



T·H·E Quick Answer

The independent monthly guide to Q&A expertise

Print Labels One at a Time on Your Laser Printer

You might think that your laser or inkjet printer is useless for printing individual labels. But there's a way to print just one label on a sheet, then use that same sheet for the next label, and the next one . . .

By Tom Marcellus



HAVE you ever wished you could quickly run off a label or two on your laser or inkjet printer without wasting the rest of the label sheet? Unlike dot matrix printers, lasers and inkjets must be fed whole pages. If you need a label, you have to run the entire sheet through the printer. But so what? By exercising reasonable care, no harm is done by putting the sheet back through the printer. And by setting up Q&A so it knows which label to print, you can use all the labels on the sheet and take advantage of the high quality output laser printing provides.

I'll show you how to "train" Q&A to print a label in a precise location on a sheet of labels so you can use the remaining labels as you need them. I'll illustrate using Avery 5164 labels and a LaserJet 4P printer. You can adapt the technique to any printer with a sheet feeder and to most any standard sized label sheet.

Continues on page 3

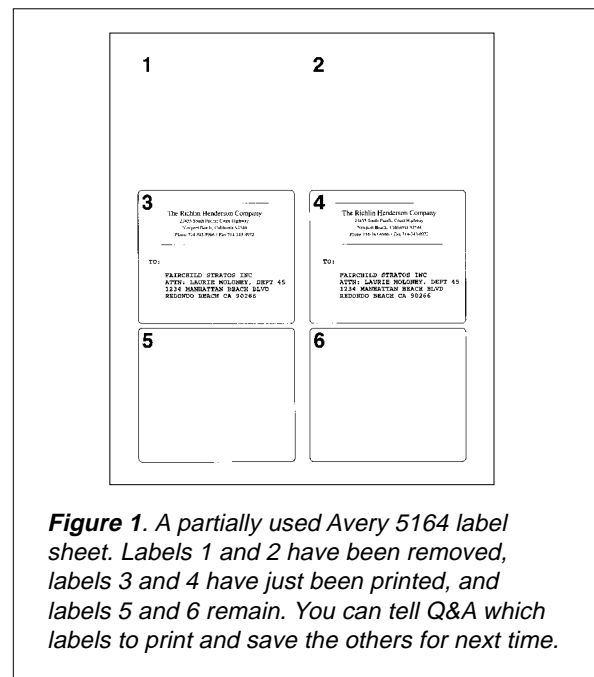


Figure 1. A partially used Avery 5164 label sheet. Labels 1 and 2 have been removed, labels 3 and 4 have just been printed, and labels 5 and 6 remain. You can tell Q&A which labels to print and save the others for next time.

CONTENTS	
Print Labels One at a Time, Tom Marcellus	1
README.1ST, Tom Marcellus	2
@HELP, Dave Reid	7
QuickTips	6, 13
Tweak Macro Files, James Pogany	9
Data-Entry Form Layout, Tom Marcellus	11
Simplify with Indirection, Jeff Nitka	14





The Quick Answer

Editor Tom Marcellus
 Managing Editor Roland Winkler
 Copy Editor Laurie Moloney
 Production Editor Paul Gould
 Publisher Susan Harker
 Editorial Director Dian Schaffhauser
 Circulation Director Sharon Whiting

The Quick Answer (ISSN#1052-3820) is published monthly (12 times per year) by Pinnacle Publishing, Inc., 18000 72nd Avenue South, Kent, WA 98032. Cost of domestic subscriptions: 12 issues, \$79.00; 24 issues, \$142.00. Outside the U.S.: 12 issues, \$99.00; 24 issues, \$172.00. Single copy price: \$10.00; 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; multiple-copy discounts are available.

For European newsletter orders, contact: Tomalin Associates, The Old Bakery, High Street, Great Bardfield, Essex CM7 4RF, United Kingdom
 Phone: +44 (0)1371 811299; fax: +44 (0)1371 811283
 12 issues: £69; 24 issues: £109

Second-class postage paid at Kent, WA and additional entry points. USPS#006941.
 POSTMASTER: Send address changes to *The Quick Answer*, PO Box 888, Kent, WA 98035-0888.

Copyright © 1990-1995 by Pinnacle Publishing, Inc. All rights reserved. No part of this periodical may be used or reproduced in any fashion whatsoever (except in the case of brief quotations embodied in critical articles and reviews) without the prior written consent of Pinnacle Publishing, Inc.

Address editorial correspondence, @HELP questions, or requests for special permission to: The Editor, *The Quick Answer*, PO Box 888, Kent, WA 98035-0888. Or leave CompuServe messages at 72600,140. We will publish only those questions we feel are of general reader interest; individual responses are not possible. For subscription information, back-issue requests, and bulk orders, call our Customer Service department at 800-788-1900 or 206-251-1900.

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

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

This publication is intended as a general guide. It covers a highly technical and complex subject and should not be used for making decisions concerning specific products or applications. This publication is sold as is, without warranty of any kind, either express or implied, respecting the contents of this publication, including but not limited to implied warranties for the publication, quality, performance, merchantability, or fitness for any particular purpose. Pinnacle Publishing, 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 Pinnacle Publishing, Inc.



Volume 6, Issue 2

README.1ST

THOUGH I purchased my first laser printer more than five years ago, I hung on to my trusty old dot-matrix for one reason: to print mailing labels. I frequently print one or two labels at a time, and my laser offers no way to print a label without wasting the rest of the labels on the sheet. But, as they say, necessity is the mother of invention.

When my dot-matrix was away on sick leave, I decided to run a sheet of Avery labels through the laser several times to see what happened. I knew the printer manufacturer cautioned against this, but I was determined to find out whether or not it was possible without causing a jam. To my delight, by exercising reasonable care, no harm was done by putting the same sheet back through the printer.

If you'd like to be able to quickly run off a label or two on your laser or inkjet printer, and use that same sheet the next time you need a label, I'll show you how. A caveat, though: improperly handled or carelessly fed laser label sheets can jam and even damage the printer.

Speaking of risks, in the November 1994 issue ("Use Your Programming Allowance More Effectively"), Spencer Wulwick pointed out that you can damage a Q&A database by programming it too heavily. This month, beginning on page 14, **Jeff Nitka** tells you how to tap the hidden power of subroutines and indirection not only to avoid that risk, but to make your programming more manageable and Q&A more responsive.

In past issues of *The Quick Answer*, our experts have offered a variety of techniques to automate procedures that usually require manual intervention. Building on these earlier approaches, **Jim Pogany** has come up with an inventive DOS batch file technique that, with a keystroke, captures the key value from the record you're viewing and uses it to display related records in another database.

