## Sesame Seeds

# Sesame, Memory and Speed

**MARK LASERSOHN**

**Y**OU could have knocked me over with a feather. I've been a professional software engineer for 18 years. I've built software for everything from robots the size of Volkswagons to bingo parlor flashboards to fighter jet control systems. But when Bill Halpern (head honcho around Lantica) said, ". . . and, by the way, can you make it as fast as Q&A?" I nearly cried.

Let's take a look at that requirement. Q&A has been optimized for performance for 15 years, probably tens of thousands of man-hours. Q&A runs on a single platform and is written to run right on the metal. Q&A never needs to run true client/server. Q&A has a simple text based interface (very low overhead). Q&A has the full attention of the CPU all the time. Q&A has "speedy fields," inflexible indexing, internally only supports one field type, and isn't relational or hierarchical. Wow.

*Sesame*, on the other hand, has to be graphical, has to be true client/server, has to support relational, has to support multiple operating systems (and share resources), isn't pre-indexed, supports over a dozen field types, and must be able to run on a Web server. Again—wow.

Well, it took three years, the help of a lot of friends, and more late nights than my wife can count, but I can finally answer Bill's question.

### A question of RAM

The answer is: "sort of." *Sesame* is as fast as Q&A—or faster—as long as you give it what it needs: RAM. *Sesame* uses RAM whenever possible. It uses RAM to hold the application file. It uses RAM to hold the bookkeeping we do on the application. It uses RAM to hold everything it displays. It uses RAM to run client/server. If given a choice, *Sesame* chooses RAM. Nine out of ten *Sesames* recommend RAM for all your data storage needs.

Why RAM? RAM is fast. RAM is cheap. RAM can grow as your database grows. Right now I can buy a single 256Mb module for my Gateway from a reputable dealer (Kingston) for $86. For around $300, I can have 1 Gigabyte of RAM in my otherwise ordinary PC. Most database systems use the hard drive where we use RAM. This gives them an advantage on *very* large databases. The performance will be approximately the same for a 4 Mb database as for a 400 Mb database. The same can be said for the computer itself. A 128Mb computer will run the 60Mb database at the same speed (more or less) as the 640Mb computer. The only way to really increase the speed, in either case, is to upgrade the hard drive to a faster model, and that will only marginally affect performance. And the performance will only get worse over time.

### What's different about *Sesame*?

*Sesame* on a 128Mb computer will perform most operations on a 30Mb database as fast as or faster than Q&A. But will be considerably slower than Q&A on a 90Mb database. Upgrade the memory to 256Mb, and *Sesame* catches right back up. Why is that? *Sesame*, like all Windows/Unix programs begins to swap when the RAM runs out. Swapping is what an operating system does when it needs to run a program that uses more memory than is available in RAM. It "swaps" some of the data that is in RAM, writing it to a file on the hard drive. It then takes some of the data that it already wrote to the hard drive and puts it back in RAM—where it is available to the program that needs it. The part of the hard drive it writes to is often called "Virtual Memory." A computer might have 128Mb of RAM, but can use up to 2Gb of virtual memory.

Why does that slow down *Sesame*? Swapping is slow. Swapping is caused when a program tries to access a memory address it does not have in RAM (a "Page Fault"). If the program is simply running through an absolutely sequential list, it will page fault every time it reaches the size of a page of

## Quick Tip: A Database Q&A Can't Open Might Be Okay After All

Many of us who use John Dow's DTFDOCTR to diagnose problems with a database have had an occasion where DTFDOCTR says that a database is perfectly healthy even though Q&A can't even open it.

Well, yesterday, after *PCAnywhere* induced a semi-crash, my 24MB working database, open in the background, earned a system error and Q&A was shut down. When I went back in and tried to open the database, Q&A reported that it was "*incompatible with this version of Q&A*," even though DTFDOCTR said that the database was just fine.

I was really upset because I had just finished about three hours of work in this database. Here's what I found.

DTFDOCTR was looking only at the .DTF file, not the .IDX file. All was reported to be fine (and in fact was). The crash had damaged the .IDX file which Q&A also opens whenever you open a database. Q&A said: *NO GOOD!*

I replaced the .IDX file with a copy from yesterday's backup, removed and replaced the Speedy fields, and then recovered the database. (I could not do this prior to replacing the .IDX file) Viola!—everything was back exactly as it had been prior to the crash.

The morals of the story are:

- Your database crashes. DTFDOCTR says it's fine but Q&A disagrees. Don't give up. Replace the .IDX file with a similar (not even necessarily from the same database) .IDX file and do the Speedy fields/Recover routine.

- Your database is reported to be fine by DTFDOCTR but is running not quite right, or continually crashing. Again, concentrate on rebuilding the .IDX portion of the file.

—**Bill Halpern**

## Letters

I read Michael Burns article ("Importing Huge ASCII Data Files," April 2003) with some surprise. As early as 10 years ago, on very slow PCs, I was importing huge ASCII data files into Q&A with no trouble—*unless* I was importing into a database that had Speedy fields. *Then* the import got slower and slower and would have taken days to finish. I am willing to bet that a Speedy field was the root of Michael's problem, and that all his work-arounds were unnecessary. Import first, *then* create the Speedy fields.  —*John Horn*

John Horn and I have been exchanging info on other related matters but I told him I had no Speedy fields. As he and I work together, I am sure we will find out more on the matter and will transmit same to you when we have something definitive.  —*Michael Berns*

I could have sworn Speedy fields were his problem, since I had the same problem years ago. But he says no. Each week I import a Standard ASCII data file with more than 700,000 records and about 80 fields, and it only takes about 45 minutes. My PC is new —1.6 GHz—but even on my old PC—one-third the speed—it took less than two hours. A couple of years ago I was doing this job on an older laptop and the performance was awful. (I understand laptops are generally slower.} Could the "Intelligent Assistant" be the culprit? I seem to remember a long time ago I had a database where the IA was somehow turned on and that caused some sort of a problem.  —*John Horn*

*"Teaching" the IA or Query Guide in a database does index fields even though they don't show up in the Speed-Up Spec as Speedies. My guess would be that it would cause a similar slowdown effect when importing a huge data file into such a "taught" database. If the issue is important to you, it should be easy enough to setup and run some informal tests on two copies of the same database—one taught, the other one untaught.  —Ed.*

# Sesame Speed cont'd from page 1

memory. That would be optimal swapping. Unfortunately, rarely is data arranged in an absolutely sequential list. So the CPU is usually forced to hop around in memory, causing frequent page faults and frequent swapping. And worse, because the system will swap a whole page at a time, it may be reading and writing much more data to the hard drive than is actually being used by the program.

## Swap in more memory

Okay, okay already, swapping is bad. So how can it be avoided? The simplest way to avoid swapping is to keep your data small. But given that that is not always practical, the next best thing is to size your RAM with an eye towards your database's requirements. Determining those requirements is pretty easy, once you get used to some of the terms (like "Page Fault" or "Virtual Memory"). Here, Windows NT/2000/XP can really be a life saver. If you hit "Ctrl-Alt-Del" the NT series will bring up a dialog box. On that dialog is a button marked, "Task Manager." Click on it.

Task Manager has three tabs along the top row. The first, "Applications," gives you a list of currently running applications. The second, "Processes," is a table showing the processes running on your system and various system metrics pertinent to each. The last tab, "Performance," (see Figure 1), shows two graphs. The upper graph shows CPU activity. The lower graph is total memory usage. That includes both RAM (also called "Physical Memory") and Virtual Memory. This image was a snapshot of my

computer just after reboot, so the minimal amount of memory is being used. In my case, running Windows 2000 with 128Mb of RAM, I have used 68Mb (or so) simply for the operating system—about half. That should leave me with about 62Mb to run a *Sesame* database.

