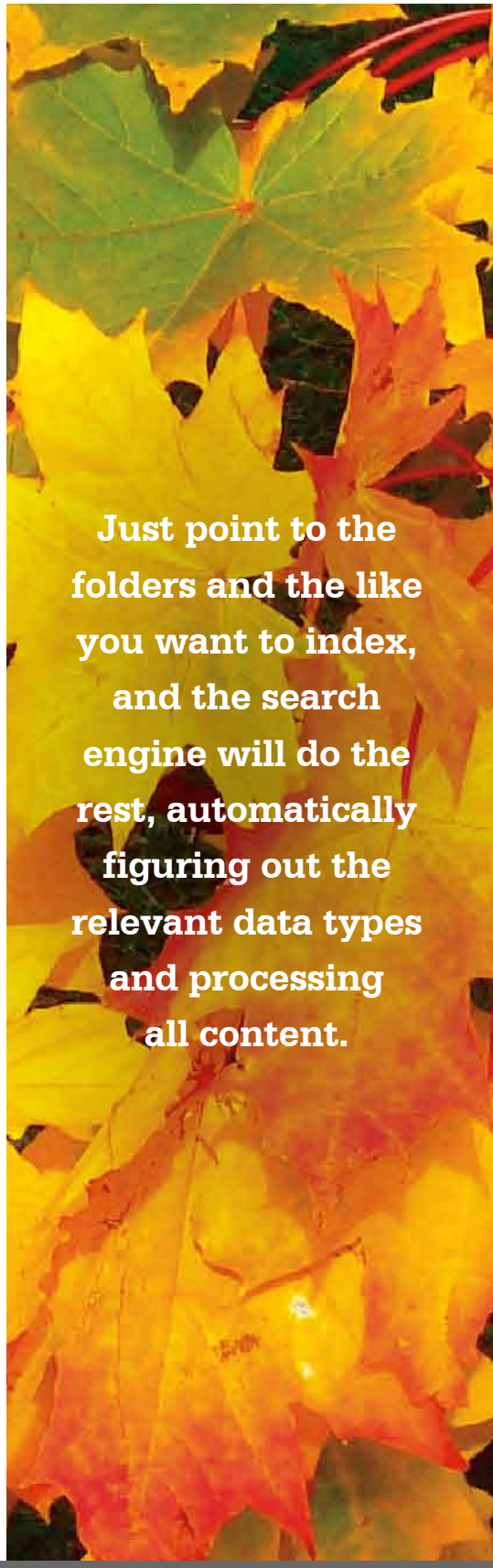
# National Clean Your Virtual Desktop Day: A Search Engine To Declutter Your Virtual Desktop

Article contributed by dtSearch®




**October 17th marks national clean your virtual desktop day – a day dedicated to backing up files, reorganizing folders, and deleting content you might not need anymore ... Skip the celebration altogether by utilizing a full-text search engine that allows you to instantly sift through terabytes of online and offline data without organizing a thing.**

A staggering quantity of data can pile up in a single year. Over many years, accumulated data can reach epic levels. As this holiday suggests, you can go file by file, email by email and the like, meticulously sorting every item. Or you can instantly cut through terabytes with a search engine.

A seasonal analogy may better contextualize the two options. Suppose you have a large pile of fall leaves. You could sort the large pile into red, yellow and green sub-piles. As each of these three sub-piles might still have an unruly mass, you could undertake additional sorting, dividing each of the color sub-piles into smaller piles according to leaf size or tree type.

Effectively, you would be sorting the piles using leaf metadata: color, size, tree type, etc. You can use the metadata to limit a search. Rather than examining each leaf in the original leaf pile, you could decide to only comb through the sub-pile containing large, red maple leaves.

## A Search Engine For Your Virtual Desktop

What if you could instantly navigate to anything in the original leaf pile,  no cleaning or sorting required? That is analogous to how a search engine works. To enable instant navigation, the search engine builds a search index across all the data: files, emails, web-ready data, etc.

Of course, the index holds metadata. With emails, for example, the index would store information like author, recipients, date, and subject. But the index would also contain the full text of each item as well. For emails, that would include the full body text along with the content of any email attachments.

A search engine has to figure out what type of data it is working with, as each file format requires a different parsing specification, some hundreds of pages long. You might think that the search engine could determine the format by filename extension: .DOCX would suggest a Microsoft word file and .PDF would suggest a PDF file. But that method would hardly be foolproof as it is possible to save a Microsoft word file with a .PDF extension and a PDF with a .DOCX extension.

The more reliable method of determining file format is to go directly to the binary content of each item. This is not necessarily a simple process. Emails can have ZIP or RAR compressed attachments with any mix of document types. And inside any single file, a secondary nested file may lurk, such as an access database embedded inside an excel spreadsheet. The search engine has to resolve all of that.

