

Searching for Solutions in Back Issues

HAVE you ever used the *Inside Sesame Keyword/Topic* or *Subscriber Express* search index at www.insidesesame.com only to come up empty-handed?

Did you assume that these indexes included every word or term in every issue of *Inside Sesame* since its inception in January 2004?

They don't.

But there *is* an index you might have overlooked that *does*.

Types of indexes

We occasionally get a call or email from a subscriber with a question about back issues of *Inside Sesame*. It's usually about something they remember reading months or even years ago that they didn't need then but could use now. But they can't seem to find it in the back issue indexes. (See Figure 1.)

- The online *Subscriber Express* index is the usual way folks search for what they need in the back issues — but it's limited because it's based on a half dozen or so keywords assigned to each *Inside Sesame* article, tip and Help Desk item by a human and usually restricted to the *main topic* of the article, as suggested by its title.
- The *Issue* index (Figure 1) isn't really an index at all, just a summary or synopsis of each issue by month and year.
- The *Keyword/Topic* index is an alphabetical index of assigned keywords only. So if you're looking for issues containing something on "RGB," you click on the "R" and scroll down the resulting list until you get to that term.

The *Subscriber Express* index lets you search on a variety of criteria, such as issue year, title, keyword, author name and summary (synopsis) — with an optional universal search that looks for matches anywhere in the index file.

But the index file itself — which originates in a Sesame database designed for the purpose — is limited.

So when searching on a term or phrase, unless it's specifically related to the topic or primary focus of the article, chances are these indexes won't find it.

Fuzzy logic not supported

Here's an example. The programming command "@XResultSetSearch" might appear in an article or tip (in

some sample programming code, perhaps) but if it isn't directly related to the *topic* of the article, it may not be on its keyword list. So the indexes won't find it.

You might recall an article you read that had to do with *invoice line-item subforms*. It had some programming for a particular task that used @XResultSetSearch. You remember that the program did something special that you could now use in an altogether different project, but you can't find what issue it was in.

Using the *Subscriber Express* index, you type "@XResultSetSearch" into the **Universal Search** box, click the **Run Search** button and get some hits. But none of them seems to be the tip or article you were thinking of.

So you type in "subform" because that's what you remember being the topic of the article. You get a boatload of hits, but none of the titles or synopses sounds like what you remember. And you have no way of knowing if any of those issues contain the particular @XResultSetSearch technique you remember without opening each PDF.

Word for word

In cases like these, the only kind of search that's going to help is one that'll find *every* occurrence of "@XResultSetSearch" *anywhere* in *every* back issue, regardless of the topic. Kind of like a Sesame *Universal Search*.

The only way to do *that* is to search the *Inside Sesame* PDF files themselves.

And the only way to do *that* is to have those PDF files on your own computer so you can use Adobe Acrobat's built-in search facilities.

Ergo, the best way to organize your *Inside Sesame* PDFs is to save them from insidesesame.com to your computer and *store them all in the same folder*. Then you need search only that one folder.



LOOKIN' FOR SOMETHIN'?

Three Indexes to **INSIDE SESAME** now online...
[Issue](#) | [Keyword/Topic](#) | [Subscriber Express](#)
[Download the INSIDE SESAME Index Database](#)

Figure 1. Access the back issue indexes from the home page at www.insidesesame.com.

(If you store your issues in separate folders by *year*, you'll have to run individual folder-by-folder searches until you find what you're after.)

Using Acrobat's Search facilities

Assuming you have your *Inside Sesame* issues on your computer, the first step is to start Adobe Acrobat.

I'm embarrassed to say that I still use Adobe Acrobat 6.0 Standard Edition (the full, paid program that includes *Distiller*), so I start my searches by clicking *Edit / Search* or the *Search* icon on the toolbar. (See Figure 2.)

The latest of Adobe Reader at this writing is 9.1 and can be downloaded free from <http://get.adobe.com/reader/>.

I seem to recall that earlier versions of the free Adobe/ Acrobat Reader didn't have the same search capabilities as the full, paid version. But I recently downloaded the new, free Adobe Reader 9.1 and it's search capabilities are every bit as good as my paid version of the full Acrobat program.

When the *Search PDF* panel opens (see Figure 3), type in the word or phrase to search, then check the *All PDF Documents in* option.

Specify the path and folder to search by selecting the *Browse for Location...* option on the dropdown, then navigate your file system and select the folder where you store your *Inside Sesame* issues.