Does that mean that I can now load a 62Mb *Sesame* .db /.dat file combination? No, bad news: *Sesame* itself is going to want about 4Mb of that just to load the executable. And a database file is highly compacted compared to a database that is actually up and running. So let's load up the oft tested "AMSearch." Amsearch.db is 49,452 bytes and AMsearch.dat is 162,307 bytes—for a total of 211,759 bytes. When loaded into *Sesame*, the *Sesame* process goes from just below 4Mb to just over 5Mb. Where did it get the additional megabyte? That would be the "True" or running size of the AMSearch application. That is the total of all of its data, plus all of the memory used by *Sesame* to organize, operate on, and display that data. Most databases will not, of course, quadruple in size when running. But all databases will grow. The ratio is based on the number of records, the amount of data an average record holds, and the graphical requirements of the forms. A database with a large number of records, each with lots of small empty fields, will have a very high overhead ratio. A database with very few records, each with only a few, but very full fields, will have a very low overhead ratio.

## Are we swapping yet?

So let's assume you have been using your *Sesame* database for a few years now and the performance has suddenly dropped off. How can you tell if that was due to swapping? The fastest way to tell is to run a search with a sort. If your hard drive light is on and the hard drive sounds like it is grinding a fresh cup of java, then it is very likely to be swapping.

If that isn't scientific enough for you or you want to make it stop—let's go back to the Windows Task Manager and attach some hard numbers to that grinding noise. (See Figure 2 on following page.)

Here we can see the full task list of all of the processes running on my 128Mb computer. If you look carefully, you will see *Sesame* listed near the middle of the page. Under the column "Memory Usage" you will see that Sesame was using 60Mb when this snapshot was taken. Further to the right, you will see that its Peak Memory usage was 98Mb. Even further to the right are its Page Faults, counted out to 874,356. That would be quite a few. Next to that is VM Size. VM Size is the amount of virtual memory used by *Sesame*. Overall, these numbers do seem to indicate that *Sesame* was swapping, mercilessly.

The thing to look for is the unusually high ratio between currently used memory and VM Size. Figure 3 on the following page shows a similar snapshot taken on a computer with 512Mb of RAM.

This was taken during an identical operation on the same database. Here we can see that the current Memory
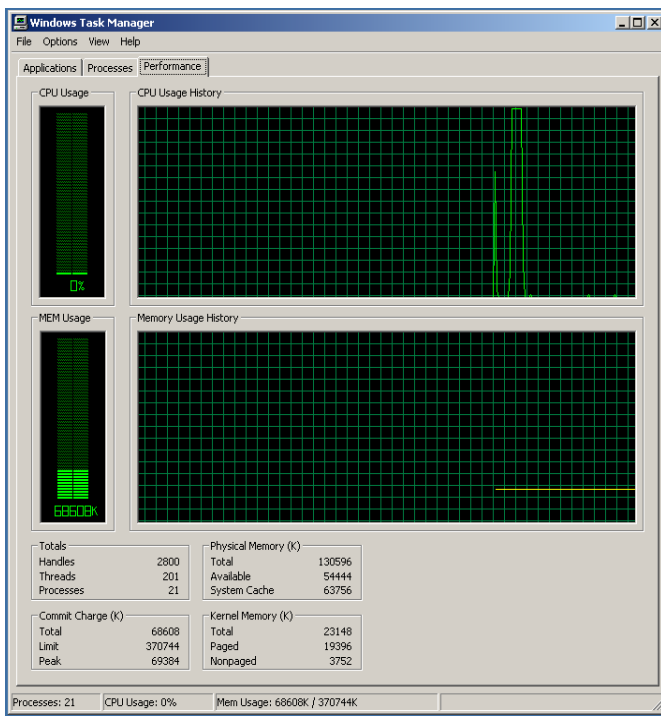


**Figure 1.** Task Manager's Performance tab.

usage of 231Mb is actually higher than the VM Size. Also note the much smaller number of Page Faults. On this computer *Sesame* is not swapping. It fits.

To give you some idea of the speed difference between these two based on amount of RAM alone: Both computers used the same 330,000 record database. Both performed the same search/sort. The search returns a result set with 22,389 records. The sort was designed to make sure there are many ties and thus is forced to use the second and third sort parameters. The first two sort fields were numeric, the third was text.

**128 Mb Machine: 1 Minute and 15 seconds.**

**512 Mb Machine: 11 seconds.**

**Q&A on Win2000 (128Mb): 17 seconds (to get the first record)**

**Q&A on XP (512 Mb): 25 seconds (to get the first record)**

The second number for Q&A on XP confuses me too! But I make it a policy not to look too hard at problems 18 year old DOS programs have on one-year-old operating systems. So long as they run at all, I count my lucky stars and praise the designers.

**Penny Pinching**

Given an informal survey, I've found that swapping will not be a problem for most of the folks switching from Q&A to *Sesame*. Given that the average Q&A database is well below 30Mb and that the average PC these days is running well above 128Mb—the advantages of running in RAM (speed, data integrity, data flexibility) outweigh the potential price (swapping). But let's say that you are one of those unfortunates with a fair chunk-o-data and a puny little computer, and your spouse, boss, business partner refuses to shell out for even one RAM upgrade. What can you do for *Sesame*, and what can *Sesame* do for you?

First, let's be reasonable in our expectations. *Sesame* can't help but swap if you are trying to run a moderately sized *Sesame* database *and* your entire office suite, and your web browser, and Outlook, and Q&A, amd a movie you downloaded, and Paintshop, and sort through a few Gigs of clip art, and the fancy new screen saver that makes a cartoon dog chew up your spreadsheet... all at once. If you can actually hear your computer gasping for breath, its too late—you *are* going to swap. All of your applications are going to swap. The operating system is going to make a lot of decisions for you about what gets to run and
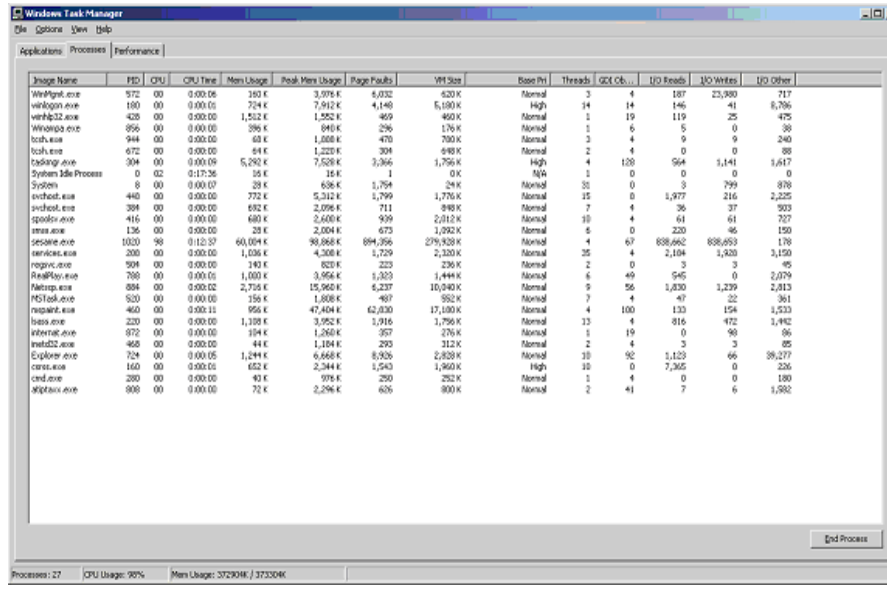


