

Extending Local Text Search to Cloud Data

Article contributed
by [dtSearch®](#)

For those who aren't familiar with dtSearch®, what does dtSearch do?

dtSearch has enterprise and developer products that run “on premises” or on cloud platforms to instantly search terabytes of “Office” files, PDFs, emails along with nested attachments, databases and online data. Because dtSearch can instantly search terabytes with over 25 different search features, many dtSearch customers are Fortune 100 companies and government agencies. But anyone with lots of data to search can download a fully-functional 30-day evaluation copy from [dtSearch.com](#)

What is today's topic?

A common question we get at dtSearch is whether dtSearch products can handle cloud-based data like files from OneDrive, DropBox and other cloud services as well as documents synced from SharePoint. The answer is yes, dtSearch can handle such data in a similar fashion to how it handles non-cloud content, as long as the data is visible through the Windows file system. Let's start with how a search engine like dtSearch handles data generally.

How does a search engine generally work?

An enterprise search engine instantly searches terabytes after first indexing the data. An index stores each unique word and number in the data and its location in the data. To start indexing, just point to the folders to index and the search engine will do everything else. In doing so, the search engine will iterate over all of the binary formats, identifying each file type so that it can correctly parse and index all text and metadata.

Do you have to identify the file types for the search engine?

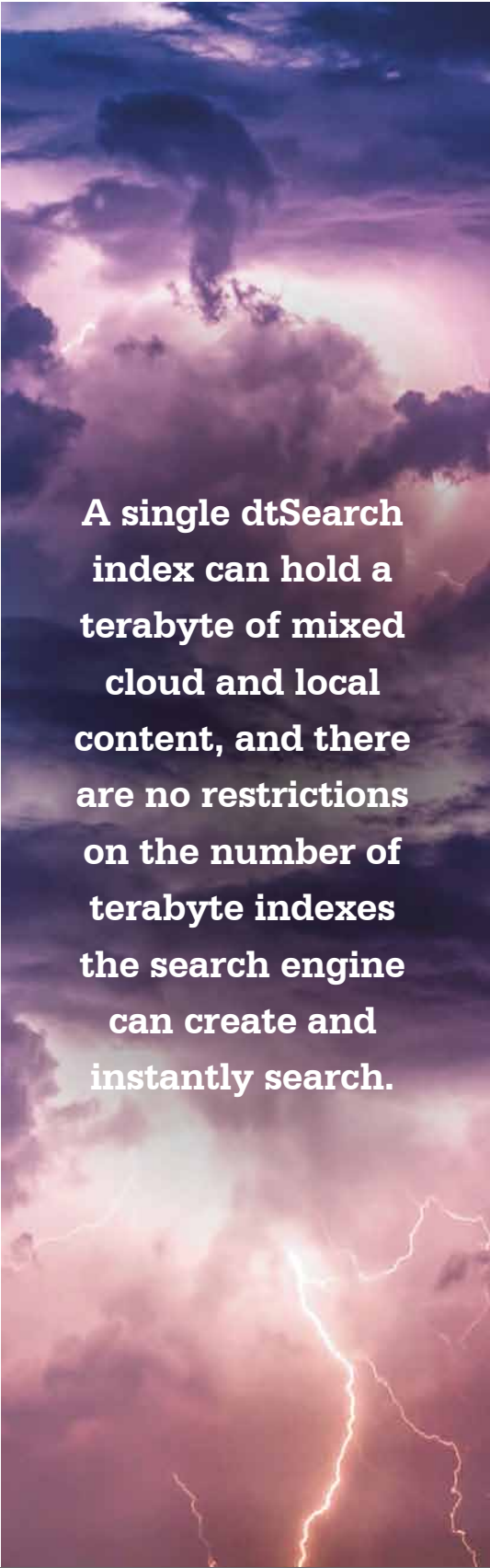
The search engine can determine the applicable file type from the binary formats themselves. Looking inside a binary format, the search engine can figure out if it is an email file like Outlook or Exchange; an Office format like Word, Access, Excel, PowerPoint or OneNote; a PDF; a web-based format; etc. A mismatched file extension will not trip up the search engine since it looks inside the binary format to determine the file type without reference to the file extension. The search engine can also detect and work with compressed archives like RAR or ZIP. And it can automatically drill down through multiple embedded layers like an email with a ZIP attachment that contains an Excel spreadsheet which itself contains a Word document.

So how does this extend to cloud formats?

The same process of adding folders to index that works for local content also works for cloud storage like files from OneDrive, DropBox and other cloud services as well as documents synced from SharePoint, as long as these are visible through the Windows file system. A single dtSearch index can hold a terabyte of mixed cloud and local content, and there are no restrictions on the number of terabyte indexes the search engine can create and instantly search.

How is all this made available for searching?

After indexing, search can proceed on an individual basis or a multithreaded concurrent basis. Concurrent search threads can



A single dtSearch index can hold a terabyte of mixed cloud and local content, and there are no restrictions on the number of terabyte indexes the search engine can create and instantly search.

operate independently, with no built-in limits on the number of instant network or online search threads that can run at once. As new content comes in, the search engine can automatically update indexes to reflect the new content without impacting instant concurrent searching.

So what types of searches can dtSearch so?

