *x2* adds *two* carriage returns—that is, *one* blank line, not two.

There are two additional codes you can add at the Fields Spec. If you have *Print Field Labels* set to *No*, you can still print labels for fields selectively. Just add the *L* code as in the following example:

```
Position: 120 + L
```

Furthermore, you can use a field label other than the one on the database form. After the *L* (upper or lower case), simply add the field label you want, in parentheses, as follows. (Figure 5 shows the result of doing this.)

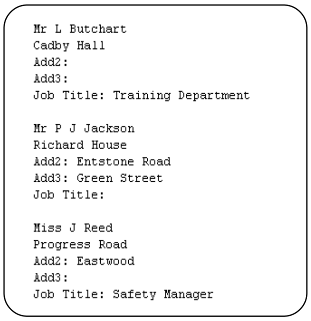
```
Position: 120 + L(Job Title:)
```

Notice that the field labels **Add2:** and **Add3:** are printed even for blank fields.

> ***TIP:*** You must add the colon (“:”) yourself if you want it printed, as in the **Job Title:** example in Figure 5.

Another, perhaps less useful, enhancement is that you can specify the number of characters of a field to print. You do this by adding another comma and a number after the *number of spaces to skip*. Here’s an example:

```
Position: 120+,2,5 L(Job Title:)
                               Telephone: 130x
```



```
Mr L Butchart
Cadby Hall
Add2:
Add3:
Job Title: Training Department

Mr P J Jackson
Richard House
Add2: Entstone Road
Add3: Green Street
Job Title:

Miss J Reed
Progress Road
Add2: Eastwood
Add3:
Job Title: Safety Manager
```

Figure 5. A field label (Job Title) selectively included.

This yields:

```
Job Title: Train  01 204 2020
Job Title: 020-8809 2828
Job Title: Train  061 872 9611
Job Title: Safet  0702 924106
```

> ***TIP:*** To use the *Only print n characters* code, you *must* use a *Skip n spaces/lines* code. This, again, is not exactly intuitive, as the codes have to be entered in the opposite order from which the results appear. In the earlier example, **120+,2,5** prints the first five characters of the **Position** field, and *then* leaves a two-character gap before the next field.

Text Enhancements

You can enhance text the same way you can in Q&A reports. You can select a regular (default) font for the whole Print Spec by pressing Ctrl-F9 at the Fields Spec. You can apply bold, underline, italics and fonts to any individual field by putting your cursor in the field and pressing Shift-F6. You can enhance the field data (by applying the enhancement to the number), or the label (by applying the enhancement to the “L”), or to both (enhance both). This can be useful in making important fields stand out with bold or perhaps a larger font, or for printing out a wordy comments field in italics.

Co-ordinate Print Specs

Co-ordinate Print Specs differ from Free-form Print Specs only in the way you specify the position of the fields at the Fields Spec. Instead of *10+*, *20x* and the like, you put the actual page co-ordinates (printing position) in each field you want to print.

You can use row and column co-ordinates, inches, or centimetres. Here are a few examples:

City: 3,7 means “print the city field on the third line starting in column 7.”

City: 0.5",2.5" means “print the city field 0.5-inches from the left margin and 2.5-inches down from the top margin.”

City: 0.5cm, 2.5cm means “print the city field 0.5 cm from the left margin and 2.5 cm from the top margin.”

> ***TIP:*** In a Co-ordinate Print Spec, if you specify more than one record per page, the next record prints at the specified page position based on the *last field* of the previous record. So you *can* print multiple records on a page using this method.

Other than the above, Co-ordinate Print Specs are identical to the Free-form variety.

Summarizing the different printing methods

The five different printing methods in Q&A vary in the appearance of the printed output, whether the method includes saved Retrieve and Sort Specs, whether you can include things like headers and footers and page numbers, whether you can include derived columns or their equivalent, and also where Q&A stores the information.