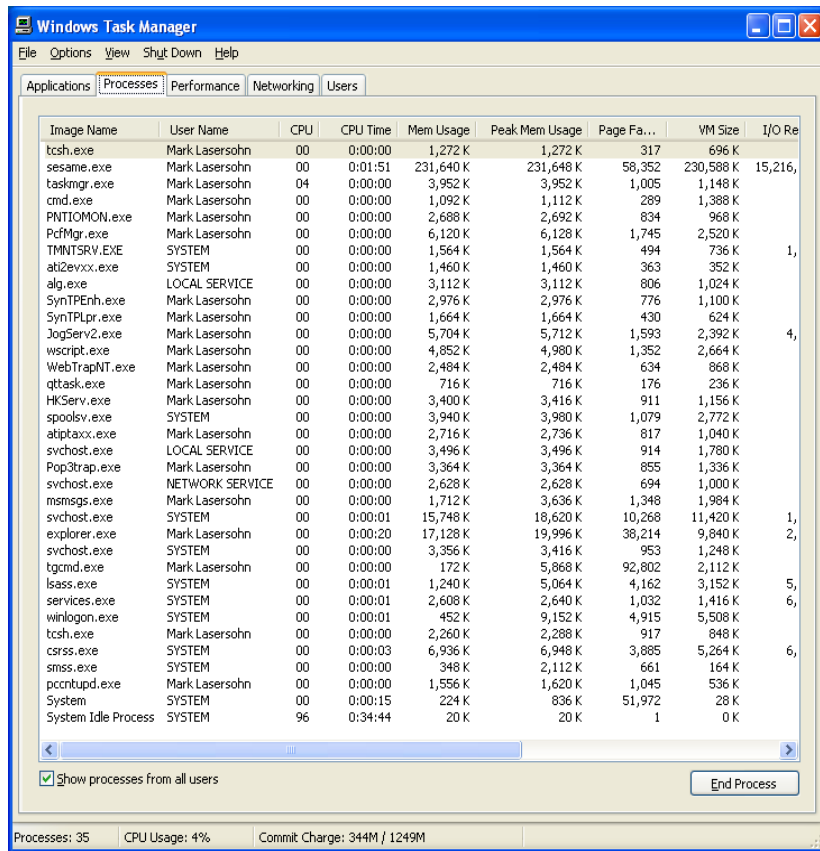**Figure 2.** Running Processes shown in Windows Task Manager on a 128Mb RAM PC.



**Figure 2.** Running Processes shown in Windows Task Manager on a 512Mb RAM PC.

when. In this worst case scenario, *Sesame* can do one thing for you: by swapping, it frees up your CPU and lets the little doggie chew. My advice: take control. Decide what you are really doing. Bring up the applications that are needed for that task, and shutdown, untray, or at least minimize, everything else.

## Wasteful design elements

In Q&A there are no variables. That means that if you want to use a value during programming you have put extra fields on your form to hold those values. In *Sesame* these are unnecessary, and waste a tremendous amount of space. I have in my possession about fifty Q&A test databases. Some of these (most) come from real-life small businesses. Of those, something like 20% of the total storage and overhead is being used up by fields that only exist for use in Q&A Basic programs.

 *Sesame*, in contrast, provides variables in programming. These use no permanent storage and only exist while programming is running. And even while they do exist, they have much lower overhead per value than do permanent fields. For example, if you have to calculate the sum of two fields on a field entry event, *Sesame* creates a memory location for that sum value when the on-field-enter program starts to run. It then frees that memory when the on-field-enter program is done. If you use a field for that same purpose, memory must be allocated for that field in every record in your database, and it must keep that memory allocated all the while *Sesame* is running that application—fiercely wasteful.

 It is easy to convert a Q&A database over to *Sesame*, then in the *Sesame* Designer change these fields into variables. Simply, remove the fields from the form using the delete key, then in the programming editor, run "Test." Test will tell you where you have "unknown identifiers." If you click on the error it will even navigate right to the line where the used-to-be-a-field-but-ought-be-a-variable resides. You can then declare your variable on a line before its first use:

```
VAR SUM_VALUE AS INT
SUM_VALUE = COOKIES_EATEN + COOKIES_COOKED
```

 By using variables instead of fields you will save field overhead times the number of records in your database. This alone will bring most swapping databases under the line.

 Another wasteful practice that cannot be helped in Q&A, but is easily remedied in *Sesame*, is the use of fields as buttons to kick off programming or macros. *Sesame* allows you to declare push buttons that exist merely for the purpose of kicking off a program or macro. Again, going into *Sesame* Designer and eliminating these fields and replacing them with buttons is easy to do, and will save you memory for every record in your database.

 Q&A has the @*Modified* flag to indicate if any part of the record has been changed. It cannot, however, tell you if this field or that has been changed, or for that matter

retain the original value without your allocating a "shadow field" on the form. In *Sesame* these hidden fields are completely unnecessary and can be deleted. *Sesame's* event model allows you to determine if any particular field has been altered through the use of the *On Field Modified* event. And variables can be used to "shadow" the original value. In addition SBasic has a function called "NotifyForm" that will prevent the user from advancing a form with incorrect values. Using these features in combination eliminates the need for any "shadow fields" and can potentially save an immense amount of storage space.

 All of the previous recommendations fall under the category of generally good advice. From here on in, the advice gets increasingly more dicey and a lot less applicable to the general user. So while these changes may save you from the grief and heartbreak of swapping, they should only be applied as a last resort.

## Rarely-used fields

So let's say you start using variables instead of fields, you haven't talked the boss into a RAM upgrade, you sent the doggie screen saver to the dustbin, and your *Sesame* application is still swapping. What then? The next thing to look at is the structure of your data. Q&A is a flat database, so if you want to know something you have no choice—you have to put a field for it on your form. Even if it is rarely relevant to the record in question, that field sits there eating up resources, empty. So how can *Sesame* help? *Sesame* is a hierarchal database. If you have a set of fields (let's say fifteen of them) that are rarely used, but are sometimes necessary—put them on a subform. That way they only exist if you use them and thereby only have the overhead of one field rather than fifteen when they're empty, but are still available for use when you need them. There are a couple of minor downsides to this approach. If you use any one of the fields you will get the overhead of all of the fields on the subform. Also, a small bit of programming would be necessary to prevent users from advancing the subform and entering more than one record (if that is not your intent). But since you will not be using up all of the overhead of all of the fields on every record, the savings can be quite high.

## XLookups

Many Q&A databases use XLookups to achieve a kind of relational structure. Using XLookup this way allows each Q&A database to act as a table in a larger application. *Sesame*, being code-compatible with Q&A, also supports XLookups with precisely the same arguments used in Q&A. All well and good—unless you are swapping. To achieve good speed on subsequent XLookup operations, *Sesame* will "load" the external database on the first XLookup call. For most applications, this is appropriate and lets XLookup run nearly as fast as relational subrecords. But for a database with too little RAM

# @Help

EDITED BY WILLIAM HALPERN

## More on Q&A, XP and Novell

**We converted to Windows XP from Windows ME and suddenly Q&A won't open our database because it says it cannot open the index file. Other systems running in XP work fine. Any assistance would be appreciated.**

Darrell

**We are using Q&A 3.0 and have run into a problem. We have a new Windows XP Professional workstation to install on a Novell 4.11/4.2 network. When the Windows client for Novell from Microsoft is installed, I can get into Q&A with no problem, but cannot get into the e-mail server because it requires Novell's Client32 software. When I install the Novell Client32 software, I get a Q&A error that I cannot access "qaovr19.ovl" and therefore am not able to use Q&A. We do not have this problem with Windows 98 workstations running the Novell Client32 software. Do the Q&A 4.0 for Windows or Q&A 5.0 for DOS versions have this problem? The firm is looking to upgrade our current version of Q&A.**

Kevin

Here are two issues that involve Windows XP, Q&A and Novell. As I reported in the past, a fix was released for the Novell client 4.82 in Windows 2000 or XP to solve the can't-open-a-required-file problem (not only with Q&A). Since our first reports, there has been a complete service pack 1 and now a service pack 2 with follow-on patches. We do not use Novell enough to be able to give you any real advice as to how effective these fixes are in solving the problems. What I *can* tell you is that our company is getting more database repair jobs from Novell systems than ever. Again, the problem is not a Q&A one. It is in the way Novell handles file and record locking and write-behind caching. We have also seen corruption caused by the way Novell manages the volume sizes for the end user. In every case, a test repair which involved moving the database off the Novell server has solved the problems. At this time, all I can recommend to Q&A users is to migrate away from any version of Novell to a Windows-based server. Windows 2000 is my preference.

