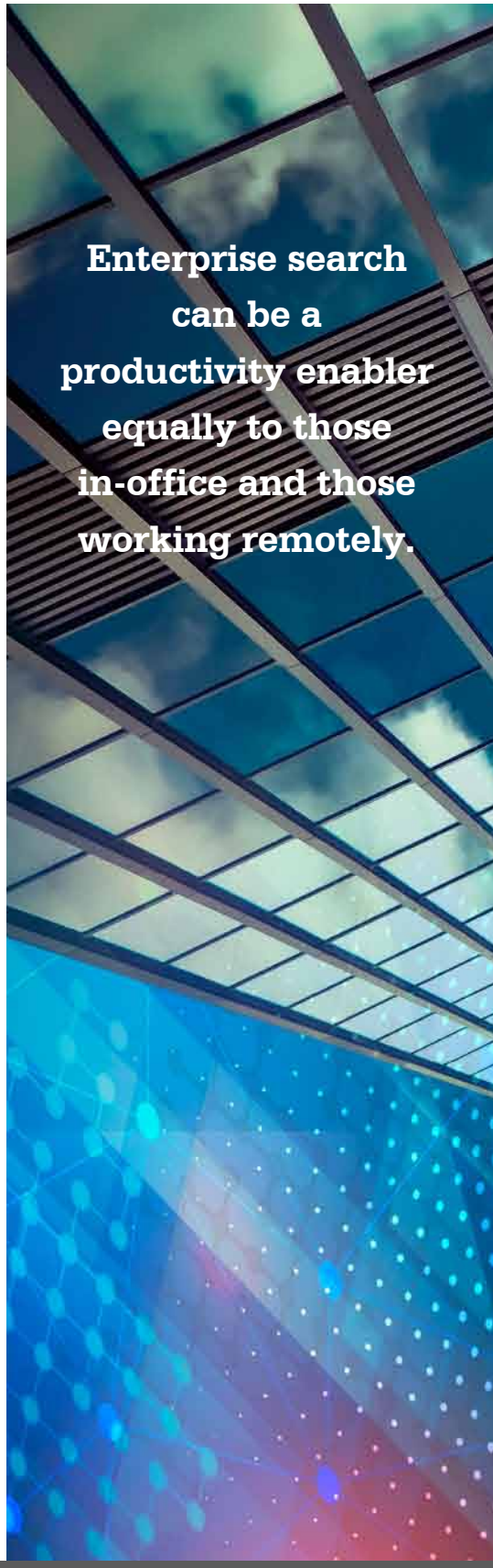
Enterprise Search and the "Back to the Office" Debate

Article contributed by dtSearch®



As the "should employees go physically back to the office" debate rages on, enterprise search can be a productivity enabler equally to those in-office and those working remotely. The key is indexed search.

When you randomly rifle through files looking for certain keywords, that is a basic unindexed search. In contrast, in indexed search, the search engine first builds what is effectively a treasure map to the data. The index stores each word and its location in the data. Having an index enables instant search, even across terabytes and even for those working remotely.

To start indexing, just point to the folders you want to index and leave the rest to the search engine. An index can hold up to a terabyte of text and there are no limits on the number of indexes that the search engine can build and simultaneously search. In building its index, the search engine will go through all files as they sit there in their binary formats, rather than pulling up each one in its native application.

Knowing the exact file type—Word, Access, Excel, PowerPoint, OneNote, PDF, email formats, web-based formats, etc.—is essential for the search engine to apply the correct parsing specification. However, the search engine can figure out the file type on its own from the binary file, without even referencing the file extension. And that is a good thing, as it is all too easy to save a Word document with a .PDF extension, or vice versa.

The search engine can make available multithreaded concurrent searching for everyone, whether in-office or out-of-the-office through a WAN or web-based connection. Online search can run from an "on premises" server or from the cloud such as Azure or AWS. From a technical perspective, online search can operate in a stateless manner, with no built-in limits on the number of search threads.

With regard to data updates, the search engine can update indexes automatically using the Windows Task Scheduler. In doing so, the search engine checks which files have been added, modified or deleted since the previous index run, and limits reindexing to the new content. Updating an index does not block out local or remote searching, so both in-office and out-of-office concurrent search can continue without interruption.

Another feature to facilitate remote data access is caching. After an indexed search, the end-user can click on an item to retrieve a full copy of that item with highlighted hits for convenient browsing. That works great if the underlying data is immediately accessible. But if the underlying data is not immediately accessible, a search engine can still retrieve a copy with highlighted hits with caching enabled.

Caching stores the full content of items in compressed form along with the treasure map part of the index. Caching makes indexes a lot larger – about the size of the original data. But it allows a search engine to immediately display everything with highlighted hits even if the original data is slow to access or simply gone.

Turning to searches in-office and remote users can run, the most basic type of searching is natural language or unstructured search requests. This category includes "all words" and "any words" searches. "All words" looks for files, emails and the like that contain all of the words in a search request like fall, leaves, rake. "Any words" would take the same fall, leaves, rake search request and look for files that contain any one of these three terms.

By default, the search engine will rank search results by relevancy, using a vector-spaced ranking model. The rarer the term in the indexed data, the higher its relevancy ranking. If fall and leaves appear in thousands of files but rake makes its way into just a handful of files, then files with rake would get a higher ranking. And files with the densest mentions of rake along with fall and leaves would get the highest relevancy ranking, letting those running a search go straight to the most pertinent files.

Structured search requests enable even more precision searching: (leaves falling or pumpkin pie) and fall season raking and not peppermint stick. Structured search can also include proximity elements like requiring mocha within 37 words of chai latte, or mocha within 6 words before chai latte. Structured search can also require that certain terms appear in specific metadata.

Structured search can also expand a search request to variants of the same route term like rakes and raking for rake, as well as looking for synonyms or similar concepts. A search can further look for numbers or numeric ranges as well as dates or date ranges, even across multiple date formats. All this increases productivity for employees both in-office and working remotely by letting them seamlessly hone in on the right data.

There is one more important type of searching within a shared data pool. A search engine can flag credit card numbers in a dataset. In doing so, the search engine takes any X digits that might represent a credit card number and runs those digits past a credit card checker to see if it is in fact a credit card number. Armed with that knowledge, those maintaining the dataset can take steps to eliminate any credit card numbers from the searchable data.

About dtSearch®. 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

Article contributed by dtSearch®

All this increases productivity for employees both in-office and working remotely by letting them seamlessly hone in on the right data.