Where Q&A stores printing elements

Print Specs and reports, for example, are stored in the database (the .DTF file). This means they travel with the database and are available to anyone who can access the database. You don't have to remember to back these up separately. The downside is that you might put a lot of effort into a Print Spec but Q&A doesn't give you a way to use it with a different database that might have some or all of the same fields.

Merge documents, in contrast, are stored as discrete files. This means that they can easily be modified to print data from different databases. They have to be backed up separately though, and might be stored on one user's local drive and not available to other network users.

Module or Method	Stored in	Use with any DTF?	Headers / Footers	Spec Includes Retrieve / Sort?	Specify a Font?	Multiple Fonts?	Derived Columns?
F2 Search/Update	Not stored	N/A	Yes	N/A	Yes	No	No
Mailmerge	A document	Yes	Yes	No	Yes	Yes	Via Programming
Report	.DTF file	No	Yes	Yes	Yes	Yes	Yes
Mailing Labels	QALABEL.OVL	Yes	No	No	Yes	Yes	Via Programming
Print Spec	.DTF file	No	Yes	Yes	Yes	Yes	No

Table 1. Comparison of Q&A's five printing methods.

Mailing Labels are different again. Q&A stores them in a single file named QALABEL.OVL which is always in the main Q&A program files directory. This might be convenient, but then again, it might not, because it means that if Q&A is run from each network user's local drive (as is recommended), then each user will have their own mailing label file. This can obviously cause standardisation problems.

Table 1 summarises some of the differences between the five printing methods covered in this series.

Conclusion

By familiarising yourself with *all* of Q&A's printing methods, rather than just one or two, you'll be better able to print out your data more efficiently and attractively.

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

Printing Fixed Text

Print Specs only allow you to print field contents and labels. They don't give you a way to print fixed text (as you can in a mailmerge document or mailing label). You can still do it, though, with a little ingenuity. Here's how.

A client (a boarding school) requested a listing of all the subscribers to the *Old Girls' (alumni) Newsletter*. A report wasn't appropriate and I decided a Free-form Print Spec printed in one column was the way forward. We needed to print the following on three lines:

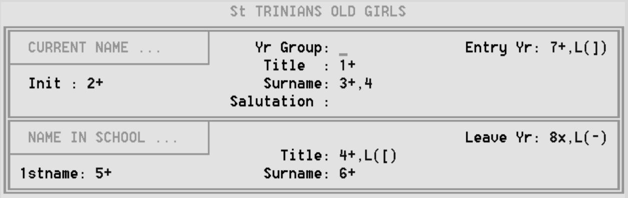
```
Current name  [ Name when in school ]  years attended
Full Address
Telephone number(s)
```

The *name in school* had to be distinguished clearly, ideally in brackets, and the years in school had to be shown in a format with the **Entry Year** followed by a hyphen and then the **Leaving Year**, like this:

```
Lady Sophia Douglas [Miss Sophia Duchamp] 1956 - 1961
```

The challenge was to add the necessary "fixed text"—the brackets and the hyphen—so I set *Print Field Labels* to *No* at the Print Options screen and used the custom label option to add the required text. (It helps to sketch this out on paper because the closing bracket, for example, is the custom field label for the following field—the **Entry Year**.)

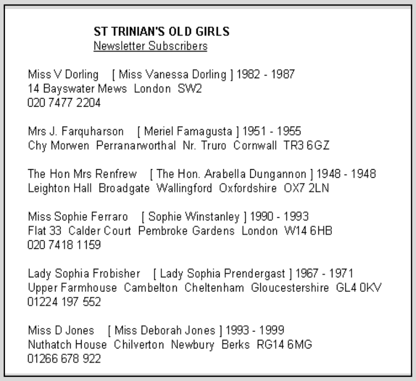
Figure 6 shows the Fields Spec, and Figure 7 shows the resulting output.



```
                      St TRINIANS OLD GIRLS
 CURRENT NAME ...        Yr Group: _              Entry Yr: 7+,L(])
                          Title   : 1+
 Init : 2+               Surname: 3+,4
                       Salutation :

 NAME IN SCHOOL ...                               Leave Yr: 8x,L(-)
                            Title: 4+,L([)
1stname: 5+              Surname: 6+
```

Figure 6. Fields Spec (partial) showing custom field labels.