## Q&A Suddenly Slows Way Down

**Q&A is running on our new computers just "Okay." Very slow sometimes in searches, frequent "network busy, please wait" messages. Sometimes, Q&A prints a report line by line (you can watch it as it says "*Page 2 Line 23 Line 24 Line 25...*") and this is for only *two* users with Q&A running locally on each computer and the data stored on the "Q" drive of the dedicated "fileserver" in a peer-to-peer network. Windows 2000 Pro is on the "fileserver" and both workstations (about to become *three* workstations).**

Andy

This seems to be an issue with Norton Antivirus. The clue was when Andy said that it only happened with two or more users in the database. This is the same symptom for Q&A slowing down in Windows 98 with Norton 2000 Antivirus. To resolve the problem, we went into Andy's Norton Antivirus setup options and excluded *.IDX and *.DTF files from automatic checking. The system then sped up. We will watch for further progress/problems over the next few weeks.

## Budget PC's May Lack Expanded Memory

**I have been using Q&A 5.0 for years and I just bought a new Dell 2350 computer. I can't get it to give Q&A any Expanded Memory. Do you have any suggestions or the brand and model of a computer I can purchase.**

Steve

The 2350 is Dell's entry level model. It has *shared* memory, where the video card uses the on-board system memory instead of its own. The memory the video card uses is the same memory that Q&A wants as Expanded Memory. This is how computer makers keep the cost of their budget systems low. You need a model that does *not* use shared memory, such as the Dell 4500 series. A bit more pricey, but you'll get Q&A to work and get much better graphics.

## Programmable Report Headers

**Is it possible to make Q&A 4.0 for DOS' built-in functions, other than @Date and @Time, work in a report header? I prepare a series of reports each month based on data as of the first of that month and I need the header to say, for example, April 1, 2003, regardless of the day of the month I prepare it. I've tried using** *(@Month$(@Date) + 1, +@Year(@Date))* **in all sorts of variations without success. @Date prints the current date, but everything prints as typed (as literal text).**

AC

The bad news is that there is no way to do what you want in Q&A 4.0 The good news is that it is easy in version 5.0

You have to add a derived column to your report that looks like this:

```
Heading: First of Month for header
Formula No. 1  @Month$(@Date) + " 01, " +
@Str(@Year(@Date))
Column/Sort Spec: 2,I
```

Note that this is derived column #1 (the first one on the first Derived Columns page) and it is set to be invisible (*2,I*).

In one of your report header lines at the Define Page screen,  type:

```
!Report Printed for *PG {?1}*
```

Here, the number after the "?" refers to the number of the *derived column* ("1" in this case), not the report column.

For a report printed anytime during the month of April 2003, this will give your report a header that reads:

```
Report Printed for April 01, 2003
```

So, either you upgrade to Q&A 5.0, or you manually update your report's header each time you run the report.

## Sort a Report by One of a Group of Fields

**I need to print a report of completed jobs by their completion date. Sounds simple but here's the catch: I do not want to redesign my database. A job is completed when it is Mailed, E-Mailed, Faxed, or Posted to the Internet, or some combination of these. Each of these dates is placed in a separate field in the database. I want to sort the report by the completed date. Is there any way that you can set up a report to look at multiple fields, pick the one that is filled and then sort by it?**

Kristen

Seems tricky, but it is really easy if these are all date fields. You can use a Q&A function and its ability to do date

math to accomplish what is needed. Here's the report's Column/Sort Spec:

```
Mailed date: 2,I
Email Date: 3,I
Fax Date: 4,I
Internet Date: 5,I
```

Here's the derived column:

```
Heading: 10:Completed
Formula No. 1  @Min(#2..#5)
Column/Sort Spec: 10, F(D16), AS, R
```

If these are true date fields, Q&A will look at all of the fields and pick the one with the earliest date (not blank). It will then use the result as a real field on which the rest of the report can be sorted. The report, when run, will do multiple passes on the database, first to calculate and then to sort, but the result will be what you are looking for.

## Not a Crosstab, Not a Totals-Only Report, But…

**I need to design a very special kind of report  that is part Totals only, part columnar, and part crosstab. I have a feeling that it can be done in Q&A, but I don't know how. I use Q&A 5.0 and any help will be greatly appreciated by both me and the boss. The report is a comparison of work done by the various teams in the Company and needs to look like this spreadsheet that I put together every week by doing manual retrieves in the Q&A database and using pencil and paper to write down numbers of records and then put them in Excel.**

| Team | Total Rec'd | Received Not Rev'd | Reviewed Not Mailed | Received Not Mailed | Received Not Typed | Typed Not Rev'd |
|-------|-------|------------|------------|------------|----------|----------|
| 1 | 65 | 16 | 5 | 19 | 6 | 13 |
| 2 | 98 | 41 | 13 | 38 | 30 | 13 |
| 3 | 36 | 12 | 4 | 16 | 7 | 7 |

**Each record has the following fields:**

```
Team Number
Date Received
Date Reviewed
Date Mailed
Date Typed
```

Kristen

Tell your boss he owes me big for the solution to this one! You have a pretty good start by stating what you need:

1. A Totals-Only report which is not really totals only but only prints one line for each team's totals.

2. A Crosstab report for columns that don't exist in the database.

# Freeform Reports in Q&A for DOS

## TOM MARCELLUS

**Y**OU can actually get quite serviceable freeform reports out of Q&A for DOS without going through the complicated process of setting up a detailed Print Spec.

I used this approach a while back when I designed a little database to store information on Q&A's and *Sesame's* built in functions. This database (BIFS.DTF) has fields for function name, abbreviation, type (string, financial, context-specific, etc.), arguments, return value, and whether it's a new *Sesame*-only function. It also has two multi-line fields, one containing a brief description of the function and another for a programming usage example. Figure 1 shows all the fields (screen pages 1 and 2) in a sample record.

To support a recent *Sesame* beta release, I needed to output this database to a report format—one that would list all the functions in alphabetical order as shown in the opposite column.
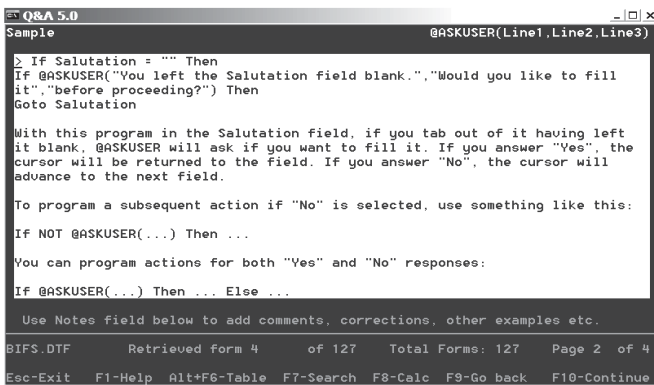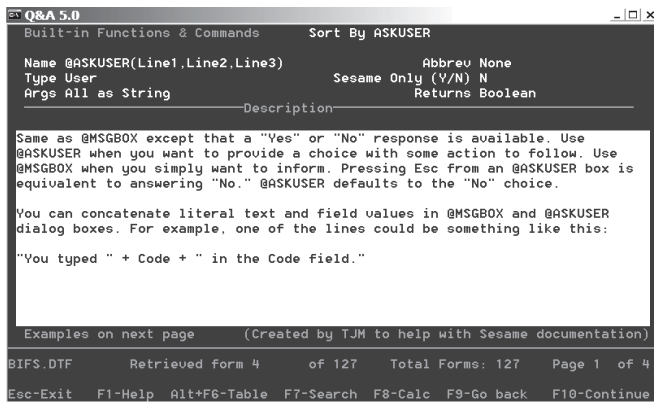




