acrobat search

chapter seven

Acrobat Search employs Verity, Inc.'s text-searching technology, as do many other popular products. It is very important to consider the real-world presence and staying power in the fluid timescape of software development. By pursuing the very effective strategy of getting onto the most desktops, Verity is following in the footsteps of Microsoft, Netscape and even Adobe, which decided to offer the Acrobat Reader free over the Internet. The advantage of building a base of millions of users proves the efficacy of giving software away free. Verity technology is embedded in Adobe Acrobat, Lotus Notes, the Netscape software family and other extremely pervasive products. The real benefit to the digital author and librarian is that these products achieve a life of their own because millions of users depend upon them.

"With SEARCH'97, content providers and corporations will have a powerful and ubiquitous search platform," says Phillippe Courtot, chairman and CEO of Verity. "Content providers will be able to make their information personalized and searchable and deliver it to the enterprise. Corporations will be able to link their multiple sources of information and create their 'corporate memory' and make it easily accessible by their employees as well as customers."[1]

Why Use Search?

The vital key to Courtot's statement is Verity's unswerving focus on becoming "ubiquitous," and commitments from leading industry players ensure that Verity will remain a safe standard for building your organization's search engine.

The Verity capability built into Acrobat 3.0 and the Verity SearchPDF Web search engine are designed specifically for information retrieval on PDF collections. Additional modules add more features, such as Verity's Search '97 Agent Server Toolkit, to search collections of many other types of files, including HTML and common office applications such as word processing, spreadsheets and email. Also, topics, or pre-defined searches, can be stored and executed on demand. There are even third-party products that offer entire sets of topics that can be selected from menus, putting extremely sophisticated search capability within the reach of the average or occasional user. The Search '97 Agent Server Toolkit can be configured to "watch" and retrieve specific information from many sources, including Web servers, databases, newswires and netnews.

All of the above growth and flexibility options offer plenty of development paths for the future. However, there are other full text retrieval engines that can also search PDF, HTML and many other forms of information, so the PDF author and publisher is not limited to Verity. Other text-search engines that currently support PDF include Open Text, Excalibur, Personal Library Systems and Fulcrum. (See Chapter 13 for details.)

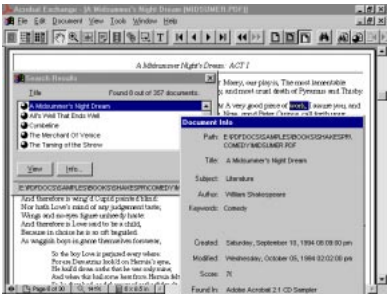
For more information on search engines, see the following Web sites

http://www.opentext.com
http://www.excalib.com
http://www.pls.com
http://www.fultext.com
http://www.verity.com

Basic Text Searching Via Exchange

The Portable Document Format is an "encapsulated" document format that contains significant amounts of information about the file, as well as the contents of the file. The fields described in Chapter 4 can combine with text search to take advantage of these built-in features of PDF files.

Adobe Acrobat Exchange is a database for collections of PDF files that provides the capability to extensively search the collection using a combination of techniques. On the most elementary level, documents can be retrieved via specific fields. Alternatively, the contents of the document themselves can be queried via text-search techniques. These two techniques can be combined to perform very sophisticated queries. For example, to retrieve all files by certain authors published during a specific time period, you would combine Author and Date Created fields. Within this defined subset of documents, the user can search for specific words, terms or phrases within the contents of the text.



On the left, a compound search of the Shakespeare Collection on Acrobat CD.
On the right, the multi-level organization of the Catalog database: the Source Doc, the Source doc Info, and Rank.

Adobe Acrobat Catalog is required to build the index for these collections of PDF documents. Acrobat Exchange is then used to query and retrieve the documents on a network or Intranet, and *Acrobat Search for CD* can be used to query and retrieve collections that are published on CD-ROM. A combination of these packages can be used to publish collections on both CD-ROM, to take advantage of the large storage available on CD, and on a net, to take advantage of frequent updates.

Web links in PDF files allow information on CD-ROM to be linked to complementary files on the Web or on an Intranet. For example, a tech manual published and distributed on CD could be used on an individual workstation. By clicking on a Web link in the CD text, the user would be linked via his browser and communications connection to the Web or Intranet.

For example, this book contains many references to products that are current as of this writing but that will undoubtedly be updated as time passes. By making Web links of all of the references to products and companies, the information would stay current by always connecting to the dynamically updated information on the Web.

tip

In general terms, any text search follows four basic stages:

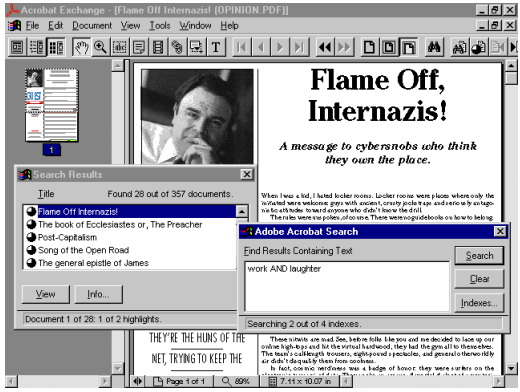
First, the query terms are entered and joined as necessary.

