

# If Big Data Is the Immovable Object, Enterprise Search Is the Unstoppable Force

Article contributed  
by [dtSearch®](#)

How do you approach Big Data? You could try to organize the heck out of it if you have all of the time in the world and your data isn't constantly changing. Or you could kick back and let enterprise search provide immediate access. If Big Data is the immovable object, enterprise search is the unstoppable force.

Whereas a scan-the-Internet search engine like Google crawls the web, enterprise search lets you do an in-depth exploration of your own Big Data. To instantly search terabytes, enterprise search first has to index the data. The index is simply an internal guide that pre-tabulates unique words and numbers in the data and the specific location of each, including across multiple data repositories and locations.

**Indexing, a technical overview.** Let's start with what the indexer does *\*not\** do. It does *\*not\** move, copy, delete or in any way alter original files. And it does *\*not\** pull up files in their associated applications – like you would review a Microsoft Word document in Word or a PDF in Adobe Acrobat Reader. Such an approach would take way too long.

So, what *\*does\** the indexer do? The indexer goes straight to the binary format of all files. If you looked at a binary format, you'd see a mess of binary codes, making it hard to read individual words much less complete sentences. However, a search engine can tackle binary formats because it has built-in document filters.

The document filters need to apply the correct parsing specification to each binary format before indexing. Different file types, and sometimes even different versions of the same file type, will have their own custom parsing specifications, some hundreds of pages long. Without the right parsing specification, parsing the text of a binary format will quickly hit a dead end.

**The indexer, unleashed.** With all this emphasis on precision parsing, you might expect indexing to take a lot of effort. While the document filters have their work cut out for them in the data recognition department, all you have to do is point to the folders, email repositories, etc. to cover, and let the indexer do the rest. On its own, the indexer can figure out the parsing specification to apply to each binary format. (A search engine

**Let's start with what  
the indexer does  
*\*not\** do. It does  
*\*not\** move, copy,  
delete or in any way  
alter original files.**

needs to review the binary format for this determination, not the file extension. Saving a PDF with a .DOCX extension or an Access database with a .ONE extension is all too easy.)

On the plus side, the indexer can review the data on a much deeper level than a human looking at files in their associated applications. For example:

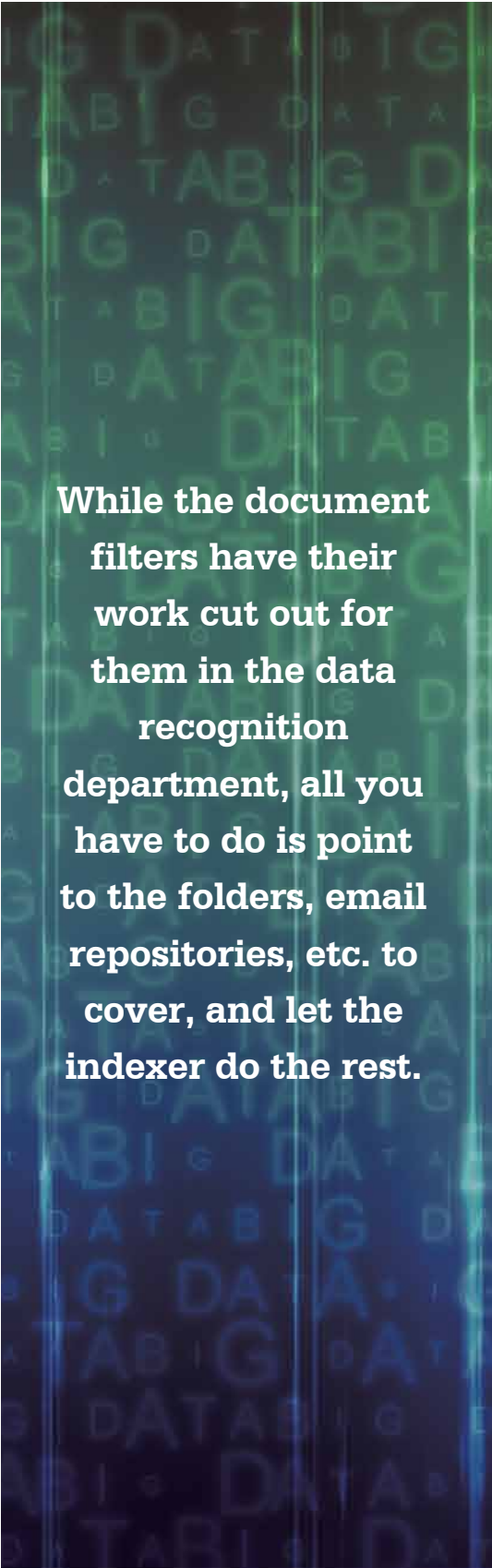
- ◆ “Invisible” text like black writing against a black background or white writing against a white background inside an associated application view is just straight-up text when it comes to indexing a binary format.
- ◆ Metadata that might take a huge amount of clicking around to find from within an associated application is readily available in binary format.
- ◆ The search engine can drill down seamlessly through multi-layered file structures, like an email with a ZIP or RAR attachment with a PowerPoint inside and an Excel spreadsheet buried inside the PowerPoint.
- ◆ Unicode ensures automatic support across hundreds of international languages, including multiple languages in the same file.

Unstoppable force. After indexing, let the searching begin. Here are just a few reasons why indexed search is an unstoppable force:

- ◆ Any number of concurrent indexed search threads can proceed at once. For online search, the index structure permits each search thread to run in a completely stateless manner, so there are no limits on scalability.
- ◆ The index structure makes available over two dozen full-text and metadata search options. These range from free-form natural language to precision word and phrase Boolean (and/or/not) and proximity search requests. Options like fuzzy searching sift through typographical errors that may appear in files like emails or OCR'ed text.
- ◆ Beyond words, the search engine can also find number and numeric patterns, including numeric ranges and date and date ranges across different date formats. The search engine can further flag items like credit card numbers that may have accidentally snuck into the Big Data repository.

Finally, when Big Data inevitably evolves, automatic index updates can handle reindexing the new items, removing the deleted items, etc., while concurrent searching continues without stopping. Move over immovable object!

Article contributed  
by [dtSearch®](#)



**While the document filters have their work cut out for them in the data recognition department, all you have to do is point to the folders, email repositories, etc. to cover, and let the indexer do the rest.**