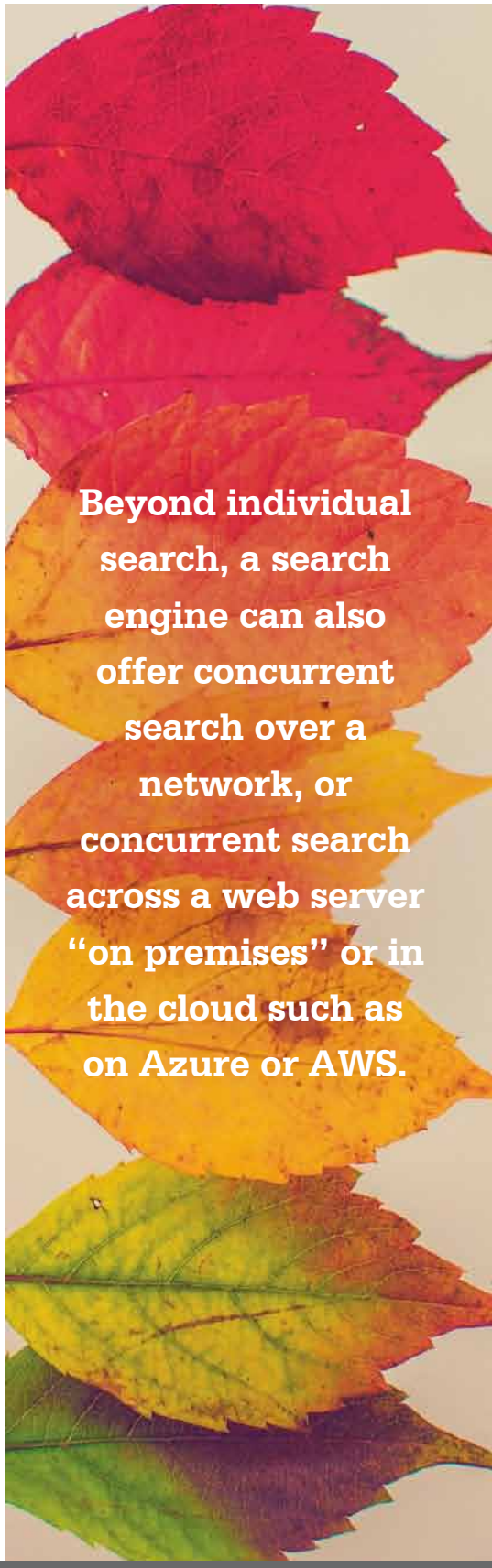
## Indexing for Easy Data Retrieval

At this point, you are probably realizing that indexing takes a lot of work, and wondering if pre-sorting items yourself along the lines of the mini leaf piles might not be a bad idea. But while indexing is a lot of effort, it is work for the search engine, not you. Just point to the folders and the like you want to index, and the search engine will do the rest, automatically figuring out the relevant data types and processing all content. A single index can hold a terabyte, and there are no limits on the number of indexes the search engine can create and instantly search at once.

Because the index holds all text in addition to metadata, search syntax can seamlessly combine full-text and metadata criteria. That way, you might choose to limit a search to files with only specific metadata criteria. Or you could cover everything regardless of metadata attributes.

In fact, indexing can enable over 25 different search options: words and phrases anywhere in files; positional search (words or phrases only at the top or bottom of a file or only in specific metadata); Boolean search (and/or/not); proximity search (a word or phrase within X words of another word or phrase in either direction or only prior to another word or phrase); concept search (extending a query to dictionary-defined or end-user-defined synonyms); wildcard search; regular expression search; fuzzy search (adjustable from 1 to 10 to sift through potential typographical errors in OCRed text or in emails, for example); and many other word-oriented search options.

Article contributed by dtSearch®



Article contributed by dtSearch®

A search engine can work with not only English text, but all languages covered by the Unicode standard. This includes double-byte character languages like Chinese, Japanese and Korean; right-to-left languages like Hebrew and Arabic; and even historical languages like ancient Egyptian hieroglyphics. A search engine can also search for specific numbers, numeric ranges, specific dates, date ranges, and hash values. The search engine can even find any credit card numbers that may appear in data.

Sometimes when you do a search, you'll immediately land on just the right item. But sometimes, the search engine will return a huge number of items in search results. In that case, the search engine has multiple options for relevancy ranking, as well as for generally sorting and instantly re-sorting search results. After a search, the search engine can display retrieved items with highlighted hits. For items that may be temporarily unavailable, a caching option can store the full text inside the index for a convenient hit-highlighted display.

## Closing Thoughts

But what if the data you need to search goes beyond your desktop? What if you also need to make available content on your organization's network or information residing online? And what if multiple people need to search the data at once?

Beyond individual search, a search engine can also offer concurrent search over a network, or concurrent search across a web server "on premises" or in the cloud such as on Azure or AWS. Unlike indexing, which tends to be resource-intensive, an online search can operate in a stateless multithreaded manner, remaining instant no matter how many search threads run at once. Efficient concurrent search can even continue while indexes automatically update themselves to accommodate new, modified and deleted data.

At this point, you may be wondering: Ok, but who else is going to see my leaf pile? The answer is only those end-users for whom you securely enable searching. A search engine like dtSearch does not send data back to the "mother ship," either at the time of indexing or searching.

This fall, use a search engine to streamline your data retrieval, and save the cleaning and organizing for your Halloween candy stash and Thanksgiving table.