Second, the query terms can be expanded or restricted.

Third, the search is run and the results list is presented to the user.

Fourth, the user can then view the documents containing hits, or refine the search based on the results list.

As we will see below, and in Chapter 13 on Advanced Text Search, these simple steps can be greatly automated and enhanced by the text-searching software. It is important for users of digital libraries to become proficient in searching because of the tremendous access such skill provides to the information within digital collections.



The user has an option to display the document with other info in the search screen.

The Exchange search screen illustrates the steps between simple and advanced text searching. The three Action buttons on the right side, including Search, Clear and Indexes, are the primary controls for issuing a query. Search executes the query based on all of the information entered on this page. Clear empties out all of the query fields for a new query, which is very important because variables left over from previous searches will often cause the incorrect data to be retrieved. The Indexes button allows the user to choose between several databases upon which the search will be performed. Many separate indexes can be searched simultaneously, and the responses will be delivered in one ranked list of hits or results.

The user may review Index Information to determine the likely relevance of a particular document. The following information is available for each Acrobat Exchange Index:

Title	Name in the Index Selection dialog box
Description	Publisher or author description of the index
Path	Full path location and name of .pdx index file
Last Built	Date of last update when new and changed documents were added
Created	Index creation date
Documents	Number of PDF documents in the collection

In a digital library where there may be a great number of subcollections, each with its own index, this feature is very helpful for the users. There are many reasons for the creation of a number of separate indexes, including operational efficiency, disparate and unrelated collections, and multiple sources of collected documents.

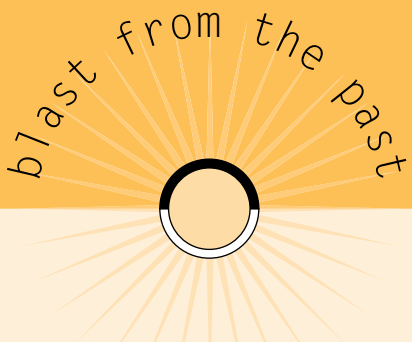
Upon selecting one or more indexes, the user can conduct a search by creating a query. In the first field the user can Find Results Containing Text by entering the actual words or terms of interest.

Even when searching for a simple word, this is a different function in a text database from that in a word processing file. The Find feature in a word processor will only find the one, single word or term that is entered in the particular document that is open. In a text database, such as that created by Acrobat Catalog and used in Acrobat Exchange, the query will search an index of all the documents in the collection. Usually, the results will be reported back as a list of documents ranked by occurrences of the query within the documents.

Building More Extensive Search Queries

A query begins with a single word or phrase, and a simple query will retrieve a set of documents containing that word or phrase. This method works perfectly well as long as the single word or phrase is relatively unique in the database. If the word or phrase is relatively common in the database, or if it is likely to appear in irrelevant documents, the query will retrieve too many documents to be useful.

For example, if you were searching for information on the famous aircraft designer Kelly Johnson, you might enter the term “Kelly” or “Johnson.” In any moderately sized collection, this query would retrieve any document with the individual words occurring



Boolean algebra is named for the English mathematician George Boole, who lived between 1815 and 1864. It is a mathematical system originally devised for the analysis of symbolic logic, in which all variables have the value of 0 or 1. For this reason, it is widely used in computers.

within them. However, in a big collection, you'd probably get a results list that is too long to comfortably peruse—a major disadvantage of searching for a single word.

To improve upon this example, you could search for the phrase "Kelly Johnson" in quotes instead of the individual words. Now the results list will contain every occurrence of the entire phrase, but will miss "Clarence Johnson," which is his given name. It will also miss "K. Johnson" and "Johnson, Kelly" because they do not match the phrase in quotes.

Boolean logic is the method used in text searching to combine multiple query terms. The three primary Boolean Operators are And, Or, Not.

To improve the results of a text search, the user is encouraged to use more than a single word or phrase to describe information of interest. Boolean logic allows the user to build a query that contains many terms used in combination. It should be noted that many text databases reduce the size of the index and speed the search and retrieval by removing the "stopwords." Such stopwords include common articles and prepositions, such as "of," "the," "by," "for" and so on. Because the removal of such stopwords may limit the user's ability to search for specific phrases, this feature is optional. In creating some text databases, including Acrobat Catalog, the author has the option to decide to include some or all of the words in the index.