In Q&A for DOS, the database you design is the data-entry form you'll use day in and day out, so it's in your best interest to make it easy on the eyes as well as functional. In "Master Data-Entry Form Layout," you'll discover ways to place fields on your form to make it both attractive and easy for anyone to use.

Tom Marcellus
 Editor

The Quick Answer is looking for tips, tricks, and application techniques that solve problems and make Q&A easier to work with. If you have found a way to work smarter with Q&A, submit your tip or article (and a biographical sketch) on disk in Q&A Write for DOS or Windows, WordPerfect for DOS, or Microsoft Word for Windows. Enclose a printout—and a daytime phone number.

Mail your submission to: The Editor, *The Quick Answer*, PO Box 888, Kent, WA 981035-0999. Honoraria are paid upon publication.





Print Labels *Continued from page 1*

Label design considerations

The Avery 5164 label design includes six labels per sheet, two across and three down. (See Figure 1.) Each label is 3/8-inches high by 4-inches wide, an ideal size for parcels and catalog envelopes and large enough to include your company's name and return address in addition to the addressee information.

For the label in Figure 1, I used Q&A's HPLASER4.FNT font file. In the return address area, the company name is in CG Times Medium 12 point, with the address in 8 point. The addressee block is printed in Courier 10 Bold. The lines above and beneath the return address are underscores.

Step 1: Modify the built-in label template

Though I'm using Avery 5164 labels, you can use other label sizes. (See the sidebar, "Laser Label Printing Precautions.") Q&A 4.0 comes with an Avery 5164 label template. Instead of overwriting this original template, copy it to a new name and work with the copy. This way, you'll have the original template in case you need it.

Copy the original 5164 label

My approach requires seven copies of the Avery 5164 label; one copy to correspond to each of the six label positions on the label sheet and another copy

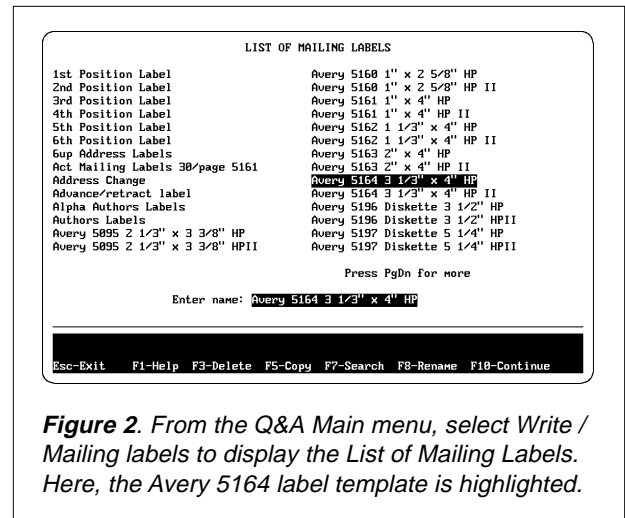


Figure 2. From the Q&A Main menu, select Write / Mailing labels to display the List of Mailing Labels. Here, the Avery 5164 label template is highlighted.

for printing one or more entire sheets of labels.

Start by copying the 5164 template to the name 6up Address Labels. (The missing space in 6up places this label in its proper position on the list. See Figure 2.) To copy the label, highlight it, press F5, type 6up Address Labels, and press Enter.

Add text and merge fields to the label

This 6up Address Labels template will be your default for printing entire sheets of labels (six or more, in this case) when merge printing addresses. You can also use this template to print multiple copies of the same label by changing the Number of copies at the Print Options screen to the number of

Laser Label Printing Precautions

Laser printer manufacturers offer guidelines for printing adhesive-backed label sheets. For example, use only label sheets that meet the manufacturer's specifications and store them flat and away from excessive moisture.

Avoid jams by feeding label sheets through the manual feed slot rather than from the cassette, and use the printer's straight-through paper path so the printed label sheets exit from the rear of the printer.

Though Hewlett Packard cautions against printing label sheets with labels already removed, I've run partially used label sheets over and over again through LaserJet IIP, III, and 4P printers without a hitch. I take the following additional precautions:

- I don't use sheets with more than eight labels.
- I never run a "hot" label sheet right back through the printer.

- I make sure to store partially used sheets flat.
- I print the last label on the sheet first, followed by the second to last label and so on up the sheet. This way, the remaining labels at the top of the sheet reinforce its leading edge, helping it pass safely through the printer.

To add even more stability to a partially used label sheet, you can place a sheet of plain bond under it and run both through the printer as though they were a single sheet.

The success you'll have depends on your printer's design (consult your printer manual) and condition and the number of times you run the same sheet through the printer. If you can't afford to risk a jam (label jams in any printer can be a pain to clear) this technique isn't for you.





labels you need. This way, if you routinely mail packages to the same addressee, you can print sheets of labels in advance.

Load your 6up Address Labels template and, at the Label Layout screen, delete Q&A's default merge fields and add the text and merge fields you need. Enhance your company's name and return address with fonts to impart a clean professional look.

How you lay out the addressee block depends on how you'll use the label. If you'll be merging addresses from a database, add your merge fields. (Press F1 from the Label Layout screen for help on placing merge fields.) If you plan to type in the addressee information when you print the label, leave that portion of the label blank for the time being.

Figures 3 and 4 show two versions of the same template. In the template shown in Figure 3, the addressee information is typed in at printing time. The template shown in Figure 4 is the same design

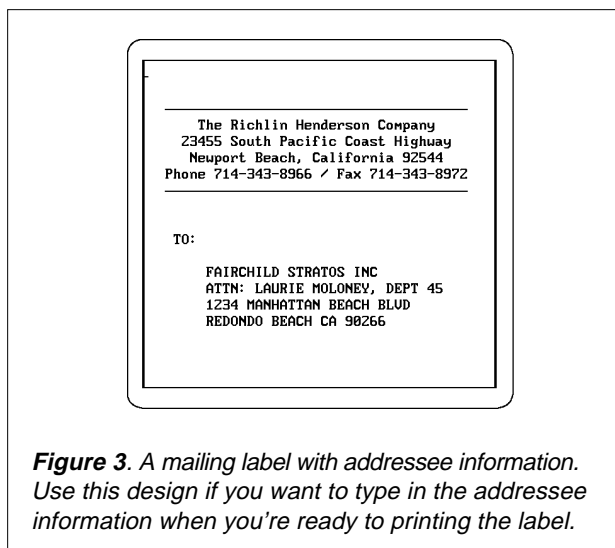


Figure 3. A mailing label with addressee information. Use this design if you want to type in the addressee information when you're ready to printing the label.