Unstructured natural language queries are the simplest searches. Just type in a few keywords or copy and paste a passage into the search box. Select “any words” natural language searching to find any files matching at least one term in the query. Select “all words” natural language searching to retrieve only files that contain every term in the query. Either way, the default relevancy ranking will take you straight to the most relevant files. For example, suppose you search for *accountant*, *doctor* and *lawyer*. If *accountant* and *lawyer* are prevalent throughout indexed files, but the word *doctor* only pops up in a few places, then *doctor* would get a higher relevancy ranking. And files with the densest mentions of *doctor* would get the highest relevancy ranking. But you probably wouldn't want to stop there in your search.

How would you extend that search?

Add in concept searching to get synonyms like *comptroller*, *physician* and *attorney* in a query for *accountant*, *doctor* and *lawyer*. For greater precision, enter a structured word and phrase search request like *GAAP accounting and (legal fees or licensing charges) and not surgery*. You can also add in proximity elements like *asset depreciation w/23 amortization*. You can limit one or more elements to metadata, like requiring asset depreciation w/23 amortization to appear in subject metadata specifically. Activate fuzzy searching to sift through minor typographical and OCR errors. That way, if *amortization* is mistyped as *amorqization* in an email, or an OCR'ed PDF has a similar misspelling, fuzzy searching will still pick that up.

Any other ways to extend a search?

In addition to word-based queries, the search engine also supports numeric-oriented searches. You can add a specific number or number range to a search request. Or you can add a date element like *March 16, 2022* or a date range like *March 10, 2022 to April 13, 2023*. The search engine can also automatically find date variants like *3/16/22*. Credit card search can find specific credit card numbers—or identify any credit card numbers anywhere in indexed data.

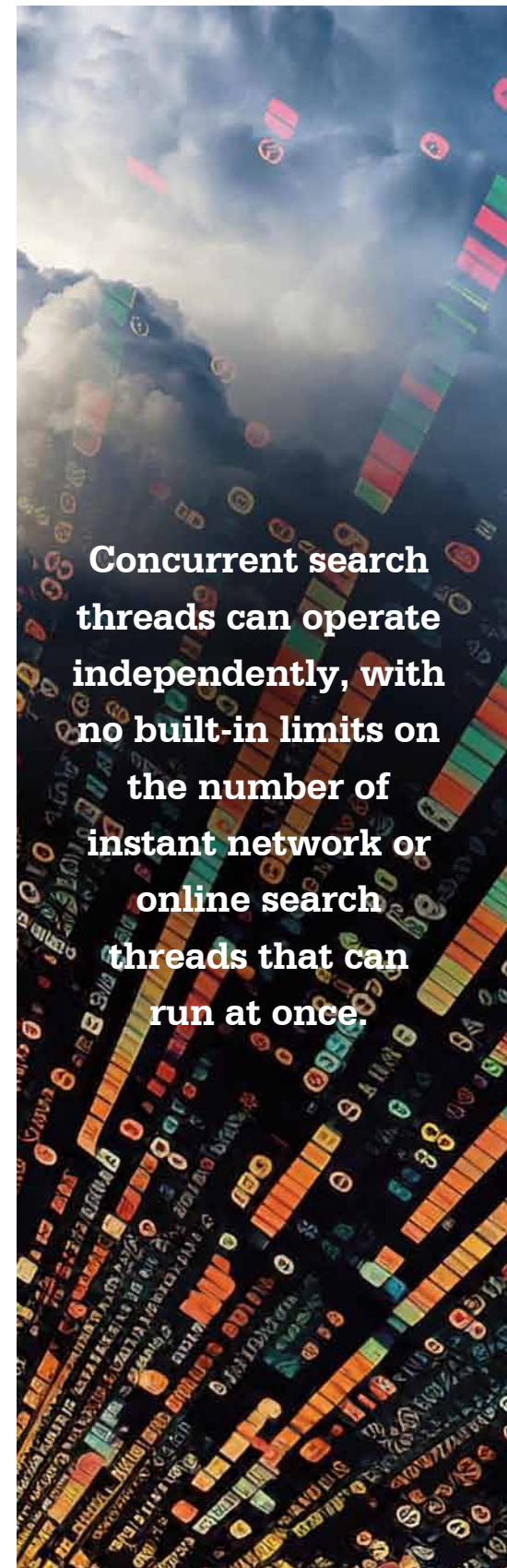
Does relevancy ranking work the same for a structured search as an unstructured search?

Yes, the same vector space relevancy ranking can apply to both structured and unstructured search requests. You can also apply variable term weighting on top of the default relevancy ranking. For example, instead of adding a *not surgery* element to a structured search request, you could simply give any file that mentions *surgery* a negative weighting. Or you could limit that negative weighting strictly to specific metadata occurrences or positional appearances at the top and bottom of files. For a new window into search results, instantly re-sort search results by some completely different criterion like file name, file location or file date. Regardless of the type of sorting, the search engine will display a complete copy of retrieved files and emails with highlighted hits for easy browsing.

Final thoughts?

dtSearch.com has a fully-functional 30-day evaluation download so you—and everyone else you work with—can try instant searching across all your data.

Article contributed
by [dtSearch®](#)



Concurrent search threads can operate independently, with no built-in limits on the number of instant network or online search threads that can run at once.