Below are a few search examples:

Query	Finds documents that contain
1. price **And** discount	Both “price” and “discount”
2. price **Or** discount	Either “price” or “discount”
3. price **And Not** discount	“Price” but not “discount”
4. (total profit) **And** “revenue” or “income”	The phrase “total profit” and either (revenue or income)
5. “profit and loss”	The phrase “profit and loss”

Examples 1-3 above demonstrate the self-evident functions of the Boolean operators. Example 4 introduces another convention derived from mathematics, namely the use of parentheses as a symbol of grouping. In a text search, as in math, the entire contents within the parentheses are considered one result. In this the case, the parentheses define a term (total profit) made up of two words, and a combined OR term (revenue OR income).

In the fifth example above, quotes are used to define a phrase that overrides the Boolean And operator and simply treats the word “and” as a query term. This means that both parentheses and quotes can be used to form phrases, but the double quotes are required when the phrase includes search operators, like Or in this example.

Another way that Boolean text-search logic corresponds to mathematical formulae is that queries can be comprised of a great number of related elements. And these numerous elements can be grouped or “nested” to virtually any depth within layers of parentheses.

By default, the Boolean And operator is evaluated before the Or operator. The Not operator is evaluated before either of the other two operators. This is logical because And more precisely defines the search than Or; and Not still more precisely defines the search because it specifically excludes certain matches.

Parentheses can be used to change the default order of evaluation of the Boolean operators and can dramatically change the results of a query. For example:

The query “(darwin or origin) and species” would find all documents that contain either “darwin” and “species,” or those that contain “origin” and “species.”

The query “darwin or origin and species” would return all documents that contain “darwin” or “origin and species.” You could get documents that contain “darwin” but do not contain “origin” or “species.”

The potential length of such complex queries is suggested by the size of the box under the prompt Find Results Containing Text in the Acrobat Exchange search windows. In the hands of a skilled user, precise queries can be built that will search vast databases and return only highly relevant information. This is increasingly true in direct relation to the user's knowledge of the contents of the database.

Expanding Your Search Terms For Better Results

Straightforward Boolean searching as described above depends upon the user knowing the specific appearance of the term as it occurs in the database or databases. Remember, Boolean searching was designed for simple logic systems where the only values are 0 and 1.

This means that the spelling in the query must exactly match the spelling in the occurrence. The query term "search" will not find "searching" or "research." To overcome this severe limitation, many methods have been developed to make text searching more flexible. As we will see, Term Expansion includes a wide range of techniques that cover not only alternate spellings of query terms, but even various meanings of the terms.

tip

Acrobat Exchange ignores punctuation in query terms and searches because many relevant hits would be obscured by punctuation. For example, if punctuation were not ignored, the query "Johnson" would not find the word at the end of a sentence because it would appear as "Johnson."

Wild Cards

Wild cards are symbols that can represent any character or any string of characters within a query statement. This same feature is available even in simple operating-system-level functions. For example, a simple DIRectory command in DOS can contain wild cards. The command "dir *.doc" will return a list of every file in a directory that has the ".doc" extension because the asterisk stands for any string of characters.

There are two types of wild cards available in Acrobat Exchange, and they are single-character wild cards and string wild cards, symbolized by "?" and "*" respectively.

Below are a few wild card examples from the Exchange Help file:

Wild card	Matches
geo*	words such as geode, geodesic, Geoffrey, geography, geometry, George and geothermal
*nym	words such as antonym, homonym and synonym
?ight	words such as fight, light, might, right and sight
555-????	all seven-digit numbers with the 555 prefix
pr?m*	words like premature, premeditate, prim, primate, promise and promontory

In the first two examples above, the two sides of the problem are covered. In some advanced text-search descriptions, these two functions are referred to as left-hand and right-hand truncations. In the English language, these functions are very helpful in handling prefixes and suffixes. For example, the query "*fill*" will retrieve all terms using the root of "fill" including such words as "refill" and "filling."

A more precise version of this right and left truncation is accomplished by the single-character wild card, "?," which substitutes only one character in the search string. This can be very helpful in applications where the user is searching part numbers or file names that are predictable in length if not in content.

Word Stemming

Word stemming offers a way of expanding query terms in the most relevant directions. Rather than randomly replacing each character as if it were a simple 0 or 1 in a logic argument, word stemming uses actual language resources to determine the core word in a query term. Word stemming is the default option from Exchange Search dialog box.

Using word stemming, query terms are stripped down to their linguistic antecedents, such as their Latin or Greek root words. For example, when the word stemming option is applied to "build," such words as "building" and "rebuild" will be found.

Sounds Like

Sounds Like is another expansion option in the Exchange search, and it is similar to the Soundex method of most spell-checking programs. As the user types in a word, character by character, the Sounds Like expander displays a list of words that are somewhere near the query term in the alphabetical listing in the database. This

method of expansion should be very carefully used because of its gross level of association. Much more than the above methods, Sounds Like can add an inordinate number of irrelevant terms to the query.

