

# Thinking Of “Spring Cleaning” Your Data So You Can Quickly Find What You Need? Try This Instead

Losing time looking for that crucial document or email? You could spend countless hours spring cleaning and reorganizing. Or you could run a text search application.

This article explains the different ways a text search application (dtSearch, as an example) can operate. The first mode of operation is unindexed search. With this mode, the text search application goes through each file looking for whatever search terms you enter.

In some way, unindexed search is not dissimilar to a human searching file by file. But whereas a human would typically pull up each file in its associated application, looking at each word processing document in Microsoft Word, each email in Outlook, etc., the text search application takes a different approach: reviewing each file in its binary format. This is the state of a file while sitting on your computer, prior to its retrieval in its relevant application.

While a text search application can go through files more quickly through binary format access, the process takes a lot of backend parsing work. If you look at the binary format of many documents, you'd be hard pressed to read the text through an ocean of surrounding binary codes. The text search application has to take the sea of binary data and meticulously sift through all of that. The first step, however, is figuring out the correct file type parsing specification to apply.

You might think that the text search application could rely on the filename extension to figure out the file type. But this doesn't always work. For example, someone could save a Microsoft Word document with a PDF file extension. For accuracy, the text search application has to look inside the binary format itself to determine the file type.

Using information from inside the binary format, the text search application can recognize document types like PDF as well as Microsoft Office formats like Word, PowerPoint, Access, OneNote, Excel and PowerPoint. The text search application can also identify and work with email formats like Outlook and Exchange, as well as web-ready content like XML and HTML. And the text search application can sift through compressed archives like ZIP, RAR and TAR. (A developer version can also parse SharePoint, NoSQL, SQL and other database BLOB and referenced files, along with the database contents itself.)

Article contributed  
by dtSearch®

**You might  
think that the  
text search  
application  
could rely on  
the filename  
extension to  
figure out the  
file type. But  
this doesn't  
always work.**

Of particular note, the text search application can also search through multilevel nested attachments. For example, if you have an email with a ZIP attachment containing a Microsoft Word file and embedded in that is a Microsoft Access file, the text search application can work its way through all of that. After parsing a binary file, the text search application can access not only document main text but also all metadata. While some metadata can be relatively hidden in an associated application view, requiring much clicking around before it becomes visible, such metadata is readily apparent in binary format.

To sum up so far, a text search application's unindexed search is typically faster than human search, including because of the former's binary format data approach. And a text search application's unindexed search is usually more thorough because it can seamlessly cover portions of files that may be harder for a human to access like buried metadata and multilevel embedded document contents. But there is an even faster way to search.

The speediest search method requires the text search application to index data first. The text search application can then use the index to search comprehensively across even disperse data stores, rather than going item-by-item looking for matches. So how do you get this index? All you have to do is point to the document folders, email repositories and other data repositories you want to cover in the index, and the text search application will do the rest.

Just as with unindexed search, the text search application will figure out for itself the relevant file formats so there no need to tell the indexer what mix of data you have. And as with unindexed search, the text search application will automatically sift through obscure metadata, multilevel nested file formats and the like. After indexing, the text search application can perform searches spanning all indexed data repositories at once with unified relevancy ranking across the entire collection.

The index or indexes can sit on your own computer enabling individual search. Alternatively, the index or indexes can reside on a shared network, on a local web server, or on a remote cloud host like Microsoft Azure or AWS, enabling instant multiuser concurrent search. And not only can a text search application instantly search terabytes after indexing, but it can do so with over 25 different search options for precision searching.

As an alternative to spring cleaning, you are welcome to go to dtSearch.com anytime to download a fully-functional 30-day evaluation version of dtSearch to instantly search through terabytes on your computer, across a shared network, over an "on premises" web server, or on a cloud-based repository.

Article contributed  
by **dtSearch®**

**The text search application will automatically sift through obscure metadata, multilevel nested file formats and the like.**