**Figure 1.** Sample record from BIFS.DTF (screen pages 1 and 2)

```
@ABS(n)
@AB(n)
Type: Mathematical
Args: n as Double
Returns: Double

Description:
Returns the absolute value of n. The absolute value of a
number is the number whether it's negative or positive. So
the absolute value of -20 is 20.

Example:
AbsVal = @ABS(Temperature)

If Temperature contains 50 or -50, AbsVal will be set to
"50".

++++++++++++++++++++++++++++++++++++++++++++++++++++++

@ADD
@AD
Type: Context
Args: None
Returns: Boolean

Description:
Limits programming execution to only when adding new
records. Typically used in a conditional statement like
this:

If @ADD Then
{
..statements..
}

Example:
If @ADD Then EntryDate = @DATE

When adding a new record, EntryDate will be set to the
current date.

++++++++++++++++++++++++++++++++++++++++++++++++++++++

@ASC(x)
@AS(x)
Type: Text/String
Args: x as String
Returns: Integer

Description:
Returns the ASCII decimal value of the first character of
x. The x must be typed inside double quote marks if it is
a string value. X can also be a field name, variable name,
or an expression that results in a string value.

Example:
ASCII_Code = @ASC("Sesame")
ASCII_Code = @ASC(SomeField)

The first example would return "83" because that's the
ASCII decimal value of capital "S". The second example
would return the ASCII decimal value of the first
character in SomeField. ASCII decimal values 65 through 90 are the
capital letters of the alphabet A through Z.
(Their lowercase counterparts are 97 through 122.) You can test for
this using @ASC in a program like this:

If @Asc(FirstName) > 96 And @Asc(FirstName) < 123 Then
@Msg("The first letter of the name should be
capitalized.")

Consult any ASCII table to for a listing of characters and their
corresponding ASCII decimal values.

++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

As you can see, this kind of structured output is not at all suited to a columnar layout. It requires a freeform layout—or something that mimics a freeform report.

Although I'd designed and filled the database in Q&A for DOS, I reckoned that Q&A for Windows' Freeform report feature would do the trick. But the result was

disappointing.

The big problem was with the multi-line **Description** and **Example** fields. For a given record, these two fields might contain just a line or two, or a dozen or more lines. At Q&A Win's Freeform report layout screen, I had to make each of these fields large enough to accommodate the record with the most data in each one.

When I printed the report to a file using a generic text-only printer driver, then opened it in Word, there were huge gaps (blank lines) in those records that contained only a small amount of data in the two multi-line fields. Worse, whereas most of the data was printing starting at the left page margin, some lines randomly started printing outside the left margin. Q&A for Windows had made a right mess of it.

Back to Q&A for DOS.

The solution turned out to be pretty easy. Not a Print Spec, which can produce output roughly equivalent to a Freeform report but takes a lot of doing to set up—but a simple merge document as shown below in Figure 2 .

When I designed the database, I added a field named **Sort By** and programmed Q&A to fill it with the function name without any preceding "@" character or trailing "(…)"as in @*DEL(n,x,m)*:

```
If @Left(Name, 1) = "@" and @Instr(Name, "(") > 0 Then
Sort By = @Mid(Name, 2, @Instr(Name, "(") - 2);

If @Left(Name, 1) = "@" and @Instr(Name, "(") = 0 Then
Sort By = @Mid(Name, 2, 100);

If @Left(Name, 1) <> "@" and @Instr(Name, "(") > 0 Then
Sort By = @Left(Name, @Instr(Name, "(") - 1);