ST TRINIAN'S OLD GIRLS
Newsletter Subscribers

Miss V Dorling   [Miss Vanessa Dorling] 1982 - 1987
14 Bayswater Mews London SW2
020 7477 2204

Mrs J. Farquharson   [Meriel Famagusta] 1951 - 1955
Chy Morwen Perranarworthal Nr. Truro Cornwall TR3 6GZ

The Hon Mrs Renfrew   [The Hon. Arabella Dungannon] 1948 - 1948
Leighton Hall Broadgate Wallingford Oxfordshire OX7 2LN

Miss Sophie Ferraro   [Sophie Winstanley] 1990 - 1993
Flat 33 Calder Court Pembroke Gardens London W14 6HB
020 7418 1159

Lady Sophia Frobisher   [Lady Sophia Prendergast] 1967 - 1971
Upper Farmhouse Cambelton Cheltenham Gloucestershire GL4 0KV
01224 197 552

Miss D Jones   [Miss Deborah Jones] 1993 - 1999
Nuthatch House Chilverton Newbury Berks RG14 6MG
01266 678 922

Figure 7. Print Spec output with custom labels to emulate fixed text.

Navigation Menu…cont’d from page 8

The first of these statements has the menu selections hardcoded into the Userselect statement. The second one inserts an ASCII file containing preformatted menu items that you can modify at any time without locking users out of the database. The selections can and should include fields on other pages of the database.

5. Use an *@Left* command such as *#75=@Left(#75,1)* to strip away all but the first character. This saves programming space and is useful if you decide to use the inserted ASCII file method.

6. Use *Goto* commands to move the user to the selected field, like this:

```
If #75=”1” then goto #120;
If #75=”2” then goto #40;
If #75=”3” then goto #115;
```

7. Place an on-field-exit program in the first field of the database that will send the user to the index field. (I prefer my users to be able to see the entire first page when the record first displays before partially blocking the record with the navigation menu.)

8. Consider putting index fields like this on each page, with connections on each page’s Userselect menu to the other pages’ index fields as well as other regular fields.

Enhancing the menu

With the Userselect navigation menu, you can restrict access to particular fields while providing maximum flexibility by displaying all the options. Userselect navigation can also be employed once the user has selected a field, if you want to further control and display his options. However, a problem remains. *How do you communicate to the user what information exists in a field on a back page so he can determine whether or not it’s worth going there?* The answer is simple—by dynamically *changing* the Userselect menu!

To do this, you add one more field that should be made invisible on the form. In the sample Medlist database, page two contains space for three different doctors, a memo field for the treatment history by each doctor, and a field containing their bills to date. The goal was to create a design that would permit the user to see whether or not a doctor is listed on page 2 and, if so, permit him to easily move to that doctor’s field. To do this, an additional field I’ll call *Compiler* was added to the database and made invisible by coloring it (at the Change Palette Spec) the same as the form’s color. On entry to the index field, before the Userselect programming, you’ll need programming that places the information in the Compiler field to be used by the Userselect statement.

Assume you have the following fields:

```
#40 on page 1, where the Case Memo is;
#110 on page 2, where the First Doctor is listed;
#125 on page 2, where the Second Doctor is listed;
#145 on page 2, where the Third Doctor is listed;
#50 is the Compiler Field;
#75 is the Index field.
```

Type an on-field-entry statement like the following in the index field:

```
< #50=@Str(“Case Memo”+”,”+”Doc1-”+#110+”,”+”Doc2-
”+#125+”,”+”Doc3-”+#145)
```

This statement combines the field name on page 1 that you might wish to navigate to with the field *contents* on page 2 that you might wish to navigate to. *Doc1* is the location of the first Doctor and *#110* is the name of that Doctor. The same is true for *Doc2*, and so on. The comma after each entry creates a new line in the Userselect menu. The result is a dynamic navigation window that provides information to the user as to where they can go and what’s there. If there are no doctors listed, then there’s no reason to go to page 2 (unless you want to add one). If the Doctor you want to add is already listed, why waste your time?