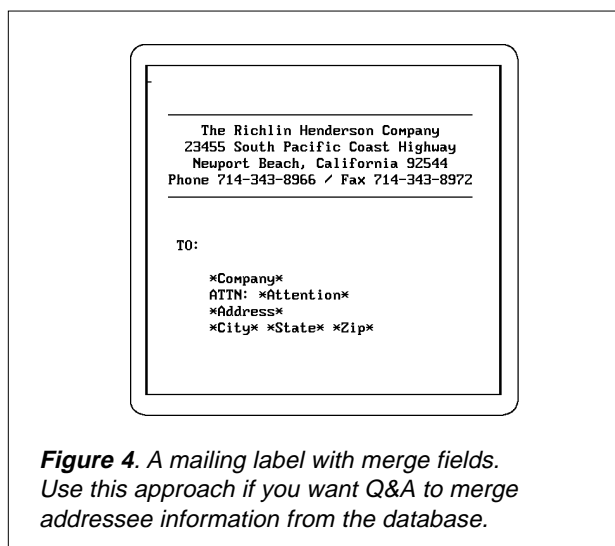


Figure 4. A mailing label with merge fields. Use this approach if you want Q&A to merge addressee information from the database.

but substitutes merge fields for the addressee information. The merge label design assumes that you've selected a merge database and that you'll specify the record or records to merge at the Retrieve Spec. (See the sidebar, "Automate Your Label Printing.")

When you've finished typing and enhancing your label text, press Shift-F8 to save the template.

Step 2: Copy your default label

Next, copy the 6up Address Labels template to the name 1st Position Label. This new template will correspond to the first label on the label sheet. (See Figure 1.) Eventually, you'll have 2nd Position through 6th Position labels—they'll all appear at the top of the List of Mailing Labels in numerical order, making them easy to find and select. (See Figure 2.)

Bring the 1st Position Label onto the Label Layout screen, then press F2 for Print Options. Figure 5 shows the Print Options settings for the 1st Position Label. Note that Number of copies is set to 1, Type of paper feed is set to Manual, and Number of labels across is set to 1. The other settings are Q&A's default settings for the Avery 5164 label template.

When the 1st Position Label template is finished, test print it to confirm that it prints in the space allocated on the label sheet. (Don't waste label sheets while testing. Use plain paper until you've got it right.) Press Shift-F8 from the Label Layout screen to save the template.

Step 3: Make five more copies of the label

Next, make an additional five copies of your 1st Position Label. Copy it to 2nd Position Label, 3rd Position Label, and so on through 6th Position Label. When you've done this, all six position labels, in addition to the 6up Address Labels template, should appear in order on the List of Mailing Labels. (See Figure 2.)

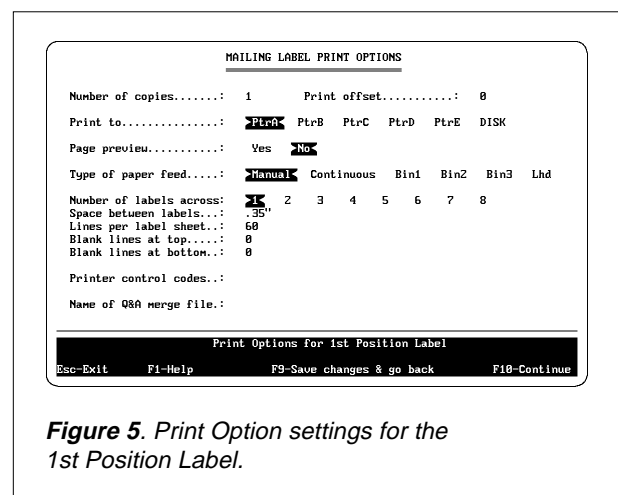


Figure 5. Print Option settings for the 1st Position Label.





You can now modify, in turn, the Print Options for the 2nd through 6th labels.

Select the 2nd Position Label and press F2 for Print Options. In the Print offset field (see Figure 5), type 40 and press F10 to test print the label. It should print at the second label position on the label sheet (to the right of the first label). When you've got it squared away, press Shift-F8 to save the label.

Continue by selecting labels 3 through 6 in turn. Set each label's Print offset and Blank lines at top as shown in Table 1. Test print and save each one.

When you've changed the Print Options of each label in accord with Table 1, your new one-at-a-time address label system is ready. When you start a new label sheet, print the 6th Position Label first. The next time you use this sheet, print the 5th Position Label, and so on through the 1st Position Label.

Q&A can't remember the position of the last label printed (watch some smarty-pants prove me wrong), so before you select a label, you'll have to look at your label sheet to determine the next available label position.

Table 1. Print Options settings for the six labels. The Print offset sets the horizontal (left to right) position where Q&A begins printing the label. Blank lines at top controls the vertical (top to bottom) position of the label. All labels in this table must have Number of labels across set to 1.

Label Name	Print offset	Blank Lines at top
1st Position Label	0	0
2nd Position Label	40	0
3rd Position Label	0	20
4th Position Label	40	20
5th Position Label	0	40
6th Position Label	40	40

Two or three labels at the same time

Though the technique I've described prints one label at a time, it'll print multiple labels as well.

Suppose you have a six-label sheet with label 6 already used, and you want to print two identical

Automate Your Label Printing

If your database already contains the address information you need, you can save keystrokes by merging the addressee information with the label, using the Retrieve Spec to select the record. You can automate label printing even further by having Q&A fill out the Retrieve Spec for you. All you need is a new field in the database with a simple program, and a macro.

Start by redesigning the database form. Add a new field in an out-of-the-way location and a line of text like the following:

Press Ctrl-L to Print Address Label

Go to the Program Spec, move to the new field, and enter #500 = @Number.

Press F8, enter 500 in the On Exit Field ID line, and save the Spec. This statement will assign the next highest number to any record you view, allowing the macro to select it when you give the command to print a label.

Create the macro

Use Search/Update to display any record from the pertinent database. Start the macro recorder, and perform these steps:

1. Press Shift-F10 and Esc to save, exit, and return to the Q&A Main menu.

2. Select Write / Mailing labels to display the List of Mailing Labels screen.
3. Press Alt-F2 to pause the macro.
4. Press the down arrow key to highlight the address label that corresponds to the correct position on the physical label sheet.
5. Press Alt-F2 to end the pause.
6. Press Enter to select the highlighted label.
7. When the label appears, press F2 and F10. Q&A will display the Retrieve Spec.
8. Move to the new field, type in MAX, and press F10. Q&A will retrieve the record you were viewing when you began the macro and prompt you to press Enter to print the label for it.
9. End recording at this point and save the macro with video off.

