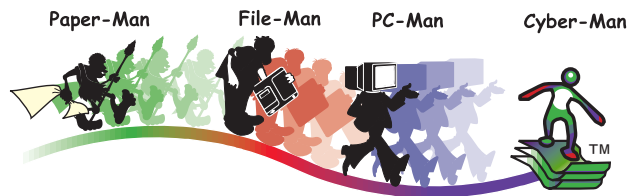


# Intelligent Searching: **Flavors** of Web-Based Text Retrieval

Article reprinted from PC AI Magazine, July/August 2000, [www.pcai.com/pcai](http://www.pcai.com/pcai)

**T**ext retrieval on the Web has one main objective — to find information in a knowledge database. Within that parameter, there exists a surprising number of possible applications, all quite different. This article reviews three text retrieval paradigms on the Web, each with its own individual requirements:

- an Intranet application that searches and manages documents;
- an Internet search engine focusing on a specific category of knowledge on the Web; and
- a Web data mining operation requiring no programming.

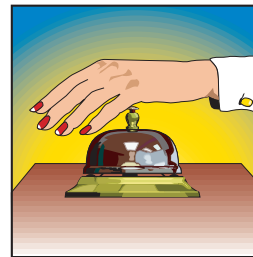


The differences among these Web-related paradigms are key to selecting the correct text retrieval solution. To highlight these differences, this article describes applications using the same text search and retrieval component. The dtSearch® Text Retrieval Engine acts as a common denominator to illustrate what is unique about the paradigms that the applications exemplify.

## Managing a Document Database with an Intranet Application

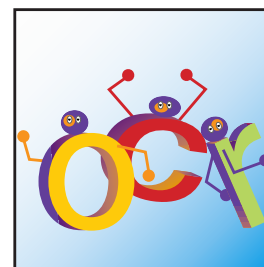
This application, effectively document management evolved to the Web, has the same essential features as traditional desktop or LAN document management, but operates through a Web-browser interface. Advantages of the Web browser interface include a smaller learning curve, access from remote locations through the Internet, and generally less expense than the establishment of a corporate WAN.

Document management is responsible for managing large and diverse collections of documents in various formats such as word processing, spreadsheet, database, etc.



Central to document management is document check-in when a document enters the document repository, and check-out when a document leaves the document repository.

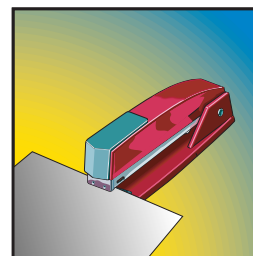
If the document is already electronic, checking-in usually involves no more than dragging and dropping into the relevant document management folder. For documents that exist purely in paper form, the document management product frequently includes a scanning and OCR function, so that the document goes



from paper to the relevant electronic file folder.

Another important feature of document management is version control, which ensures two people are not unwittingly editing two different copies of the same document at the same time.

Security access control ensures that documents checked into certain folders are only accessible to specific users. Document stapling allows a user to tie documents together. When one document is retrieved, the software notifies the user of other documents stapled to that document.



Security access control ensures that documents checked into certain folders are only accessible to specific users. Document stapling allows a user to tie documents together. When one document is retrieved, the software notifies the user of other documents stapled to that document.

With thousands of documents inside a repository, document management applications must provide methods for instantly locating managed documents. The system can ensure that each document that is checked into the repository contains certain

# Flavors of Web-Based Text Retrieval

	Web Document Management	Web-Based Content Search Engine	Do-it-yourself Web Data Mining
Medium	Intranet	Internet	PC
Files Supported	Broad Range (word processor, spreadsheet, database, email, ZIP, HTML, PDF, XML, etc.)	Web-based only (HTML, PDF, XML)	Web-based only (HTML, PDF, XML)
Requires file conversion (e.g. to HTML)	Yes	No	No
Database definition	Document check-in, check-out	Spider controlled by editors	Spider controlled by you
Example using dtSearch Text Retrieval Engine	WebDocumentz™	CodeHound.com	SmartSpider
Other features	Version control, stapling, security	IDE add-in	AI site selection

specific searchable field attributes. More sophisticated products also offer full-text searching of documents. (See side bar for a description of some possible search features in a document management product.).