Click the **Search** button and Adobe Reader will search all the issues in the specified folder for your word or phrase and return a list of them by filename. They should be in date order from the earliest issue at the top to the most recent at the bottom. (See Figure 4.)

The list of issues where your search term was found can be expanded and collapsed like the Sesame menu tree.

A particular issue might contain several occurrences of your search term (see Figure 5 on the following page). In this case you can click on the "+" icon next to that issue to see all the occurrences. Each one is listed with a bit of the surrounding text to give you a rough idea of the context.

When you click on one of the occurrences, the PDF opens to the relevant page with the search term highlighted (Figure 5).

Earlier versions of Adobe Reader open just one PDF file at a time, which I like. The latest version opens a separate instance of Reader for each PDF, which can get a bit messy if you're checking a number of issues.

Conclusion

So download your issues and keep them all in the same folder. If you have a lot of back issues to download, the fastest way is to right-click on the issue link (**JAN 09**, **FEB 09**, and so forth) then click *Save Target As...*

Then should you need to do an all-issues global search for a term or phrase that you believe might lead you to what you're after, you'll always have a more powerful tool just a few clicks away.

Searching the back issues this way is also a great way



Figure 2. Starting a search in Adobe Acrobat. (Optionally use *Edit / Search*.)

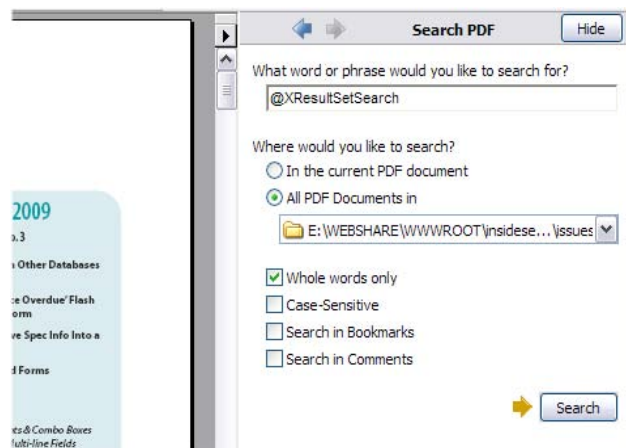


Figure 3. Typing in the search term or phrase and specifying the folder to search.

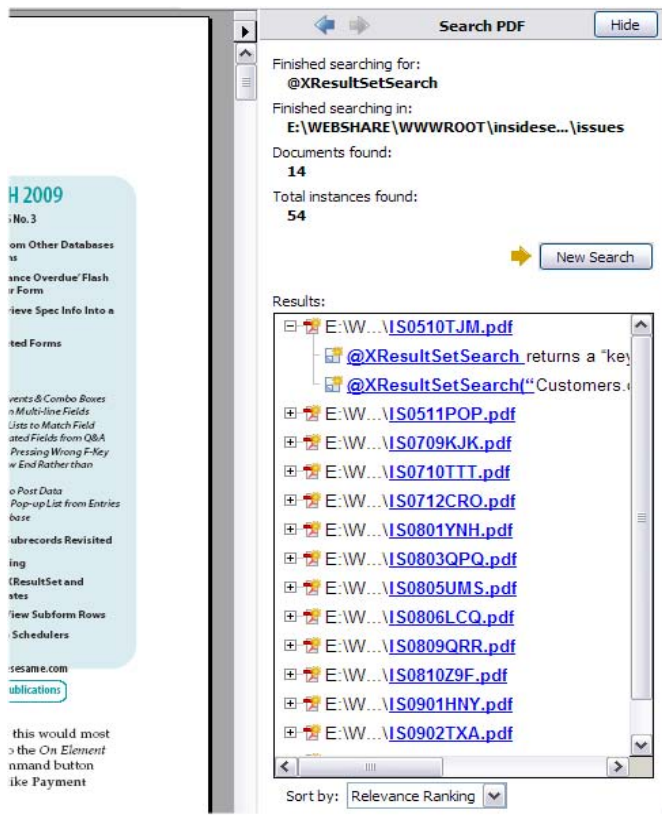


Figure 4. Getting the result set of relevant PDFs and clicking the "+" icon next to an issue to see the context in which the search term is used.

to get more usage examples on a particular programming command. The *Sesame Programming Guide* has good examples, but you might want a few more (to make sure you're covering all the bases) before writing your routine. A search on *@FormAsDialog*, for example, will turn up four issues with more than a dozen occurrences. If you're about to embark on a project where you might want to employ *@FormAsDialog*, reviewing what's been written about it (including sample code) in *Inside Sesame* can only broaden your understanding of how this feature works as well as what its strengths and weaknesses might be.