Now, to quick-print a label for the record you're viewing, simply press Ctrl-L. Q&A will display the List of Mailing Labels and pause. Select the label that corresponds to the next position on the label sheet and place the label sheet in the manual feed slot.





labels (4 and 5) in one fell swoop. You'd expect to start with label 4. However, if you select the 4th Position Label and simply change the Number of copies to 2, you'll run into trouble because Label 4's Print offset value (40) will prevent Q&A from printing either label in the right position.

To print labels 4 and 5 in this case, select the 3rd Position Label (which has no offset value). At the Print Options screen, temporarily change the Number of copies to 2, and the Number of labels across to 2. Q&A will then print the two labels in positions 3 and 4, leaving label 5 intact to use the next time you need to print a single label.

If you're printing multiple labels starting at an odd-numbered position (position 1, 3 or 5, in this case), simply select the label that corresponds to the starting position on the sheet (see Figure 1) and temporarily change the Number of copies to the number of labels you want, and the Number of labels across to 2.

Tom Marcellus is editor of The Quick Answer, and the author of PC World Q&A Bible, published by IDG Books, and The Quick Answer Focus special report on monthly billing statements, published by Pinnacle Publishing.

How to Set the Number of Copies

Follow these rules if you're unsure about what to enter in the Number of copies field at the Mailing Label Print Options screen:

- If the label doesn't contain merge fields, set the Number of copies to the number of *identical* labels you want and be sure the Name of merge file at the bottom of the Print Options screen is blank.
- If the label contains merge fields, and you want to print one label for each record that meets your Retrieve Spec, set the Number of copies to 1. To print two identical labels for each selected record, set the Number of copies to 2.

QUICKTIP

Retrieve Records Where Two Address Fields Differ



If you can't retrieve the records you need with the usual parameters, try using a retrieval expression. Retrieval expressions give you greater record retrieval flexibility. Moreover, because they can reference field names, you can place them in any field at the Retrieve Spec—you don't have to move the cursor to the field on which you're retrieving.

Here's an example. Suppose you have a database of names and addresses. Some records contain two addresses: a mailing address and a business address. Other records contain a mailing address or a business address, but not both. If you wanted to retrieve only the records where the mailing address is different from the business address, you could use the following expression in any field at the Retrieve Spec:

```
{BusAddr <> "" and BusAddr <> MailAddr}
```

This expression retrieves the records where the business address (BusAddr) field isn't blank and differs from the mailing address (MailAddr). You've defined two retrieval parameters in one expression. By using other expressions in other fields, you can perform even more complex retrievals.

In the above example, I used field names. You can, however, temporarily number the pertinent fields and use those numbers in your expressions. This will save room if you have a lengthy expression.

If you create retrieval expressions you're likely to use again, don't forget to save your Spec by pressing Shift-F8 and assigning a name to it. This way, you won't have to retype the same expressions the next time.

Jerry Altshuler, Chicago, Illinois





Edited by Dave Reid

My Fonts Aren't Printing!



Q I use Q&A for DOS 4.0 with an HP LaserJet IID. When I try to print using fonts other than Courier, they don't come out, nor do they show up in Page Preview. I'm able to assign fonts by pressing Ctrl-F9, and even apply them to my document by pressing Shift-F6, but all I get out of the printer is the same old Courier. How come Q&A won't let me use other fonts?

*Sue Wagner
Troy, Michigan*

A It sounds like you're trying to use fonts that aren't available in your printer. For example, you may be assigning fonts from a font cartridge that isn't installed. Q&A 4.0 can only use fonts that are installed in your printer.

When you press Ctrl-F9 to assign fonts and choose the appropriate .FNT font file to use, Q&A displays a list of available fonts. These are the fonts that Q&A is capable of printing, but Q&A can print them only if they're installed in your printer. Hewlett Packard and other companies sell font cartridges that plug into your printer and allow it to print fonts other than Courier. Without such a cartridge, an HP LaserJet IID is limited to its internal fonts: Courier and Line Printer.

In addition to font cartridges, you can purchase and use what are called soft fonts. When you download these to your printer (they normally come with a download utility) and assign them to your document, Q&A can use them.

Don't expect Q&A's Page Preview to show exactly what a font will look like. Page Preview calculates the widths of characters based on the font, the size, and the style, and shows only an approximation of what the page will look like.

Create a Database to Automate Address Printing



Q My company types a large number of envelopes each month. We would like to set up a database to help automate what we're now doing manually with typewriters, but I don't even know where to start.

*Anne Hardman
Waterloo, Ontario, Canada*

A Creating a database in Q&A takes only a few moments. To get to the database design screen, select File / Design / Design a New File. Type in the name you want to assign to this file, such as ADDRESS, and press Enter. Create the database by typing the names of the fields followed by colons, leaving enough space for the information. If all you need are fields to store names and addresses, you can type the following field names:

First Name:	Last Name:	
Company:		
Address:		
City:	Prov:	Postal Code:

When you've finished typing the field names and colons, press F10 to save the database design and press F10 again when the Format Spec appears. If all your fields will contain text, you won't have to fill out the Format Spec. Otherwise, press F1 from the Format Spec to find out what codes to enter for number, money, date, and other types of fields.

To add records to the database, select Add Data from the File menu. Enter the information for each address and press F10 to save that record and get another blank form. When you've finished adding your last address, press Esc to return to the File menu. You can now use Search/Update to display your records and change them if you like.

You can print your addresses onto mailing labels or directly onto the envelopes. To create a mailing label, choose Mailing Labels from the Write menu. At the List of Mailing Labels screen, highlight the appropriate size label and press Enter. Q&A will walk you through the steps (you can press F1 for help at any time). To print individual envelopes, choose Type/Edit from the Write menu. Printing onto envelopes is more involved. See your Q&A manual for details.

Search and Replace Fonts



Q I created a lengthy Q&A Write for Windows document using Times Roman for the body and Arial for indented quotations. I decided I wanted a different font for the quotations, so I purchased and installed a new TrueType font. I now want to apply this font to all the text that's currently in Arial. I know that I can go through the document and manually apply the new font, but I'm wondering if there's an easier way.

*Dana Riley
Eugene, Oregon*





A There is. You can use Q&A Write for Windows' Find & Replace Formats feature. Move to the top of the document, click on the Edit menu, and choose Find & Replace Formats. The dialog box gives you options for finding and replacing even complex combinations of text and paragraph formatting, but your needs are straightforward.