On the other hand, the Sounds Like option can give the user an instant preview of the terms as they appear in the database that is being searched. When used this way, as a preview of good search terms, Sounds Like can be extremely helpful. The benefit to the user is twofold. First, the search terms can be restricted to only those terms that appear in the database. Second, and more important, the Soundex may suggest forms or alternative spelling of terms that had not occurred to the user.

Getting Help From Computer Intelligence

In the previous section we considered a number of techniques for expanding the individual terms entered by the user. Further help is available from computer intelligence, which comes to the aid of the user by suggesting additional query terms.

Where a term expansion varies the spelling and extensions of a *particular string* entered by the user, a query expansion varies the meanings and emphasis of the entire idea of interest.

Thesaurus

The Thesaurus expands the search to find words that bear some semantic resemblance to the search terms you enter in the Find box. For example, searching for "begin" finds "start," as well as "attack," "produce" and many other terms.

Of all the query expansion techniques, Thesaurus has the greatest potential to expand the search to irrelevant topics. To help the user manage this tendency, Acrobat Exchange offers the Word Assistant to allow a preview of the expanded terms before they are included in the search.

Individual Observation And Creativity

By individual observation, the user will notice certain words and features of the most interesting documents returned on the hit list. On the Web, for instance, you'll find that certain sites seem likely to contain the most interesting content through your prior experience. The labor required to gain the advantage of such perfected searches means that the user must read or review the documents returned by the hit list.

Advanced Text Search includes the ability of the software to "read" documents and "understand" them based on the concurrent appearance of certain words or phrases. For details on the most advanced technique described, Search by Browsing, see Chapter 13.

An individual user can quickly inspect the articles as they appear in the Relevancy Rank listing. A first pass through the search reveals the first results list, so the user can read only the articles of greatest interest and then choose terms from the most interesting of the articles to build a new search. This feature is discussed later under "Refine Search."

Variables to Restrict Queries

Proximity

The Proximity option instructs the search engine to find occurrences of two or more query terms that are appear close together in the document. In Acrobat Exchange, the terms must appear within a few pages of one another for the Proximity option to take effect. Without the proximity option, two query terms related by the And operator may appear anywhere within a document. The reasoning behind this option is the common sense logic that says if the words appear close together, the passage in which they appear is more likely to be of interest.

The effect of the Proximity option is demonstrated in relevancy ranking of the documents. Not only are documents with the highest number of hits found, but also most closely occurring hits, or clusters of hits, are ranked higher. And the greater the proximity of the hits, the higher the ranking.

Match Case

Match Case allows the user to specify such items as proper names, initials and other case-sensitive terms. For example, using the Match Case option to search for the chemical symbol for helium, "He," would not find the male pronoun "he."

The Match Case option allows the user to restrict the search to only those items most likely to be of interest, and to minimize the number of irrelevant hits. Match Case does not work with Word Stemming, Thesaurus or Sounds Like options because they are all term expansions.

Word Assistant

The Word Assistant offers the users controls to refine the expansion options. The user is presented with a dialog box where a query word may be entered and the effect of Sounds Like, Word Stemming and Thesaurus may be previewed. All the words that will be found when a certain option is used are listed. From this list the user can selectively cut and paste suggested terms to the search box.

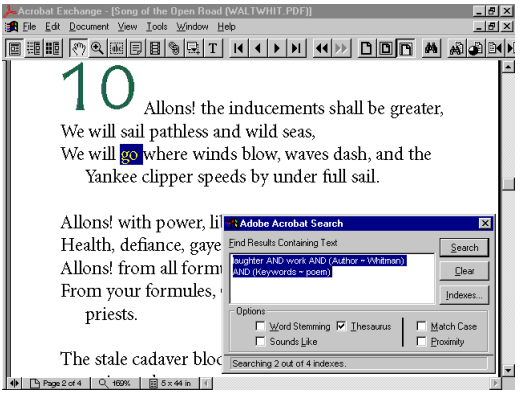
Without this Word Assistant, the Sounds Like and Thesaurus would often expand the query term to the point where many irrelevant hits would be returned. This option allows the user to enjoy the full benefits of expanded search without the downside of imprecise queries.

The user can select only the best terms in thesaurus list.

Narrowing Your Search Using Document Information Fields

All of the text-search capabilities described above can be directed to act upon a specific subgroup of documents by using field values to restrict the search. Through the combination of Text Search and Field Values Search, the user can take advantage of simultaneous unstructured and structured searching. Most important, the user is given finer control over the entire process of searching for information within a digital library.

For example, if we were searching a library for post-war American writers, we could specify “Vonnegut or Kerouac or Pynchon” in the Author field. By concentrating on this subset of authors, we could get an alternative view of World War II compared to a wider search that would include history and newspaper stories and so on.



Boolean expressions can be used within the document info fields to select certain groups of documents. This means that all of the inclusive and exclusive terms described above can be applied to the following.

The **Document Info fields** displayed above are separated into different types of info:

The standard **Document Info fields** are **Title**, **Subject**, **Author** and **Keywords**.