Of course if you don't *have* all the back issues, then your searches will be limited to just those issues you *do* have. If you plan to be doing a lot of Sesame development, you might want to invest in the complete back issue set (where just *one* article in *one* issue might pay for the entire set). Contact us 800-780-5474 / office@insidesesame.com. If you're a current subscriber, we'll make you an offer you'll find hard to refuse.

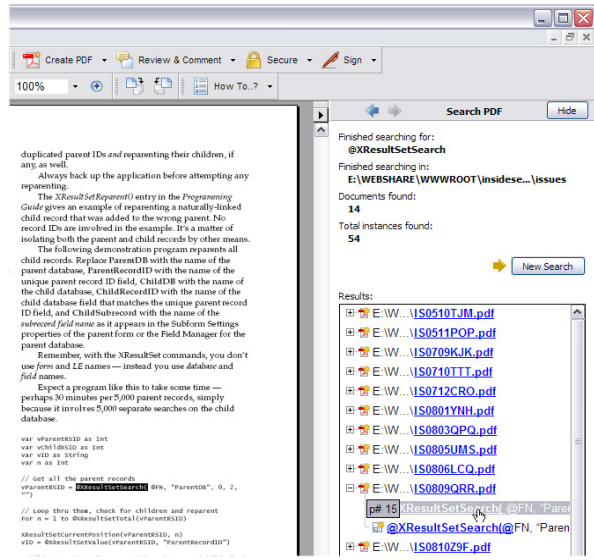


Figure 5. Clicking an occurrence of the search term. Acrobat opens the selected PDF and highlights the term.



Alec Mulvey

Simplify Mass Updates for Users

Here's an easy way to make mass updates simple for users. It applies to a common scenario where records are assigned a new value that varies each time the mass update is run.

Say you've just run a mailmerge and now want to update the records by adding the name of the mailing to a keyword field named **Mailings**. You do this by running a mass update that inserts the name of the mailing at the start of the field and follows it with a semicolon. The mass update might be something like this:

```
Mailings = "Easter promotion" + ";" + Mailings
```

This is pretty straightforward — to a developer. But the poor old user might not be familiar with Sesame mass updates, and needs to know to double-click on the programmed field in the Mass Update Spec, then change the text inside the quotes — without deleting the quotes — oh, and ignore the quotes with “,” inside them... etc....etc. Not so easy if it's something the user only does a few times a year.

You can make these mass updates a snap by prompting the user for the necessary value(s). Here's how.

First, you need to use Global Code. So double click on *Global Code* in the Mass Update Spec. (See Figure 1.) For this example, enter the following program.

```
STAT stMail ingCode as String
```

```
stMail ingCode = @Questi onUser("ENTER CODE FOR THIS MAI LI NG", "")
```

Then, in any other element, add this:

```
Mail ings = stMail ingCode + ";" + Mail ings
```

When the mass update is run, an input box appears for the user to fill out. (See Figure 2).

The user simply types the text for the mailing. In the example code shown, there is no default value for the input box. If you want to show a default value (it can be a useful guide to the user), make it the second argument for the *QuestionUser* function:

```
@Questi onUser("ENTER CODE FOR THI S MAI LI NG", "Easter Promoti on")
```

Concludes on page 11

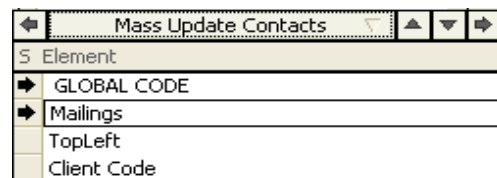


Figure 1. Mass update spec ready to run.

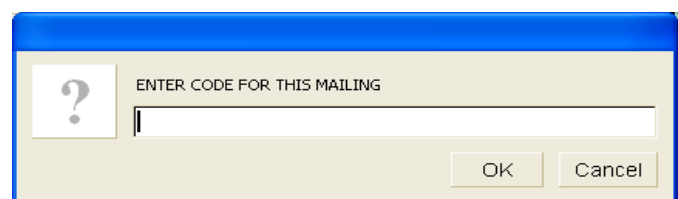


Figure 2. QuestionUser input box.