Choose Character Formats from the Find menu. This dialog box lets you specify the format of the text you want to find. Choose Arial from the list of installed fonts. If, for example, you had 14-point Arial text you wanted to leave unmodified, you could select 12 from the list of point sizes. Q&A would then find only 12-point Arial text instead of all Arial text. After choosing any additional options, click OK.

Next, select Character Formats from the Replace menu. In this dialog box, you tell Q&A the format to which you want the specified text changed. Select your new font from the list of installed fonts. You can also specify the point size and style (bold, italics, and so forth). When you're finished, click OK, then click on the Find & Replace button. At each occurrence of Arial text, Q&A will prompt you to confirm that you want it changed. If you want all occurrences of Arial to be changed without confirmation, click on the Replace All button instead.

Repeat Duplicate Values in Reports



Q My department has three employees named Murphy, and they're all in a Q&A database we use. When I print reports from this database, Q&A prints the first Murphy's first and last names, but only the first names for the other two Murphys. How can we tell Q&A to print Murphy as many times as it occurs in the database?

*Frank Gill
Saugus, Massachusetts*

A You're probably sorting your report on the Last Name field. By default, Q&A prints only the first occurrence of each duplicate entry in sorted columns. You can turn this option off for that column by adding a comma and an R to the Column/Sort Spec for the Last Name field, like this:

`Last Name: 1,AS,R`

These column codes tell Q&A to make this the first column in the report, to sort it in ascending last name order, and to repeat (R) duplicate values. Do this, and Q&A will print the full names of all three Murphys.

More on Physically Sorting a Database



Q In the March 1994 issue, a tip ("Sort Your Database") explained how to physically sort a database. I followed the instructions, and the technique appeared to work. But now, each time I search my records, the one that was first the time before has been moved to the end of the database, and the one that was second the time before is at the beginning. Why don't the records stay put?

*Frank Gill
Saugus, Massachusetts*

A The technique explained in the tip involves copying the database design, then copying the records—in a specified sorted order—to the empty database. Though this puts all the records in the new file in the desired order, Q&A can't keep them that way permanently. When you change a record and then resave it, Q&A tries to put it back in the same position in the file. However, if the edited record has increased in size, Q&A may have to put it in the next available spot that's large enough. After performing a Copy Design / Copy Records procedure, there's no free space in the file except at the end, so your edited records wind up there. To have Q&A display your records in sorted order each time you view them, use the Sort Spec. You can display the Sort Spec by pressing F8 from any Retrieve Spec.

Dave Reid is a Symantec senior support analyst providing second-level assistance to technical support representatives. He's also the coauthor of The Q&A 4.0 Wiley Command Reference, published by John Wiley and Sons, and works as an independent Q&A consultant. PO Box 12083, Eugene, OR 97440.



Have a nagging question? Send it to **@Help, The Quick Answer, PO Box 888, Kent, WA 98035-0888** or fax to 206-251-5057.

When writing, please include your name, address, and phone number, along with your Q&A version number (and whether DOS or Windows) and a detailed description of the problem. We will publish those questions we feel are of general reader interest; individual responses are not possible.





Tweak Macro Files to Run “Impossible” Procedures

Need to switch between databases and procedures seamlessly? DOS batch files can add real horsepower to your Q&A macros.

By James Pogany



DAVID Dvorin’s article on accessing related databases via hot keys (*The Quick Answer*, September 1994) encouraged me to try his technique in my apartment building management application. This application consists of three databases: APTS (apartments), TENANTS, and RENTS. I wanted to hot-key from the APTS or TENANTS database to the linked records in the RENTS transaction database and back again.

Hyperlink worked fine as long as the RENTS database wasn’t too large. The APTS and TENANTS databases, at about 200 records each, were easy to link, but the RENTS database contained thousands of transaction records. The hyperlink technique took too long to find and display the linked records because the required @XLookupr Retrieve Spec expression in RENTS forced Q&A to scan every record in the database even though the field I was searching on was Speedy.

In contrast, when I typed the Retrieve Spec parameter (in my case, the Tenant ID number) into the RENTS Speedy field, Q&A retrieved the pertinent records instantly.

The Bell maneuver

This led me to experiment with Mike Bell’s technique described in the April 1993 issue. Mike uses a programming statement to create a macro-like string containing the variable (the Tenant ID number, in my case) in a field in the current record. The macro then copies this string to a temporary file and inserts it into the current macro file, replacing the previous macro.

Mike’s macro then invokes a DOS batch file to restart Q&A, which reloads the modified macro file, making the variable available to be pasted into a

Retrieve Spec. This technique, though powerful, didn’t solve my problem because exiting Q&A kills the macro. I needed the procedure to continue in order to paste the variable into the RENTS Retrieve Spec and display the linked records.

A hybrid technique

I drew on Mike’s approach by adding a field to the TENANTS and APTS databases and programming them to create a macro-like string containing the Tenant ID number. Like Mike, I used a macro to copy the contents of this field to a temporary file, to open the QAMACRO.ASC file in Write, to replace the previous macro with the macro from the temporary file, and to resave the file.

This is where I added a new aspect to the procedure. Before exiting Q&A, my macro creates an empty file named CONDN1.FIL. The batch file checks to see if CONDN1.FIL exists. If it does, it restarts Q&A with the -m1 parameter, which invokes the autostart Alt-1 macro. (See the sidebar above.)

Alt-1’s first action deletes the CONDN1.FIL file. It then goes to Search/Update for RENTS, pastes the previously saved Tenant ID number to the linked Speedy field at the RENTS Retrieve Spec, and “presses” F10 to display the linked records. The process is fast and reliable.

With a TENANT or APTS record displayed, programming in the database creates a macro-like string containing the Tenant ID and places it in a field in the record. When invoked, the macro performs the following steps:

1. Copies the Tenant ID macro string to a temporary file, saves or resaves the record, and exits the database.

Create an Autostart Macro

An autostart macro must be assigned an Alt-number key identifier such as Alt-1, and the -m1 parameter must be added to the DOS command that starts Q&A. For example, Q&A will run the Alt-1 macro as its first action if you start Q&A with the following parameter:

```
QA -m1
```





2. Opens QAMACRO.ASC in Write, searches for the macro with the previous Tenant ID, and replaces it with the new macro from the temporary file.
3. Resaves QAMACRO.ASC.
4. Creates an empty CONDN1.FIL file.
5. Exits Q&A, passing control back to the DOS batch file.
6. The batch file sees that the CONDN1.FIL exists and restarts Q&A with the -m1 parameter, passing control to the Alt-1 autostart macro.
7. Alt-1 deletes the CONDN1.FIL file, displays the Retrieve Spec for RENTS, calls the macro that pastes the Tenant ID number into the search field, and "presses" F10 to display the linked records.