If you want to get really fancy, you could include programming to add the amount of the bill (#120) to the name in the Compiler field. (Just keep in mind the 19-character display limitation of the Userselect window. I will leave to another day how John Dow’s DTFDATA could be used to add even further functionality.)

The final steps are as in the earlier example. Have *@Left(#75,4)* strip all but the first four characters from #75 after Userselect places the entire selection in the Index field. Then, a command such as *If #75 = “Doc1” then goto #110* will take the user to the desired field on the second page.

The end result of this exercise is that your users no longer have to guess what is contained in this “busy” database record and then figure out how to navigate to that information. The learning curve for new users is shortened. The first screen of the database doesn’t need to contain all necessary information because the Userselect menu provides a window into the record’s interior. This allows you to focus on good screen design. Finally, your users will be able to quickly determine whether information must be added or updated because they’ll be automatically visually prompted as to the contents of fields located deep within the record.

A database should let you store and retrieve the most accurate information possible in the shortest possible time with the smallest learning curve. A dynamic Userselect navigation window/menu can help accomplish that for you in the least obtrusive way.

Cliff Sobin is an attorney and managing partner in the law firm of Berman, Sobin & Gross, LLP in Silver Spring, Maryland. He has designed applications for his law firm for the last nine years. 301-670-7030, bsgcbs@aol.com.

Lookup Text Files…cont'd from page 3

Now, all that's left to do is to find the match on the **PartNo** in **Price** and retrieve its associated description, bin number, quantity on hand, and price.

The program finds the matching part number in **Price** (via @Instr) and temporarily stores this position in the **QtyOH** field. It then copies to **Desc** everything in **Price** from that position plus 100 characters. We assume that within those 100 characters we'll find the carriage return that marks the end of the pertinent row. Accordingly, the program fills the **Price** field with the now isolated row containing the data for just the selected part number. If, for example, the selected part number were *V567333*, the **Price** field would now contain:

```
V567333;Cover;12C;43;8.43
```

Now, the final parsing of the values for the fields begins. The program gets rid of the part number because we've already got that. The description is extracted and placed in the **Desc** field, then deleted from the parsed string, leaving the bin, quantity on hand and price. This remaining string is then similarly parsed to extract those values and place them in their respective fields.

It all happens in the blink of an eye. It looks like an ordinary XLookup or Lookup has occurred, except all the action starts with an inserted text file rather than with an external database or Lookup Table. And, in this case, no additional fields were required to do it.

Notes

Any field where you insert a text file and/or run a parsing operation must be a text field, though you can have number, date and money fields and copy parsed values *into* them. If you don't have a few text fields you can use for temporary storage, you'll need to add them.

To work around the maximum 32K file size, you should be able to use @Shell to run CHOP.EXE with a switch to split a +32K file into two or more smaller files. You could then read-in and process each resulting file. (CHOP can be download from the "Freebies" page at www.quickanswer.com.)

I also designed a little Q&A 5.0 database that parses an address text file (*name;street;city;state;zip*) into Name, Street, City, State, and Zip fields. This one doesn't include the pick-list feature, though—you have to type in the person's name to "lookup" and retrieve the address. If you'd like a copy of this database, email us at mailbox@quickanswer.com and ask for ASCIILST.DTF.

Tom Marcellus is editor of *The Quick Answer*.

MARBLE
PUBLICATIONS
1927A Harbor Blvd. PMB 389
Costa Mesa, CA 92627

December 2000 Issue

First Class