In its WebDocumentz™ product, Tampa Bay Systems (TBS) set out to take the best aspects of traditional Windows-based document management systems and apply them to a Web-based document repository, using a simple Web browser interface. Browser supplied functions would include: instant document scans; document check-in, check out and version tracking for edits; management of users, groups, databases, sessions, folders and security; document organization by user-defined category, folders or both; and stapling.

To hold the diverse forms of data that would reside inside this system, TBS chose an SQL database for use with

its COM architecture. The SQL database stores descriptive data for each document and includes categories, folders, security, revisioning, and pointers to the documents themselves. This configuration works with popular Hierarchical Storage Management systems to provide any combination of online, nearline, or offline storage.

For a search engine, TBS needed the ability to automatically recognize and parse a wide variety of file formats, such as word processor, spreadsheet, database, RTF, PowerPoint, email message stores, ZIP, PDF and XML. TBS also had to overcome a problem that traditional non-Web-based document management did not have to face. They had to support the entry of diverse document types into the Web-based SQL repository. At the same time, the Web browser through which the end-user would be

accessing the database required HTML. So a mechanism for converting various document types into HTML.

TBS chose the dtSearch Text Retrieval Engine to supply these requirements. In addition to integrating with the COM-based architecture of WebDocumentz and having a wide variety of search features, the Engine also provided on-the-fly Web conversion to HTML for documents in popular non-HTML formats.

### Sample Implementation

WebDocumentz ships with an API enabling Web developer to quickly customize the user interface for a specific application. API functions exist for document retrieval; instant document conversion to HTML; document storage; security; check-in, check-out and version control; folders; categories; document forwarding; favorites; stapling; and database administration.

One customized implementation of WebDocumentz is a project that TBS is undertaking for Related Capital Company ([www.relatedcapital.com](http://www.relatedcapital.com)), a full-service real estate investment firm in need of a document repository and collaboration system for its asset acquisition process.

Related Capital wanted all parties (developers, attorneys, architects, and engineers) participating in an acquisition to be able to upload and comment on related documents. The system must track all required deal elements and automatically email the parties responsible if any have neglected to submit paperwork or comments within an appropriate timeframe.

Other desired features include integration with a PC fax system and transaction logging for a complete audit trail. The entire repository must also be full-text searchable, allowing parties to make postings in various file types, with dynamic conversion to HTML.

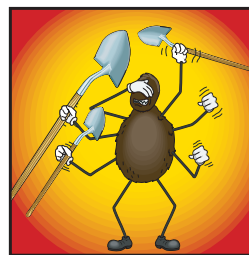
### An Internet Search Engine

Two key text search issues in Web document management are support of a broad range of file types and conversion to HTML. With an Internet search engine, all relevant documents are presumably already in HTML (or PDF, etc.) as they are already posted on the Web. Instead, the key issues are ensuring that the end-user has access to precision text search options, and defining the target database.

The example here is CodeHound.com, a new search engine focused on Visual Basic-related postings. Entering the search:

*asp and (ado w/10 sql) and not cdo*

in HotBot retrieved as the top match— with particular irony — a list entitled: “Unsolved Questions. Do YOU know the answer?” Inputting the same search on WebCrawler provided a programming tools catalog. In contrast, the same search on CodeHound found dozens of relevant Web pages.



Codehound.com used the boolean search features of the dtSearch Text Retrieval Engine to parse the boolean syntax of the above search request. On CodeHound.com, the two top-ranked pages retrieved were a newsletter on

[www.vbtechniques.com](http://www.vbtechniques.com) and an article “Using ASP and ActiveX Data Objects (ADO) for Database Access” on [VBWM.com](http://VBWM.com). The search also retrieved several ZIP files containing relevant information, and over a dozen Microsoft Knowledge Base articles.