How to Create a Macro That Exits Q&A

You can't record a macro that exits Q&A, because as soon as you exit Q&A the steps recorded up to that point are lost. (See page 107 in the *Q&A 4.0 Application Programming Tools Manual*.) To create a Q&A macro that exits Q&A, record all the steps of the macro up to the point when it arrives at the Q&A Main menu. Save the macro. Next, open the macro file in Write and insert *x<enter>* (for exit) as the last command in the macro just before the *<endef>* command. Resave the macro file with Ctrl-F8 and reload it into memory by pressing Shift-F2, selecting Get macros, and pressing Enter. The macro is now ready to be used.

AUTORUN.BAT works this way:

1. When you invoke AUTORUN.BAT initially, it loads Q&A with the default macro file (usually QAMACRO.ASC) but *without* an autostart macro parameter.
2. When you exit Q&A, control is passed back to AUTORUN.BAT, which continues where it left off. It looks for a file named CONDN1.FIL.

If it doesn't find CONDN1.FIL, it ends and leaves you at the DOS

prompt. If it finds CONDN1.FIL, it restarts Q&A with the Alt-1 autostart macro (the -m1 parameter in the batch file shown).

3. When you subsequently exit Q&A, the batch file again resumes where it left off. If it finds CONDN1.FIL, it loops back and repeats, reloading Q&A with the -m1 parameter.

By starting Q&A via AUTORUN.BAT, and by alternately creating and deleting the CONDN1.FIL file, you can retain macro control and continue the procedure.

A batch file with conditional commands

My technique requires that you initially start Q&A from a DOS batch file configured along the lines of this AUTORUN.BAT file:

```
echo off
c:\qa4\qa.com
:loop
if exist c:\qa4\condn1.fil c:\qa4\qa.com -m1
if exist c:\qa4\condn1.fil goto loop
end
```

Continues on page 13

Swap Macro Files and Application Configurations

This technique can be extended to change Q&A configurations. For example, you can swap QA.CFG (Q&A's configuration file) in addition to the macro file, and switch between a variety of custom applications and configurations. Or you can choose to run any of 10 autostart macros (Alt-0 through Alt-9) by controlling whether files named CONDN0.FIL through CONDN9.FIL exist. Here's an expanded version of AUTORUN.BAT that swaps macro and configuration files and restarts Q&A with one of two configurations:

```
echo off
c:\qa4\qa.com
:loop
if exist c:\qa4\condn1.fil copy macro1.asc qamacro.asc
if exist c:\qa4\condn1.fil copy qa1.cfg qa.cfg
if exist c:\qa4\condn1.fil c:\qa4\qa.com -m1
if exist c:\qa4\condn1.fil goto loop
if exist c:\qa4\condn2.fil copy macro2.asc qamacro.asc
if exist c:\qa4\condn2.fil copy qa2.cfg qa.cfg
if exist c:\qa4\condn2.fil c:\qa4\qa.com -m2
if exist c:\qa4\condn2.fil goto loop
end
```





Master Data-Entry Form Layout

In Q&A for DOS, the database you lay out on screen is the data-entry form you'll use day in and day out. Before designing a new form, take the time to plan ahead. You'll be glad you did.

By Tom Marcellus

DOS

THE first step in designing a database is to write down the items of information (data) you want the database to help you track. This will suggest the fields you'll need and their order on the data entry form. Talk to your co-workers and study the paper forms you'll use to enter your data. There's an economy aspect to database design. Unnecessary fields waste input time and disk space, but if you omit important fields, your database will be that much less useful.

Try to predict the longest values (entries) your fields will contain—especially name and address fields—and define your field lengths with these lengths in mind. It's better to have a little extra room for your data than not enough. Q&A makes it easy to add, delete, or move fields later, but it makes sense to get the form design as close to optimum as possible the first time.

Place your fields

To create a new database, choose Design a new file from the File / Design menu, type a name for the file, and press Enter to display the form layout screen.

You can place fields anywhere on the screen (you don't have to start at the upper left-hand corner), and you can place more than one field on a line. Keep in mind, though, that Q&A will treat as the first field the one that's closest to the upper left-hand corner of the screen. When you access the form for any task, including Search/Update, that's where the cursor will first appear, so you'll save keystrokes by making the fields you're likely to search on among the first ones on the form.

Field labels and information blanks

When you've positioned



the cursor where you want the field to begin, type a descriptive label for it—a word or two that describes the kind of information the field contains. (See the sidebar, "Make Field Labels Descriptive.")

Follow the field label with a colon (:), or a left angle bracket (<). These tell Q&A to start the information blank here (the part of the field that contains the data) and continue the field to the right edge of the screen, to the next field, or to a right angle bracket (>), whichever comes first.

When adding a field, you can leave spaces between the field label and the colon or left angle bracket, like this, to left align the information blanks:

```
First Name:
Last Name :
Address   :
City      :
State     :
Zip       :
Home Phone:
```

Or, you can right align your field labels, like this:

```
First Name:
Last Name:
Address:
City:
State:
Zip:
Home Phone:
```

Make Field Labels Descriptive

Make field labels meaningful. You know that PN means Phone Number, but someone else might think it means Part Number.

A field label can consist of several words, but don't overdo it. Fields labelled Home Phone and Work Phone are more sensible than Home Telephone Number and Work Telephone Number.

Unlike the colon, the left angle bracket won't show on the data entry screen and makes for a less cluttered form, especially if the form contains many fields. Another use for the left angle bracket is to define the beginning of a multiline field that can extend down the form for several lines. (More on this later.)





Fields without labels

Field labels aren't mandatory. You can add a field to your form that consists of nothing more than a left angle bracket or colon to begin the information blank and a right angle bracket to end it. Such labelless fields are useful for invoice line items, where you have rows of aligned fields for part numbers, descriptions, quantities, prices, and amounts.

A field defined solely by left and right angle brackets (no label) won't appear on the data entry screen until the cursor enters it, though it will behave like a field in every other respect. Q&A assigns an internal name to such fields so that you can still refer to them in programming statements, reports, and mail merge. (See the sidebar, "Assign Field Names to Labelless Fields.")