The **Date Info fields** are **Created** and **Modified**.

The **File Info fields** are **Path** (to hit document) and **Found In** (Index Title).

The **Score field** is the numerical ranking assigned by Relevancy Ranking.

tip

Document Info field searches will retrieve documents with or without terms in the Text Query field. In this sense, the Document Info serves as an advanced card catalog, or a free-standing Citation Index.

Authors and publishers of PDF files are not required to enter the information in the Standard Document Info fields, and these fields may be empty. The information in these fields is automatically captured from other applications that use such fields, such as word processing files.

Since these fields can affect the relevancy rank listings, it is usually desirable to enter the information into these fields. Acrobat Exchange allows editing of these fields for manual updates. In Chapter 9, Advanced Navigation, techniques are discussed for automatically filling in these fields for batches of PDF files.

Single-Field Search

If the user does not enter any words or terms into the Text Search field, and uses only the Document Info field, any and all documents that contain the value entered will be retrieved. For example, using just the Keyword field, a user could find every document that the publisher has determined to be relevant to the topic of Search.

ActiveX: ActiveX is the branding name for Microsoft OLE (Object Linking and Embedding) Controls. These controls allow programmers to use embedded functions within Microsoft environments to perform specific functions. This means that Web applications can take advantage of all of the programs in the Microsoft Office and Professional suites that are on 80 percent of all desktops. This is part of Bill Gates' strategy to offer the Microsoft Explorer as the universal interface to digital documents, whether they reside on your hard drive, LAN or the Internet. (It should be noted that Microsoft also licenses Java, which is similar in intent of small, fast program applets that work on all platforms.)

The best place to learn about ActiveX is http://www.microsoft.com, which is an extremely busy site. At this time, the "ActiveX, Activate the Internet" page can be found at

http://140.116.72.228/xxjyh/ActiveX/Overview.html

The effectiveness is determined by the nature of the material within the collection. For example, a very large collection of documents related to engineering and manufacturing documents could be managed with just the Document Info fields.

In this case, the documents themselves tend to be specifically identified by Work Order, Part Number, Purchase Request and similar information. By agreeing to a convention to entering specific information into the Title, Subject, Author and Keyword fields, documents could be retrieved by any one piece of information in any one field.

Multiple-Field Search

Continuing the previous example, engineering documents are often changed through many versions, and the Date Info fields could be used to track the latest and all historic versions of a document in which the only field that changes is the Modified Date.

In this case, if the user knows one of the Key fields, he can retrieve all of the versions of the file or a specific version of the file. By adding a Creation or Modification date, the user performs a simple two-field search to retrieve a specific document.

All of the Document Info fields can be used to perform combined searches. In collections where much of the field info is repetitive, such as a collection where a single author has a large number of documents, very selective retrievals may be made done on subsets of the documents.

Wild Cards In Fields

The wild card characters function in the same way within fields as they do in full text searches. Both the single-character wild card "?" and the single-to-multiple character wild card "*" are available. Therefore, the query string "Sm?th*" will return occurrences of Smith, Smyth and Smythe.

For example, if Part Number or Purchase Order or other such data is stored in a field, the user could perform wild card searches to retrieve information even when he has only partial data to start with.

Bot: This slang form of "robot" refers to almost any program that performs tasks similar to those done by a human. In other words, a bot is a program that runs on its own, performing tasks that involve decisions and discrimination, and does it in an unattended mode. The big search engines use such bots to update their Web indexes automatically. The Eliza program is often considered the first bot because it acted like a human.

A page called "World Wide Web Robots, Wanderers and Spiders" by Martijn Koster offers a WWW Robot FAQ and a list of known robots

http://info.webcrawler.com/mak/projects/robots/robots.html

Beyond Document Info Fields: Expert Searching

In any field of study where there are frequent updates or regular series of articles, these simple fields can be limiting. Of course, with the wealth of data available here, anything is possible! But in less stringent applications, where the end user is not expected to be an expert in query writing, the document info needs to be expanded.

Chapter 14 describes full-featured Relational Database Management Systems to organize document collections. In this case, documents can be cross-referenced ad infinitum, and a powerful body of meta-information about a digital library is available to expert users.

Expert users of information include many academic and commercial professionals who follow extremely dynamic fields of development. Due to very high volume of information in these fields, it is a necessity, not a luxury, to provide improved finding aids and access controls.

A citation database serves as an example. The same author may write on the same topic in many different journals, with different emphases appropriate to each journal.

You might want to find all examples where John Dvorak (author) has discussed "interactivity" (query term) in Boardwatch magazine (keyword), but you may not be interested in the same author's comments on the same subject in other publications, or in articles published before a certain date. In a customized database, all of these elements can be tracked and valued separately, and the digital library user can choose a very specific set of documents to view.

Java: The Java™ Programming Language Platform from Sun Microsystems can be thought of as the HTML of programming. Just as HTML documents are designed to run on many hardware and software platforms, the Java language is designed to allow developers to write an application with the same freedom from constraints.