In addition to precision text search options, the other important issue for CodeHound is how it obtains the relevant database. For this, CodeHound uses the CodeHound Spider. As CodeHound’s FAQ puts it:

Think of the CodeHound Spider as an untamed spider on a leash. Most search engines, like AltaVista, set their spiders free to roam the Internet. Others, like Yahoo!, rely on human editors to create their Web index. CodeHound is different, combining the two processes.

The CodeHound Spider automatically navigates and indexes Web sites giving users full-text searches (like AltaVista), but it only navigates sites deemed appropriate by Visual Basic editors (like Yahoo). In other words, CodeHound.com is a success because it doesn’t simply put a spider out there on the Internet to find anything Visual Basic-related. It also has human editors who weed out the irrelevant sites beforehand.

Another feature of CodeHound.com is that it gives users two methods of access. To try CodeHound.com search features, user can visit CodeHound.com through a

### Sample Document Management Search Features

There are certain search features that are important in any knowledge management or document management system. These include the ability to perform searches limited to specific fields in a database and full-text searches over documents. For such searches, the document management product provides for the entry of words or phrases connected by boolean (and/or/not) logic, proximity logic (one word or phrase within X words of another word or phrase), or both.

Two additional features to supplement boolean logic approximate the searching equivalent of retrieving a stapled document. Concept searching, also known as synonym or thesaurus searching, allows a document on education to lead to a document on learning, even if the two documents are not stapled together. Natural language searching with built-in relevancy ranking by search term density/rarity lets

the user enter a block of text such as a paragraph or a page from one document and find the closest match or matches to the text.

Most search facilities usually either permit or automatically add stemming capabilities — a search for stem will find *stems*, *stemmed* and *stemming*. Other less precise (i.e. broader range) text searching techniques include phonic searching to retrieve words that sound alike (such as *Smith* and *Smythe*) and wildcard searching, in which wildcards take the place of one or more characters (such as *ultra\**.)

Documents entered into a repository via scanning usually require some additional mechanism for sifting through typographical errors. Fuzzy searching can adjust to match the relative level of such errors. The higher the fuzziness, the greater the scope of “hits” retrieved, such as *alpeaqet* in a search for *alphabet*.

browser. Or users can download CodeHound’s free Visual Basic add-in, providing for searching of CodeHound.com’s knowledge base from within the Visual Basic IDE.

### Custom Internet Search Engine

This application also it involves collecting browser-ready data. However, unlike with CodeHound.com, where the staff does the winnowing to ensure only good sites are covered, this application involves making the decisions yourself.

For known Web sites of interest, a basic spider can point at these sites and collect everything to a specified depth level. One such spider is included free of charge in dtSearch Desktop 6.0, currently in beta testing. Further, once the Web sites are retrieved, it offers several convenient features for sifting through them.

First, unlike most popular Internet-based search engines, dtSearch graphically marks all hits, and provides convenient file navigation options. Second, it displays all retrieved Web documents in a browser with all embedded links and images intact. In other words, the version retrieved looks and acts just like the original Web page, except with the hits marked, and the ability to jump right to them.

As with Codehound.com, however, a key factor remains: how to select the sites to include in your desktop repository. For complex selection processes, it sometimes

pays not only to use a “smart” search engine, but also a “smart” spider.

(ESI) specializes in software products primarily for corporate clients. One of ESI’s recent projects is the development of a product called Smart-Spider. It searches the Web and provides the user with a list of relevant Web sites using artificial intelligence-type criteria to cull this list of sites. The criteria include factors like content, links, and heuristic analysis of key words on those sites.



For example, one application of Smart-Spider is Travel Finder, which combs the Web and finds travel information. If a customer is

searching for travel information on New York City, Smart-Spider can find sites that discuss not only New York City directly, but also Soho, the Statute of Liberty, and the “Big Apple,” even if such sites never directly mention New York. Conversely, while the Smart-Spider can find the Big Apple, it is intelligent enough to exclude such off-topic sites as those discussing fruit apples, like red delicious and Macintosh. After collecting Web sites, Smart-Spider has the dtSearch Engine built-in to perform full-text searching.