Ending a field

You can end a field's information blank in four ways—with a right angle bracket, another field label, screen text (see the sidebar, "Add Non-Field Text to Your Form"), or the rightmost edge of the screen. Fields with the label followed by a colon

Assign Field Names to Labelless Fields

If your form design includes fields without labels, Q&A assigns default field names to them. After designing your form, go to the Field Names Spec (File / Design a file / Program a file / Set Field Names). Labelless fields will be named F0001, F0002, and so on.

Such obscure field names will cause trouble if you use these fields in mail merge, so assign descriptive names to them even though they'll remain labelless on the data-entry form.

To overwrite a Q&A default field name, move to the field and type a new name over Q&A's default name. If you have four labelless part number fields on the form, for example, name them PartNo1, PartNo2, and so forth. When you're finished, press F10 to save the Spec.

and with no right angle bracket end-of-field delimiter can't extend onto the next line, whereas a field with a left angle bracket following the field label *can*, as long as no text or other obstruction is encountered. (See the section on multiline fields.) Table 1 illustrates ways you can set the length of a Customer Name field.

Multiline fields

If you need a memo or comments field for lengthy text entries, you have two options: you can design a multiline

field into the form or you can make the field any size and expand it during data entry.

A multiline field lets you display all the text in the field without having to move the cursor to the field to expand it. The disadvantage is that it takes up space on your form.

You can design a multiline field in two ways. Suppose you have a Product Notes field, and you've positioned the label at the left edge of the screen with several blank lines below it. If you follow the label with a colon and then place the right angle

Add Non-Field Text to Your Form

You can add plain (non-field) text to your form wherever you like. You can add a title at the top of your form, or add a few lines of instructions to assist the operator during data entry. But don't cram too much text into the form. For data entry instructions, you're better off creating custom help screens.

Table 1. Options for setting the length of a Customer Name field.

Form	Results
Customer Name:	With no end field delimiter, the field will extend to the right until it meets another field, any character, or the right edge of the screen.
Customer Name<	The field will extend to the right until it meets another field, character, or right angle bracket. The field may extend down several lines, but no further than the bottom of the current page. The < won't appear on the data-entry form.
Customer Name: ■■■■■■>	The field extends to the right angle bracket.
Customer Name< ■■■■■■>	The field extends to the right angle bracket. The angle brackets won't appear on the data entry form.
Customer Name: ■■■■■■■■■■ Phone:	The Customer Name field ends two character spaces before the Phone label.





Simplify and Reduce Programming with Indirection

Code glut got you down? In the November 1994 issue (“Use Your Programming Allowance More Effectively”), you learned how subroutines can help you avoid the risk of programming a database too heavily. Here’s how one Q&A developer tapped the hidden power of indirection and subroutines to make his programming more manageable and Q&A more responsive.

By Jeff Nitka



WHEN you program a database, Q&A verifies that the code can be executed when the database is used. When there’s a lot of code, this takes more time and memory. If there’s too much code, available memory can be exhausted and a ruinous system error can occur.

This happened to me while programming a Bill of Lading database. Users enter information about products being shipped, and the program returns information specific to particular products and customers. When the record is completed, its data is merged with a Write document whose merge fields are arranged according to the format of the shipper’s bill.

The application became so popular that I was continually asked to expand and add features to it, which increased the amount of programming until, finally, I reached Q&A’s limit and began receiving error messages. I needed to reprogram the database in order to reduce the quantity of code without sacrificing any features. Q&A 4.0’s indirection and subroutine capabilities provided the solution.

What’s indirection?

Groups of line-item fields in the database contained nearly identical programming, except that the code operated on different fields for different kinds of items to be shipped. The bulk of the code was in a series of item fields, where I needed 45 lines of code per field to perform the lookups and various calculations. This, multiplied by six items, plus the remaining code for the other 70-plus fields in the database, placed the database in jeopardy.

To solve this problem, I reprogrammed the database with subroutines to work in conjunction with *pointer* fields. They’re called pointers because they’re assigned field numbers of other fields during programming execution (in other words, they *point* to other fields). Pointer is another word for *indirection*.

For example, suppose you have a Name field that’s assigned field ID #1 in the Program Spec, and

a field named Ptr (for Pointer). If Ptr is assigned the value 1, it then points to field #1, the Name field. To retrieve the value in field #1, you can use the @Field indirection function. Thus, if the Name field (#1) contains *Jeff*, and Ptr contains 1, then @Field(Ptr) returns Jeff.

The beauty of indirection is that you can pass a field ID number to a pointer field, and your programming will reference that field instead of the original field. This opens the door to substantial programming economy because similar programs in like fields (item number fields in a line-item invoice database, for example) can perform their duties by referencing the pointer fields, which the subroutine then references to retrieve the data or perform the necessary calculations. This way, you generalize your code and use much less of it.

An illustration

Because my Bill of Lading database is highly

—BILL OF LADING—

Customer Name: #500 Purchase Order#: Address: Date: City/St/Zip: Invoice#:

Product1: #1	Lot1: #2	Amount1: #3
#Drums1: #4	Product Descr1: #5	
Product2: #6	Lot2: #7	Amount2: #8
#Drums2: #9	Product Descr2: #10	
Product3: #11	Lot3: #12	Amount3: #13
#Drums3: #14	Product Descr3: #15	
Product4: #16	Lot4: #17	Amount4: #18
#Drums4: #19	Product Descr4: #20	

Ptr1: #31 Ptr2: #32 Ptr3: #33 Ptr4: #34 Ptr5: #35 Gosub_Engaged: #36

Instruction: #37

POINTERS.DTF Program Spec Page 1 of 1

Esc-Exit F1-Help F2-Print F3-Clear Spec F6-Program editor F10-Continue

Figure 1. Sample database design shown here at the Program Spec with field ID numbers entered. Note the four Product fields and the Lot, Amount, #Drums (number of drums), and Product Descr (product description) fields associated with them. A database with groups of similarly programmed fields—such as this one—is an ideal candidate for subroutine programming.





specialized, I'll show you only enough of it to demonstrate the technique. The database could also represent a sales order or invoice database, where a series of line-item fields are filled by the user while related fields calculate and store data specific to each item. This technique is most useful when a lot of programming code in the database is used repeatedly in fields that have the same purpose.

In this sample database, the pointer fields (Ptr1 through Ptr5) are formatted for numbers because they'll hold field ID numbers (without the preceding # sign) as values. For an application such as this, the Gosub Engaged and pointer fields must be read-only to prevent user access.

During data entry, the user fills the Product, Lot, and Amount fields, and the Drums and Product Desc fields are calculated based on those entries. The amount of code in each Drums and Product Desc field was 45 and 10 lines, respectively. For six items, that amounted to 330 lines of code. Using indirection and subroutines, that code had to be written only once, greatly reducing the overall code in the database.