An excellent starting place to learn about Java is "The Java Language Environment" White Paper by James Gosling and Henry McGilton at

`http://java.sun.com/doc/language_environment`

Even further, a citation applet could be written in Java or ActiveX. This bookworm-bot would allow the raw results of a database or text search to be returned to the user for further investigation. Along with the results, a customized applet would pop up to allow instant sorting, reviewing and reporting on all of the fielded information. The user could then retrieve this carefully selected subset.

In this way, the remote user of a database can intelligently examine the contents of the library. Precise retrieval of information is the goal of this theoretical applet.

tip

Whenever you attempt to access a specific URL, such as those listed above, and find that the page no longer exists, it often helps to simply back up to the base of the URL and try again. For example, while the White Paper at Sun in the last example may change and get updated over time, it is likely that "http://java.sun.com/" will always offer relevant information on the same topics.

Range Searches: Creation And Modification Date Fields

The system-generated fields of Creation and Modification dates provide the user with an elementary version control capability. In addition to the ability to include these Date fields in the arguments described earlier, the user may directly access these fields through the Search screen.

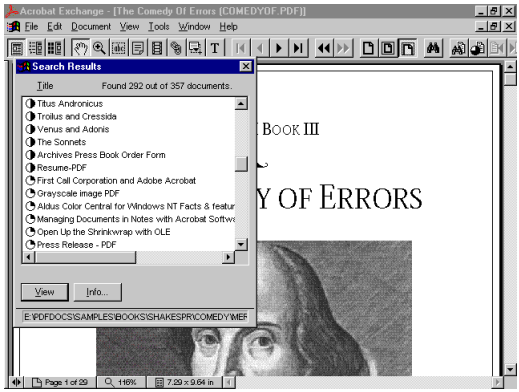
Because the user can search for both before and after, as well as exactly equal to or exactly not equal to, the user can easily specify a range of files by date.

Sometime last year I read a neat set of criteria for determining true portability, as envisioned in the dynabook idea. The author used something like “The Three B’s: Bed, Bathroom and Beach” to identify the obvious advantages of books over current portable computers. The problem is, I can’t remember the author, title or any other information about the book. And a text search for “The three B’s: Bed, Bathroom and Beach” is nonproductive because it’s not an exact phrase and the terms themselves are uselessly common.

However, these common terms applied to a short date range may be more productive. Since I remember when it was published, I can focus my search by date and have a better chance or finding the document.

Relevancy Ranking In Results List

A document’s relevancy ranking presents an orderly list that starts with the documents most likely to be of interest. Acrobat Exchange uses five icons to indicate relevance, which range from a full circle to an empty circle, with 3/4, 1/2 and 1/4 as the three middle gradations. A full circle indicates the most relevancy, an empty circle indicates the least.



Unlike a library, the user views catalog and book at once.

The actual method used varies somewhat according to the type of search, but in general there are four rules for ranking the documents:

Occurrences Of All Query Terms In One Document

When two or more query terms are associated with the OR operator, documents that have more of the query terms are ranked higher than those with fewer of the query terms. This method gives greater relevance to documents that contain every term, compared with documents that contain many occurrences of a single term.

Total Hits

In a single-term search, the documents with the greatest number of hits will have the highest relevancy ranking. All other things being equal, namely that all of the query terms appear in all of the top documents, total hits will generate higher rankings in multiple-term searches.

Proximity of Hits

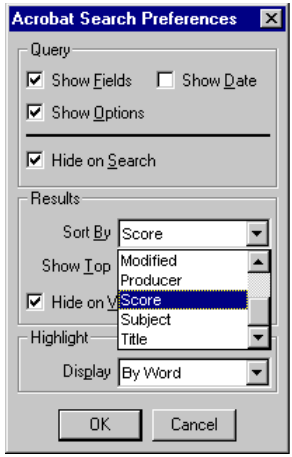
When the Proximity option is selected, documents containing occurrences of the query terms appearing closer together will be ranked higher than those with more dispersed hits. This ranking is only in effect for occurrences that are within a few pages of one another.

Hit Density

Finally, the number of hits within a document is compared to the number of words within a document to determine hit density. Of two documents with same number of terms and hits and the same proximity ranking, the shorter of the two documents will be ranked higher due to the higher hit density.

Views Of Search Results

By default, search results are listed as document titles in order of Relevancy Rank Score. The software is trying to help the user by following a simple set of relevance rank rules. However, the user may want to look at the documents according to a different set of criteria other than this so-called relevancy rank.



The user can rapidly re-rank the Results List.

The Search Preferences dialog box allows the user to specify exactly the way in which documents are sorted in the results list. Once again, through a very simple interface, the user has access to techniques of Relational Database Management Systems. In a traditional RDBMS environment, each of the various "views" of the data that the user is able to generate by simply choosing the Sort variable would have been Custom Reports, in the sense that the user could specify the appearance of the output.