If @Left(Name, 1) <> "@" and @Instr(Name, "(") = 0 Then
Sort By = Name
```

This would group, for example, @*ReadOnly* with *ReadOnly* because they'd be alphabetically equivalent. (*Sesame* includes built in functions that also have a statement counterpart. In this case, @*ReadOnly* returns a Boolean value that tells you whether a form element such as a text box is read-only, whereas you use the *ReadOnly* statement to *set* the target element to read-only.)

After finishing the merge document, I pressed Ctrl-F6 for the Define Page screen. I set the Left, Top and Bottom margins to 0 (zero), the Right margin and Page width to 144, and the Page length to 0.

At the Print Options screen, I told Q&A to Print To *Disk* and supplied a filename. I left the Retrieve Spec blank (to include all the records) and specified *1, AS* in the **Sort By** field at the Sort Spec. When I opened the resulting file in Word (it was too big for Q&A DOS Write), the multi-line fields for each record took up only the number of lines needed to print their data, and every line was neatly aligned along the left margin. Almost perfect.

The only problem was that Q&A, as expected, had inserted a page break after each record. To delete them, I clicked on Word's *Edit* menu and selected *Replace*. In the *Search/Replace* dialog, I clicked the *More* button to expand my options. With the cursor in the *Find what* field, I clicked the *Special* button and selected *Manual Page Break* from the list. I then clicked in the *Replace with* field, clicked the *Special* button again, and chose *Paragraph Mark*. I then clicked the *Replace All* button and Word replaced every page break with a blank line. Perfect.

If the report hadn't been so lengthy, I could have brought it into Q&A Write and used the F7 Search/ Replace box to do the same thing, searching for the page break symbol (Alt-F10, followed by Alt-12 on the numeric keypad), replacing each occurrence with @*Cr*, and using the *Automatic* or *Fast Automatic* replacement option.

If I had planned to print the report from Q&A Write at the outset, I could have added some text enhancements as well. For example, I could have boldfaced the function *Name* merge field by enclosing it in the appropriate PCL printer control codes (for my LaserJet printer) as shown in Figure 3. (Printer controls hold when you print to DISK.)

About the only thing I missed was the ability to set *Allow Split Records* to *No* (as you can do in a Q&A columnar report), but that turned out not to be an issue due to the way the output would be used (as a Web page).

So you *can* get the equivalent of a Q&A for Windows Freeform report in Q&A for DOS. And depending on your output needs, it might actually be *easier* in Q&A for DOS. Keep in mind, though, that you can't print a merge document report like this directly to paper. If you do, you'll get a separate page for each merge record. You need to first print it to DISK, then open the file in Q&A Write (if it isn't too large) or another word processor, remove the page breaks, and print your hardcopy from there.

Tom Marcellus is editor of *The Quick Answer*.

**Figure 3**. Enhancing the *Name* merge field with a printer control code.

**Figure 2**. A Q&A for DOS merge doc for printing a "freeform report."

# The Robocopy One-Click, Two-Minute Total Backup

**TOM MARCELLUS**

IN the February 2003 issue ("Painless, Brainless, One-Click Backups"), I talked about the easiest method I could think of to ensure that you and your people make regular and frequent backups of your vital data. It involves using an external Iomega ZIP (250Mb) or JAZ (2Gb) drive and a batch file (as a desktop shortcut) containing the commands to copy your most important files to the external drive. The idea is to make the task of backing up so brutally simple and fast that you can do it several times a day if you want and not tie up your computer for more than a few minutes. This way, there's simply no excuse for not running regular and frequent backups.

But even this wasn't quite good enough for me—for several reasons.

First of all, I've always wanted a way to back up my *entire* computer with a single click—and have it take just minutes, not hours.

With 15Gb of data currently on my PC, backing up the whole computer onto even a 2Gb JAZ drive wasn't an option. Tape backups take more than an hour. Even incremental backups tie up my PC too long. Added to this is the hassle of getting the right tape out of safekeeping, waiting for the drive to run its initialization cycle, setting the backup software, then returning the tape to storage when the backup job is done. *Too much to do*. That's one of the reasons I used the faster, easier external USB JAZ drive to routinely back up only vital files. But I've always had this nagging worry that I might not not be including some files in the backup—new files, particularly—that might turn out to be more important than I thought if my hard drive or entire PC should suddenly fail. (The advantage of a *partial* backup is that it's fast and doesn't require a huge capacity backup medium. The downside is that you have to be constantly alert to new files and folders you're creating, and remember to update your backup command set to include them. Otherwise, they won't get backed up. Backups to tape are only as good as the condition of the tape itself, *and periodically restoring from the tape is the only way to know for sure if you're getting good backups*.)

## Two deciding factors

I knew I needed a better backup strategy when my computer suddenly died a couple of months ago. (The motherboard blew.) I was able to restore my vital files to a

backup computer from my JAZ drive backup, so I could limp along while I was having a new computer built. My most recent full tape backup, however, was useless because the backup computer did not have the same (or any) tape drive.

It turned out that the hard drive in the dead PC was fine, so I had it installed as a second hard drive in the new PC. But I was without it for the two weeks it took for the unit to be built and delivered—a huge inconvenience, forcing me to put some work on hold until the new PC (with the old hard drive as HARDDISK2) arrived.

Then, a few weeks ago I ran into an acquaintance who told me that his place had been broken into and his computer stolen. That clinched it for me. I was now utterly determined to find a lossless backup strategy that would accommodate every last one of my needs. I wanted:

- One-click backup. I don't want to have to think about it. I don't want to have to fool around with options or wonder if everything is set up right for the backup.

- No tapes, CDs, diskettes or other removable media to cope with.

- Backup the entire computer in one fell swoop—including multiple partitions and drives.

- Backup folder by folder and file by file with no encoding, compression, or imaging requiring specialized software to *un*-encode or uncompress the backup files in order to use or restore them. In effect, I wanted a duplicate of my computer's drive/directory structure on the backup medium.

- Be blazingly fast—take no more than a few minutes to complete a full backup, giving me the option to back up *several* times a day without tying up my computer.

- Backup device/ media virtually theft-proof.

- Not cost-prohibitive.

Is all this possible in a *single* backup plan? Yes.

## The hardware component

Four of the requirements I mentioned are (1) no tapes, CDs, diskettes or other removable media, (2) blazingly fast, (3) backup device/media virtually theft-proof, and

available, it may well make everything run much slower.

It is easy to tell if an XLookup pushed your database over the edge. If your database is running smoothly; searches, sorts, and general interaction all run without a stutter, and then suddenly everything slows down—you probably have XLookup loading a second database. It is that second database that is pushing your application over the RAM limit for your computer.

The first thing to be done is to apply the optimizations outlined above to that second database. If that is not enough, it may be advisable to look at some small structural changes to your overall application. Let's say, for example that your main database is an invoice system. As part of your invoice system you access parts of your inventory database to get the price of an item to create the invoice. But by doing so you are also dragging a lot of information about the item into *Sesame* that does *not* get used on the current invoice— such as a description of the item, its restocking levels, its wholesale price, maybe the URL for your supplier.

Sticking with XLookups as the means to provide a relationship between these "tables," we can partition the data to optimize both databases. In Q&A the original relationship has two tables. One with invoice information; and one with item information. If we divide the second table into two parts using an XLookup in the inventory database, it becomes possible to load the price information into the invoice database, without also loading all of the other fields about the item. The inventory database would then also use an XLookup to get to the other fields about that item.

Because *Sesame* has variables, as opposed to using fields for storage in programming, a further optimization can be achieved in the example. Neither the inventory database, nor the customer database need allocate any fields for the XLookup imported information. It can be placed on a form, and/or a report, and used in programming, using only temporary variables. This reduces the duplication of fields by a factor of two, and will be a significant savings given the associated per-record overhead. The price you pay for this optimization is paid in complexity. Other than the savings in RAM, there is no good reason to support a one-to-one relation across a set of tables.

### More tricks

Okay, so you've done all that and you are still swapping. (Wow! Just how small *is* this computer?!) Still no luck getting the boss to shell out for some RAM? Don't despair, *Sesame* still has a few more tricks up its virtual sleeve. *Sesame* is fully client/server. That means you can run the server, with all of the data, on one computer; and run the user interface and all of its gizmos, whatzits, and gadgetry, on another. So maybe you have an old clunker

sitting in the closet. It has old 72 pin RAM, so you can't pilfer its parts for your very modern (but somehow way under powered) main computer. But, if it is a Pentium, with at least 32 Mbs of RAM, it will run Windows95, and may do quite well running the client side of *Sesame*. By allowing the server to run without the user interface, you can save anywhere from 3 to 10 Mbs of RAM on the server, depending on the size of your forms and the number of tabs you have open in *Sesame*. A good graphics card for business use costs less than $25 these days. Combined with a Pentium class machine (please give me at least 133 Mhz, I mean c'mon!), a network card, and a little bit of RAM, is enough to drive *Sesame's* user interface.

But if leaving the server unencumbered by the client's gizmos and widgets is still not enough, even if combined with some structural changes for XLookup, and the elimination of unnecessary fields, then there is a one more thing you can try before you shell out $86 for some more RAM (Oh, just buy it already!) You may be able to split up your database into two or more smaller databases. This can only be done if there is a natural division in the data that does not need to be reconnected via XLookup. After your form becomes two forms, you can start two servers, each running one of the two forms. That may not sound like much of a savings in that you are running two executables each with half the data the single executable had before. But there is a quirk in swapping that can be advantageous. Active processes get memory priority. So the server that is working will swap into RAM and the server that is idle will swap out to the hard drive swap file. There will be a lengthy delay when you switch between the two servers, but if your data is well partitioned that may not be much of problem. Effectively, you have cut your data size in half. It is, admittedly, a rare database that is amenable to this solution, and even among those, data synchronization may well be a worse problem than a little swapping.

### Conclusion

Well, I've presented quite a few solutions to a relatively rare problem. From simple form changes to hauling the old clunker from the closet, I know that very few readers will ever *need* to know all of this. But I hope that you have found some of the hints and tips here enlightening for applications in general, and that you will sympathize with the developer's plight the next time you decide that the cute animated doggie screen saver is worth the 72Mbs of RAM it eats while you are at lunch. If nothing else, I hope I have conveyed that RAM is cheap, your time is not.

Mark Lasersohn is a principal of Lantica Software, LLC and Technical Lead for the *Sesame* product development team.

# One-Click Backup

(4) not cost-prohibitive. The only device that satisfies all these criteria is a remotely located (concealed) external hard drive. And when I looked around, I was surprised at how inexpensive these units were. I got an external Western Digital 7200 RPM 80Gb USB 2.0/1.1 hard drive for a mere $200 at *www.westerndigital.com.*

The thing about an external USB hard drive is that you can place it up to 20 feet away from your computer, such as inside a blind cabinet shelf, a nook or an enclosure in another room, where a thief wouldn't know to look. With USB 2.0, data transfer rates are comparable to an internal hard drive, so they're very fast. Older Windows 98SE computers can be retrofitted with USB 2.0. See, for example, *www.pcmag.com/article2/0,4149,3673,00.asp.*

These drives work just like another hard drive. When I plugged mine into one of the USB ports, Windows XP immediately recognized it. No software CD required. I went into My Computer, and there it was—all brand new 80Gb's worth of it with next drive letter assigned to it.

You can partition these external hard drives if you want. Mine came with a single 80Gb FAT32 partition. Even though the two internal drives on my new computer were NTFS and FAT32, respectively, this didn't cause any problems. I simply created three folders on the new drive named *C*, *E*, and *F* to match the *C:*, *E:* and *F:* partitions on my fixed drives. (My D: drive is the CD drive.)

So the hardware part of my backup strategy boiled down to spending a couple of hundred bucks for the backup "medium" and other $15 for the longer Belkin Gold USB 2.0 cable.

## The software component

One of my requirements was that the backed up files must be native, usable files—not encoded or compressed files that require specialized software to deal with. The old (DOS) COPY and XCOPY weren't designed for the kind of backup I wanted, so I began scouring the Web for a suitable alternative. I came across one called *XXCOPY*—a beefed up version of XCOPY—but for reasons I won't go into here, that wouldn't fill the bill, either.

After "Googling" "XCOPY," though, I found a free utility called *ROBOCOPY* at *http://www.devhood.com/tools/ tool_details.aspx?tool_id=321.* Here's what its purveyors had to say about it:

> *A tool created by developers at Microsoft and included in the NT 4.0 Resource Kit. This is a nifty little app that has robust copying capabilities. I've used it to automate backups at work, copy files, gigs of files from web servers that are geographically far away without any problems. (Other utilities had problems with corruption or timing out, but this one has settings for retries and timeouts.) Lots of command line options, log files. You can even copy the directory structure first before copying the files*

> *(helps with HD fragmentation since directory structure is written sequentially). This thing works like a charm. It's an oldie but a goodie.*

After reading more about it in the robocopy.doc file that came with it, and checking out the command line options it supported (see page 14), I knew that this was just what I was after.

## Powerful, fast, and easy to use

ROBOCOPY's main strengths are that it can recreate your source drive's entire structure on the target drive (folders, subfolders, files and all) and it will *not* waste time copying files with the same name, timestamp and file size. (It copies files without altering their timestamps.) This way, if the matching source and target folders both contain a file with the same name and timestamp, ROBOCOPY just skips it. This is an efficient way to backup files (like an *incremental* backup on a tape drive) and a great time-saver. It might take ROBOCOPY half an hour to initially copy everything from your computer's fixed drive to an external drive, but every subsequent backup will take only a few minutes because the vast majority of files on your computer will *not* have changed between backups.

ROBOCOPY's *Purge* feature, which deletes any folders and files from the target drive if you've deleted them from the source drive, keeps the folders and files on your backup drive synchronized with those on the drive(s) you're backing up.

About the only thing ROBOCOY won't do is copy files that are in use on the computer at the time you run your backup. But that's to be expected.

For files that might be in use only momentarily, you can use ROBOCOPY's optional */R:n* switch to retry *n* times before giving up on a file. Retries are 30 seconds apart by default, but you can change that.

If you want, you can have ROBOCOPY generate a log file of all the files it wasn't able to copy because they were in use.

ROBOCOPY is ideally suited for running from a plain old batch file. In fact, my desktop shortcut (named *Backup*) does nothing more than run my *robocopy.bat* batch file containing the following commands.