Programming for indirection

Let's look at programming execution for just one Product field because the same actions apply to the others. Here's the code for the fields related to Product1, along with fields #31 (Ptr1) and #32 (Ptr2), which contain subroutines:

```
>#1: Ptr1 = 1; Ptr2 = 2; Ptr3 = 3; Ptr4 = 4; Ptr5 = 5;
    Gosub #32; If #1 = "" then Cend else goto #2

>#2: If #2 = "" then goto #2 else { Gosub #32; goto #3 }

>#3: If #3 = "" then goto #3 else Gosub #32

<#32: If @Field( Ptr1 ) = "" then { Clear( @Field(Ptr2),
    @Field(Ptr3), @Field(Ptr4), @Field(Ptr5) ); Return }
    else If @Field(Ptr2) = "" or @Field(Ptr3) = 0 then
        Return
    else {Gosub Engaged = "T"; Gosub #31;
        Gosub Engaged = "F"; Return }

<#31: If Gosub Engaged = "T" then { If @Mod( @Field( Ptr3 ),
    55 ) = 0 and @Left( @Field( Ptr2 ), 1 ) = "Y" then
    @Field( Ptr4 )= @Str( @Field(Ptr3) / 55 ) + "X 55 LBS"
    else If @Mod( @Field(Ptr3 ), 110 ) = 0 and @Left
    ( @Field( Ptr2 ), 1 ) = "6" then @Field( PTR4 ) =
    @Str( @Field(Ptr3) / 110 ) + " X 110 LBS"
    else If @Mod( @Field( Ptr3 ), 50 ) = 0 then
    @Field( Ptr4 ) = @Str(@Field( Ptr3 ) / 50 ) +
    "X 50 LBS" else @Msg ("This is an unacceptable entry");
    @Field( Ptr5 ) = "TRADENAME " + @Field( Ptr1 ); Return }
    else Cend
```

What the programming does

When the cursor exits field #1 (Product1), the pointer fields (Ptr1 through Ptr5) are assigned the field ID numbers of the fields that pertain to the Product1 field (Lot1, Amount1, Drums1, and Product Descr1) and programming control is transferred to field #32 (the Gosub field in this example.)

Advantages of Indirection and Subroutines

- Indirection and subroutines save Q&A time and memory when checking a Program Spec and when loading it in preparation for data entry. For a small database with little code, this isn't an advantage. For a heavily programmed database, it can mean the difference between an acceptable Program Spec and a system error that can ruin the database.
- Manually typing lengthy blocks of similar code for several fields is a tedious and error-prone process. Using indirection and subroutines saves time and reduces errors.
- Indirection and subroutines make your programming more general. For example, if I had to edit my code to plug in more calculations for the Drums fields, I'd have to change only one field's (field #31) program. Without indirection, I'd have to change the code in every Drums field.

Field #32's subroutine, which calculates the number of drums based on the related lot and amount, first checks to see if the Product1 field is empty. If it is, then the fields that pertain to that product are cleared, giving the user the option of deleting all the information for that product simply by deleting the Product1 entry. If the Product1 field contains an entry, the Gosub #32 program is engaged as long as the related fields (Lot1 and Amount1) contain entries.

Field #31's subroutine contains rules regarding packaging requirements for specific lots and amounts. (These rules are provided for illustration. Your programming will reflect your own rules.)

Field #31's first If statement tests for lots beginning with a Y and evenly divisible by 55. The second If statement tests for lots beginning with a 6 and evenly divisible by 110. The third If statement tests for an amount evenly divisible by 50. If all tests fail, the user is notified that the entry is unacceptable (the amount or lot isn't in the standard line).

If any of the above conditions are satisfied, then the number of drums, with the appropriate weight per drum, is returned to the related Drums field. The related Product Descr field, @Field(Ptr5), is then determined.

The code in the other fields provides a check against data entry errors. If any Product field isn't





blank, the cursor is moved to its corresponding Lot field to prevent vital information from being accidentally omitted in a record. It's not possible to leave a Lot or Amount field blank if the related Product field has a value. This ensures that Drums will be correctly calculated.

Also, notice that Gosub #32 is engaged after exiting the Lot, Amount, and Product fields. This allows the user to change information and have the Drums field automatically updated.

Gosub_Engaged allows the subroutine to be executed when necessary (when Gosub_Engaged = "T", or True), but execution is prevented when Gosub_Engaged is assigned "F" after a Gosub is finished. (This additional code isn't essential. The only requirement is that program control has to pass to the subroutine field if any changes are made to dependent fields. This way, the subroutine can validate or reject the change.)

Code savings realized

Each Product field has a related Drums field to be calculated (see Figure 1). The code required to perform the calculation for additional products is the same, except that the Ptr fields are assigned field numbers that pertain to those different products. For example, for Product3, Ptr1 is assigned 11, Ptr2 is assigned 12, Ptr3 is assigned 13, Ptr4 is assigned 14, and Ptr5 is assigned 15. Only the programs that check for blank fields are different—the Gosub field's subroutine remains the same.

This technique can be applied to any database where similar calculations have to be made for several items. There are no disadvantages to it other than the time it takes to add the pointer fields and reprogram the database. If each field's code is relatively brief, then indirection, while certainly workable, isn't necessary.

Using this technique, I saved about 220 lines of code (more than 13K) at the small expense of adding a few pointer fields. I also had five other fields in the database where indirection was appropriate. Altogether, I reduced the amount of code by half. To the delight of my co-workers, this allowed me to add even more productive features and bells and whistles to the database.

Jeff Nitka is pursuing a mathematics and computer science degree at Rutgers University in New Jersey. He works part time as a software developer. Epoch Software, 908-874-3989.

[The SALESORD.DTF database that comes with Q&A for DOS 4.0 is another good example of how subroutines and indirection help you simplify and generalize your programming to reduce the quantity of code needed to achieve the same results. SALESORD, a sales order, contains three rows of line item fields, headed Part No, Qty, Unit, Description, Per Unit, and Price. The programming for these fields utilizes indirection and Gosub commands to look up the information for each Part No entered. See the January 1992 issue ("Gosub and @Field in Action") for details on how the indirection programming in SALESORD works.—Ed.]



Pinnacle Publishing, Inc.
PO Box 888
Kent, WA 98035-0888
206-251-1900
206-251-5057 (fax)

Second-Class
Postage Paid at
Kent, WA

Printed in the U.S.A. on recycled paper.