The user can sort the results list by the various info fields:

Author	**Created**	**Creator**
Keywords	**Modified**	**Producer**
Score	**Subject**	**Title**

In effect, the user can choose the primary sort field for the results list, and thereby generate dynamic views of uniquely clustered hits. In this case, the word "dynamic" means that the user has great control over the view of the data and can easily change the report structure of the hit list without redoing the search.

The power of this technique cannot be overestimated, and an extremely sophisticated digital library can be built on this architecture that will serve the needs of a wide spectrum of users.

The least sophisticated, first-time user can achieve successful searches on the same database that allows the most sophisticated, hardcore users to generate extremely focused retrievals.

Adjusting The Highlighting Of Hits

In Acrobat Exchange, the user can choose three levels of highlighting within documents:

Highlight	Movement in Documents
By Word	Each click moves back and forward in highlighted hit terms
No Highlighting	No highlights, each click moves back and forward through pages
By Page	All hit terms highlighted, page movement same as above

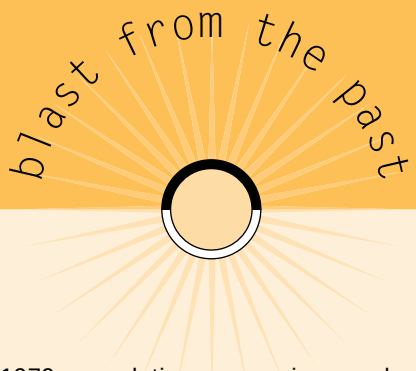
The highlighting of terms attracts the user's most intense interest, so it is a very important consideration in the design of any electronic document system. For example, the last choice above, Highlighting By Page, offers an instantaneous, visual relevancy ranking function to the user. Any user can pop up a screen full of text, concentrate on the highlights, and very quickly decide the relevance of the document.

The effectiveness of user-level enhancements such as highlighting cannot be overestimated. However, too much highlighting and emphasis can be distracting, so discretion is advised in customizing these elements for each application.

Building On Previous Retrievals

The Refine Search feature allows the user to perform a new search upon a previously retrieved set of documents. This capability can be very productive because the user can educate himself with a quick browsing of documents retrieved by a wide search. After reading a few returned documents, a sharp user can get a good feel for the nature of the documents in the collection. Based on this understanding, the user can build a much more refined query by simply including and excluding certain terms, authors and so on.

Any new set of search terms may be entered, including document info, date info and full text. Since the new search is restricted to the results of an earlier search, this process can be continued and the user can "drill" down through a collection to the most interesting documents.



Way back in 1979, a revolutionary one-piece word processor made the cover of Time Magazine. With the screen, keyboard and disk drives all molded into one sleek console, the Lanier No Problem looked quite futuristic. But what was even more revolutionary than its appearance was a slick user interface that was built upon function keys and mnemonics.

At that time, the dominant competition were earlier generation models from IBM, Wang and Xerox [and others, like Jacquard, NBI (Nothing But Initials - old inside joke), and so on, found only in history books, now]. All of the dominant early models offered a menu-based user interface.

Part of the appeal of Natural Language interfaces is related to this ease-of-use idea of dumbing down the system for the inexperienced users. In this case, rather than using menus or function keys to replace commands, simple language is used to replace complex query language. At the annual TREC convention, where the most intrepid information retrieval vendors line up in open competition, the queries are intricately crafted equations that are far from Natural Language.

Back in Sydney in 1979, where I worked for Lanier Australia P/L, when confronted with the so-called ease of use of menu-oriented systems, we would offer the following argument: “Menus are like training wheels, which are great when you are learning to ride your bike. But after you know how to ride, they just get in your way.” We would then point out that these menus were mostly impressive in sales demonstration, when the potential user was seeing the system for the first time. Most people soon came to see menus as a slow, repetitive way to use a computer.

Limitations of Acrobat Searching

Acrobat Searching is a powerful way to access information but there are limits to its present functionality.

Searching Without a Natural Language Query Interface

Natural Language Query Interface means the user can just enter an ordinary question, with no special structure to the query.

Following the common wisdom (wish!) that simpler is better when it comes to computers, it seems like a great idea to be able to perform complex information-retrieval operations by asking common, everyday questions. Expectations tend to be deeply influenced by science fiction, such as the dictation machine in the movie "Being There," upon which Peter Sellers, in the role of Chauncy, watched his master write his last will and testament by speaking to the device.

These devices do work, they just don't work like that! And the common perception of Natural Language interfaces to text databases is similarly simplistic. By surrendering precise control of the query to a generalized program, the user loses a certain amount of precision. As always, the author, publisher or digital librarian must concentrate on the needs of present and future users and provide the best tools for that clientele. If future users will probably come back and use a collection over and over, it is to be expected that they will spend a few minutes learning the elementary query language and commands because they will be self-motivated to become more efficient.