```
ROBOCOPY C:\ G:\C\ /E /XA:SH /R:0 /PURGE /NP
ROBOCOPY E:\ G:\E\ /E /XA:SH /R:0 /PURGE /NP
ROBOCOPY F:\ G:\F\ /E /XA:SH /R:0 /PURGE /NP
PAUSE
```

Taking the first line, I'm telling ROBOCOPY to copy all the files from my computer's fixed C: drive to the *C* folder on my external G: drive. The */E* switch copies the folder (directory) structure as well, giving me a duplicate directory tree of my file system on the external drive.

ROBOCOPY supports folder and file *exclusion* with switches that enable you to *exclude* by filename extension, wildcard, file attribute, and the like. I'm trying out the / *XA:SH* switch to exclude system and hidden files.

The /R:0 switch tells ROBOCOPY *not* to retry a file that is in use. Any files that *are* in use on my computer during a backup are likely to remain that way, so there's no point in wasting time on retries. (For backing up network files, though, this switch could prove useful.)

The /PURGE switch, as I mentioned, deletes any files, folders and/or subfolders on the target drive if they've been deleted from the source drive since the last backup.

Finally, the /NP switch turns off ROBOCOPY's progress indicator, which shows the percentage of the current file being copied. Turning it off seems to make ROBOCOPY run a bit faster.

The *pause* command at the end of the batch file is so that when I walk back into the room after a break, I can see at a glance that the backup actually completed.

There are a host of other optional switches you can use, including network-related switches. (ROBOCOPY was designed as a network server administrator's tool.) They're all listed in the robocopy.doc file that comes with the free download (and recapped on the following page).

Of course you also have the option of running your ROBOCOPY backup on an automated timed basis via Windows Task Scheduler.

## Setting it all up

I mentioned placing the external hard drive up to 20 feet from the computer, in a location that's not visible or easily accessible. In my case, my computer sits about eight feet from some deep enclosed cabinets that are attached to the wall like kitchen cabinets.

When I set up my workstation, I ran the cable for my cable modem through the bottom shelves of these cabinets, through the wall on the other side, and out into the area where the cable hookup was located.

For the eight-foot span of exposed cable, I bought a Wiremold CordMate kit at Home Depot. The kit's hollow rectangular channels (about ½-inch by 1-inch) are designed to protect and hide data lines or appliance cords. They attach to a wall, ceiling or floor using peel-off pressure-sensitive adhesive backing. You can snap them open and closed—to insert or remove cables—even after they're installed. If you run one of these along the bottom of a wall, it blends in so well (especially if you paint it the wall color) that people don't even see it.

I used the same channel for the external drive's USB 2.0 cable. But where the modem cable enters the side of the bottom cabinet shelf and keeps going, the USB cable ends just inside the cabinet. That's where I secreted my external hard drive. It's a deep enough cabinet with a low enough bottom shelf so that even if you open the cabinet door, get down on your hands and knees, bend your head way down to floor level and look in with a flashlight, you still can't see the L-shaped blind spot where the small-footprint drive sits. There was room to add a small 425VA UPS in there as well—so my external drive, like my main computer, has clean uninterrupted power.

There was a power outlet at the other end of the

cabinets, so I ran a grounded extension cord in from there.

I've thought about having a perforated steel enclosure fabricated that I could place over the drive like a hat and bolt to the bottom of the cabinet. And I still might litter the area in front of the components with assorted odious paraphernalia (mouse traps, old pots and pans, concertina wire, unlaundered athletic wear) as a further deterrent.

With the converter that powers the drive, the UPS, and the drive itself all giving off heat, I was worried about the temperature in there. But it turned out that the temperature rose almost imperceptibly over the ambient room temperature even after several days.

## Running backups

As I said, my initial backup took ROBOCOPY about 30 minutes to copy all the folders and files from my computer to the external hard drive—about 15Gb of stuff. (I get close to a 500Mb/minute data transfer rate to the external hard drive.) The time this takes will of course vary depending on your computer, whether you have USB 2.0 or just USB 1.1, and the volume of data you're copying.

But don't expect it to take anywhere near as long once you've done your initial backup. On an average day I (and my frenetically *far* busier operating system) might create or change only 1000 or so files, so ROBOCOPY will only copy those ones.

Anytime I want to run a full bakup, I just click my *Backup* shortcut. The lines in the DOS-like ROBOCOPY window scroll at a breakneck pace (impressive to watch) with the slightest hitch each time the utility hits a new or newer file and copies it. A scant 90 seconds later, 100,000 files have been processed and my "closet" external drive now contains the latest copies of all the files on my computer. With this kind of convenience and speed, I can easily backup several times a day.

## Restoring stuff

ROBOCOPY simply copies folder/subfolder structures and files. It doesn't encrypt or compress anything. So I don't need any special software (including ROBOCOPY) to use or manipulate my backup files. I can copy them from my external hard drive back to my fixed hard drive individually or folder by folder. I can go into Explorer and simply drag files or entire directories between drives.

If my computer suddenly melts down (by far my biggest fear because I've experienced it), I've got everything on it backed up—and the backup is no older than a few hours.

If somebody steals my computer, I've got everything that was on it stored elsewhere.

I could head over to Fry's or CompUSA, walk out with a new box and be back up and running in fairly short order. An inconvenience and expense to be sure—but *not* an unmitigated disaster.

If somebody torches the place, well, then I'm S.O.L. Almost—because I still run a weekly tape backup at night and store the tape  inside a small fireproof box cable-

locked to the inside of a locked steel cabinet.

As for my life-and-death files, I upload a ZIP file of these weekly to a hidden folder on my Web site. (Storing copies of your most important files on a password-protected remote server has gained popularity in recent years. See, for example, *www.usdatatrust.com*.)

## Parting notes

ROBOCOPY is a Unicode application and will not run in DOS or Windows 95 because these do not provide full Unicode support. Also, ROBOCOPY uses the newer CopyFileEx() Win32 API, which is specific to Windows NT 4.0 and later versions of Windows. So it will not run under Windows NT 3.5x. You need Windows 98SE or later.

A word about viruses as they relate to backups.

ROBOCOPY *copies* files—even infected ones. Unlike a tape backup, when you *copy* files to another disk, any infected file is volatile *on* that disk. If you run or otherwise open it from the backup disk, it can damage your computer in the same way it would if you ran or opened it from your fixed drive. The conveniece of simply copying files to another disk using a utility like ROBOCOPY does, in this regard, increase the risk of distributing infected files. So it's really vital that you have a solid anti-virus regimen in place, complete with up-to-date virus definitions.

In addition to ROBOCOPY, I'm looking for a way to add Norton Ghost to the mix. Ghost creates an image file that is actually a complete copy (or clone) of a hard disk or partition, including the operating system, installed software, system files and data files.

If your hard drive utterly fails, and you've recently backed up to another drive using Ghost, you can replace your failed hard drive with a new one, then restore to it using the Ghost image file saved out to the other drive. This way, you don't have to reinstall your operating system or software packages.

One problem with Ghost, though, is that it needs to restart your computer to generate its backup file *before* Windows loads. This probably has to do with it being able to fully access every last one of your computer's files, including those that would normally be in use and off limits once Windows was running. What's more, it can take Ghost a while to make its backup—10 minutes or more for a drive with 6GB of data on it.

But why Ghost at all? Well, suppose you *only* had your folders and files backed up to an external drive using ROBOCOPY as discussed in this article. In the event your fixed hard drive failed, you'd have to reinstall your operating system and all your *windows-specific* software programs to the new drive or computer before restoring all your data files from the external backup drive. This could amount to quite a production if you had dozens of programs to reinstall.

Personally, though, I don't fret over having to reinstall Windows or the programs I use. What worries me is the risk of losing (or even just temporarily losing *access to*) any

files that *I've* created and modified over the years (including this morning). In the event of a disaster, all these would be recoverable with a recent ROBOCOPY backup to my hidden external hard drive.

Another thing about having an external drive with all your data on it is that you can hook it up to your notebook computer—or any computer for that matter—and easily copy your files over to it. With Ghost backups, you need both Ghost *and* similar hardware to restore to. And a question as to how *similar* is "similar" might make the process a bit iffy if not downright risky.

In any case, I'm working on a way to launch Ghost at night, when I'm not on the computer. I plan to have it save it's image files to the external hard drive since there's no lack of space there and, again, I'll be protected in the event my fixed hard drive or entire computer dies (or is stolen).

Tom Marcellus is editor of *The Quick Answer.*

---

# A Summary of ROBOCOPY's Command Line Options*

*Usage* :
**ROBOCOPY source destination [file [file]...] [options]**
*source* : **Source Directory (drive:\path or \\server\share\path).**
*destination* : **Destination Dir (drive:\path or \\server\share\path).**
*file* : **File(s) to copy (names/wildcards - default is "*.*").**

**Options :**
**/S : copy Subdirectories, but not empty ones.**
**/E : copy subdirectories, including Empty ones.**
**/T : Timestamp all destination files, including skipped files.**
**/R:n : number of Retries on failed copies - default is 1 million.**
**/W:n : Wait time between retries - default is 30 seconds.**
**/REG : Save /R:n and /W:n in the Registry as default settings.**
**/TBD : wait for sharenames To Be Defined (retry error 67).**
**/FAT : create destination files using 8.3 FAT file names only.**
**/X : report all eXtra files, not just those selected.**
**/V : produce Verbose output, showing skipped files.**
**/L : List only - don't copy, timestamp or delete any files.**
**/A+:[R][A][S][H] : add the given Attributes to copied files.**
**/A-:[R][A][S][H] : remove the given Attributes from copied files.**
**/XA:[R][A][S][H] : eXclude files with any of the given Attributes**
**/A : copy only files with the Archive attribute set.**
**/M : like /A, but remove Archive attribute from source files.**
**/XF file [file]... : eXclude Files matching given names/paths/**
**      wildcards.**
**/XD dirs [dirs]... : eXclude Directories matching given names/paths.**
**/XC | /XN | /XO : eXclude Changed | Newer | Older files.**
**/XX | /XL : eXclude eXtra | Lonely files and dirs.**
**/IS : Include Same files.**
**/Z : Copies files in restartable mode.**
**/NP : No Progress - don't display % copied.**
**/ETA : show Estimated Time of Arrival of copied files.**
**/MOVE : Move files and dirs (delete from source after copying).**
**/PURGE : delete dest files/dirs that no longer exist in source.**

**\*See Robocopy.doc for more information**
**  *http://www.devhood.com/tools/tool_details.aspx?tool_id=32*1**

3. Some standard columnar sorts.

Let's begin with what's required to print something only once in a report. This is a special feature of Q&A reports that allows comparison of the current record with the previous record by calculating the columns in strict column number order. In the report Column/Sort Spec, you set the **Team** field to *5,AS,H(Team).* Then you create a Derived Column that looks like this:

```
Heading: Team copy
Formula No. 1  #5
Column/Sort Spec: 500,I
```

This Column #500 stores a copy of the **Team Number**. But because it's the last column in the report, any reference to it in an earlier column reads the *previous* record's **Team Number**. To see the **Total Rec'd** value only once for each team, set the **Date Received** field to *30,I*—then add another Derived Column:

```
Heading: Received Count
Formula No. 2  @Text(#30 <> "", 1)
Column/Sort Spec: 32, I
```

Then one more Derived Column:

```
Heading: Total!Rec'd
Formula No. 3  @Text(#5 <> #500, @count(#32, #5) )
Column/Sort Spec: 40,F(JC)
```

What you are doing is telling Q&A to sort by **Team Number** (*5,AS*), and make a duplicate of the **Team Number** in the report's last column, but don't print it.

Next, you create a non-printing column that has a "1" in it for each record that has a **Date Received**—*@Text( #30 <> "", 1)*.

Finally, you tell Q&A: "Whenever the team number changes" —*#5 doesn't equal #500,* where #5 is the team number in this record and #500 (which Q&A hasn't gotten to yet) is the team number of the last record *(#5 <> #500)*—"print the count." You could also use @Total for total, or total of all of the 1's in the **Received Count** derived column for only this team, as in *@Count(#32, #5).*

Using the same approach with some minor changes, you can design each of the other columns. For example, for Received, Not Reviewed, where **Received** is *30,I* and **Reviewed** is *35, I* :

```
Heading: Rept Rec'd Not Rev'd Count
Formula No. 4  @Text(#30 <> "", @Text(#35 = "", 1)
)
Column/Sort Spec: 50,I
```

When the report has been received *(#30 <> "")*, if it has *not* been reviewed *(#35 = "")* make this column a "1":

```
Heading: Received!Not Rev'd
Formula No. 5  @Text(#5 <> #500, @Count(#50, #5))
Column/Sort Spec: 60,F(JC)
```

And there it is, your non-Crosstab Crosstab, non-Totals-Only Totals Only report that will produce results that look exactly like the Excel spreadsheet you've been manually preparing.

Bill Halpern is co-owner of Professional Computer Technology Associates (PCTA) in Newtown, Pennsylvania and is President and CEO of Lantica Software, LLC. 215-598-8440. bill@pcta-usa.com, www.pcta-usa.com.

# Second Major *Sesame* Beta Release Goes to 200+ Testers

In case you haven't heard, Lantica Software, LLC just released (April 22nd) the second major version of it's *Sesame* Beta software. This one includes a trimmed-down main *Sesame* executable; improvements to the database engine, including better memory management; further improvements to *Sesame* Designer, the facility you use to create and modify databases and applications, design forms, and add optional programming; and the first comprehensive programming manual detailing *Sesame's* powerful SBasic programming language. (*Sesame* supports Q&A programming as well.)