

# Indexed Vs. Unindexed Searching: From Security Classifications to Forensics

By Elizabeth Thede

Both indexed and unindexed searching have their place in the enterprise. Indexed text retrieval is typically more efficient for uses such as general information retrieval and security classification systems. But unindexed searching too has its place – in outgoing

email filtering, searching of live data sources like RSS news feeds, and sometimes in forensics. This article will attempt to explain which search technique to use when, and why.

## Overview: Indexed Text Retrieval

Indexing the inevitable millions of documents that any sizeable organization generates on shared file servers is the fastest way to facilitate data retrieval. An index will typically store each unique word in a document collection and its location within each document. Indexing also

### Sample Objects for Document Classification

In the dtSearch Engine, an "xfilter" can combine a full-text query with a filter for specific document attributes, such as file name, date, or size, or the presence in the document of a word or field. The field component can consist of a standard document attribute, or an attribute that dtSearch adds "on the fly" while indexing.

Search	Results
<i>(user request) and xfilter(name "abc*.html")</i>	This query would match any document that contains <i>(user request)</i> with a file name matching <i>abc*.html</i>
<i>(user request) and xfilter(word "projectxyz")</i>	This query would match any document that contains <i>(user request)</i> and that also contains the word <i>projectxyz</i>
<i>(user request) and (xfilter(word "Type::projectx") and xfilter(word "classification::high"))</i>	This final query adds two field restrictions to the <i>(user request)</i> : one for a named field called <i>type</i> with an entry of <i>projectx</i> , and the second for a named field called <i>classification</i> with an entry of <i>high</i> .

A dtSearch SearchFilter uses an in-memory object, consisting of a table of bit vectors, to achieve similar results to that of an xfilter.

works with non-document data, e.g. for forensics search purposes (see below).

After indexing, full-text search speed, even across millions of documents, is typically less than a second. While indexing a very large collection of documents for the first time may be time consuming, subsequent updates of the index are usually much faster. dtSearch, for example, simply checks the file modification dates of all indexed files, and only reindexes those files that have been added, deleted or changed since the last index update. (While the text retrieval terminology here relies on the dtSearch product line, the concepts in this article are generally applicable.)

In addition to enabling precision boolean searching,

an index can also store such information as word positions, enabling word or phrase proximity searching. An index can also hold information about word frequency and distribution, enabling computation of natural language relevancy rankings across a document collection. If the company name appears in two million documents, it would get a low relevancy ranking. If the latest marketing terminology appears in only four documents, it would get a much higher relevancy rank. In that way, PR could, for example, enter a whole paragraph of proposed text for a press release as a natural language search, and zoom right in on the most relevant documents.

But full-text searching, whether boolean, natural

language, or otherwise, is only part of the text retrieval answer. Suppose HR wants to limit its search to documents with an HR *executive* designation. This type of fielded data classification can result from fields or meta data inside a document, or from an overlaying document management-type application. With the latter, fielded data classification can rely on associated database entries, such as SQL or XML, or the addition of fields "on the fly" during the indexing process.

### Adding in Security Classifications

Now suppose the goal is to enable searching organization-wide, but to keep the wrong documents out of the wrong

(Text Continued on pg. 21)

**4 out of 5 of Fortune Magazine's most profitable companies purchased dtSearch developer or multi-user licenses in the past two years.**

# dtSearch® Instantly Search Gigabytes of Text Across a PC, Network, Intranet or Internet Site

**dtSearch**  
Instantly Search Gigabytes of Text  
**DESKTOP with Spider**  
\$199  
"Industrial-strength... superb" — PC Magazine

**dtSearch**  
Instantly Search Gigabytes of Text  
**WEB with Spider**  
from \$999  
"Industrial-strength... superb" — PC Magazine

**dtSearch**  
Instantly Search Gigabytes of Text  
**Text Retrieval ENGINE for Win & .NET for Linux**  
"Industrial-strength... superb" — PC Magazine

**dtSearch**  
Instantly Search Gigabytes of Text  
**PUBLISH for CD/DVDs**  
from \$2,500  
"Industrial-strength... superb" — PC Magazine

**dtSearch**  
Instantly Search Gigabytes of Text  
**NETWORK with Spider**  
from \$800  
"Industrial-strength... superb" — PC Magazine

**Publish Large Document Collections to the Web or to CD/DVD**

- ◆ over two dozen indexed, unindexed, fielded & full-text search options
- ◆ highlights hits in HTML, XML & PDF while displaying embedded links, formatting & images
- ◆ converts other file types (word processor, database, spreadsheet, email, ZIP, Unicode, etc.) to HTML for display with highlighted hits

**dtSearch Reviews...**

- ◆ "The most powerful document search tool on the market" — *Wired Magazine*
- ◆ "Intuitive and austere ... a superb search tool" — *PC World*
- ◆ "Blindingly fast" — *Computer Forensics: Incident Response Essentials*
- ◆ "A powerful arsenal of search tools" — *The New York Times*
- ◆ "Covers all data sources ... powerful Web-based engines" — *eWEEK*
- ◆ "Searches at blazing speeds" — *Computer Reseller News Test Center*

**1-800-IT-FINDS**  
sales@dtsearch.com

See [www.dtsearch.com](http://www.dtsearch.com) for:  
◆ hundreds of developer case studies & reviews  
◆ fully-functional evaluations

**The Smart Choice for Text Retrieval® since 1991**