That said, the allure of Natural Language search capability is still very powerful. After a user has exhausted all of his ideas for query terms and fields to search, it is great to have the option of throwing his ideas to the computer for help. The greatest adventure on the Web is finding ideas and information that you never knew existed. Because you were ignorant of its existence, you could not know how to search for it.

The glory of Natural Language text searching is that your query terms are modified and expanded, syntactically and statistically, to find new information that you would probably never stumble upon by traditional methods, limited as we all are by our own ignorance. This process of discovery is something no stone-and-mortar library could ever make so easy.

tip

Today, experienced users of search engines with Natural Language interfaces usually begin to use more concise and focused query language. This behavior is exactly analogous to using Function Key shortcuts in Mac or Windows.

No Fuzzy Search

Fuzzy Search may be thought of as a form of automated wild card searching. Fuzzy Search is designed to find imperfect occurrences of the query term, and this is accomplished by a very smart software algorithm that substitutes wild cards for each of the characters of a query term.

A Fuzzy Search for the term "search" might be thought of as multiple wild card searches such as "?earch," "s?arch," "se?rch," "sea?ch," "sear?h," and "searc?." Such a multiple wild card search, which will find every occurrence where any one of the characters in the term is missing, is equivalent to a Tight Fuzzy Search. Correspondingly, a Loose Fuzzy Search would make allowances for more missing characters in the string.

The value of Fuzzy Search is that the user doesn't have to enter a different search term for each exact occurrence the target term.

tip

Fuzzy Search capability offers a solution for using unedited OCR-produced text. Rather than invest in costly quality control and editing, the publisher or digital librarian can rely upon the search engine to retrieve unrecognized text. Combined with the PDF normal format files produced by Acrobat Capture, Fuzzy Search engines can retrieve accurate representations of the original pages, with the unrecognized text represented as image inserts.

In Fuzzy Search, there is often a setting for Degree of Fuzziness, which has the effect of including more or fewer wild cards within the search terms. If the previous example, where a single wild card substitution is made for each character in the term, a "fuzzier" search might include two or more wild cards within the string.

tip

As shown in the example of wild cards and Fuzzy Search, advanced text-search techniques are very often simply automated versions of basic text-search techniques. The same discipline is applied on the character, word and document level, in terms of variable definition of what constitutes an acceptable and recognizable version of the shape, character, word or idea.

For demos of Fuzzy Search on the Web, try

http://www.zylab.com

http://www.excalib.com

http://www.pls.com

Customization Is Somewhat Restricted

The document info fields can be customized through the Acrobat Software Development Kit, but a certain amount of technical effort is required. For example, the average user may be familiar with creating new fields in a desktop database such as Microsoft Access and would find such Windows-based functions easy to modify. In the case of Acrobat, the user must declare the custom fields in win.ini, which is an area that is not difficult but rarely dabbled in by the everyday user. However, while it's not a pick-and-click process, it is not exceedingly difficult for any programmer or technically oriented user.

Report Generation Is Limited

The advantages of full database functionality within digital libraries for selective retrieval can be invaluable. Advanced listings and presentation of particular sets of files, including analysis and report generation, offer an unprecedented opportunity to create new ways of using very large, complex bodies of information.

In Verity's Topic™ product line, very specific, complex queries, called topics, can be stored and run against dynamically changing data. Because the data is always structured in a predictable format, these topics can generate very specific results or reports on the database. The capability in Acrobat is somewhat limited in this regard, and Verity provides an upgrade path to add such capabilities.

There are many third-party approaches to this function, as well as other databases that can manage PDF collections. This requirement is highly specific to each application, and many creative techniques can be applied to these areas.

Boolean Query Is Complex

To many part-time users of online information, it will seem to be a burden to learn the basic language of Boolean text searching. It is with the utmost charity and best wishes that any publisher or Webmaster can understand this complaint. It is the user's traditional role to always hope for more, easier, faster access to information. If the content is valuable and deep, users will happily learn the simple syntax because it offers quicker, more direct access to the collection.

Summary

All of these potential enhancements come at a cost, of course.

- Document info fields must be considered a bare minimum requirement to offer the most possible value, at the least possible effort.
- To include the word stemming or thesaurus capabilities, the database publisher adds a significant overhead to offer these search enhancements.
- Thumbnails can be generated automatically, but if the pages are indistinguishable in these small views, they serve no purpose.
- Bookmarks make online manuals seem familiar.
- Hyperlinks are proven productivity boosts.
- Keywords can be used for database fields, Cust#, Inv#, Serial#, etc.
- Article reading allows use of current document formats online.

All of the above items are overhead and must be very carefully weighed between their value and the effort required to create that value.

footnotes

1 Press Release, September 12, 1996, "Verity's Search'97 product family brings single user interface for searching across personal data, the Internet and the Enterprise," http://www.verity.com/PR/960912um.html.