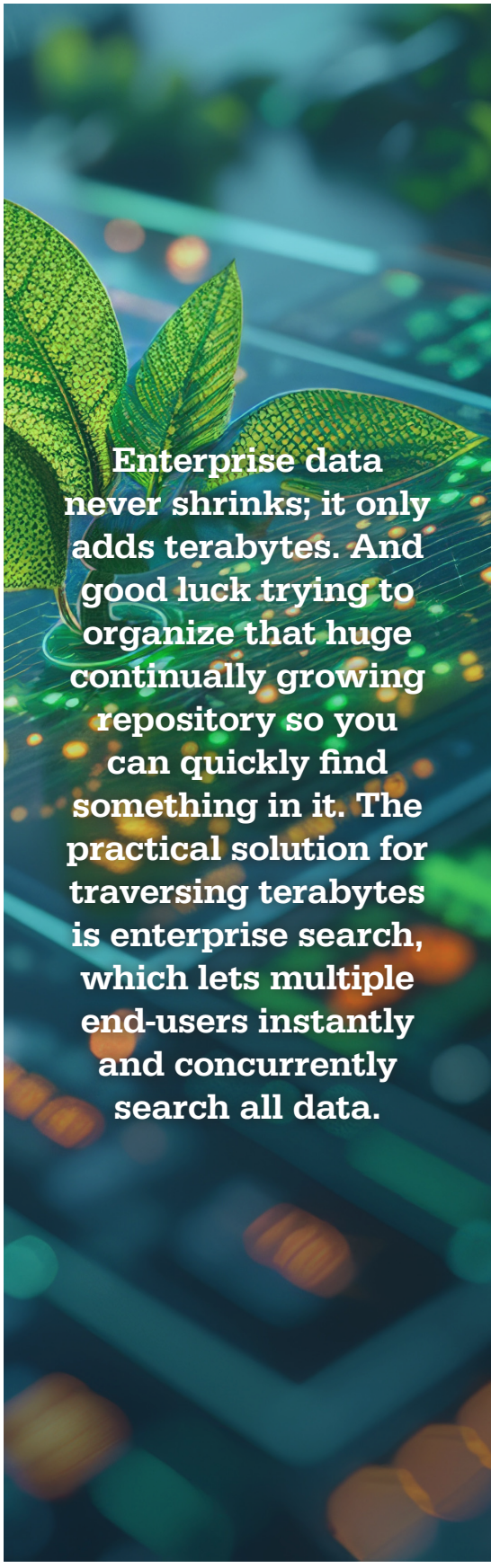
NO to a Spring Clean, YES to a Search Engine for Enterprise Data

Article contributed by dtSearch®



Spring is finally here. For your enterprise data, that means time to say NO to a spring clean and YES to a search engine. Enterprise data never shrinks; it only adds terabytes. And good luck trying to organize that huge continually growing repository so you can quickly find something in it. The practical solution for traversing terabytes is enterprise search, which lets multiple end-users instantly and concurrently search all data. But to enable this type of immediate and simultaneous access, enterprise search first has to index the data.

Indexing is different from organizing. Indexing does not move around content. In fact, a key feature of indexing with an enterprise search product like dtSearch® is that it doesn't alter enterprise data at all. The indexing process figures out each unique word or number and the location of each in the data without disturbing the data. Now, if you were going through files, you would retrieve each in its associated application – Microsoft Word, Excel, Access, PowerPoint, OneNote, Outlook/Exchange, Adobe Acrobat Reader, etc. But retrieving each file in its associated application would make enterprise search indexing way too slow.

Binary format access. To speed things up, the indexer bypasses retrieving each file in its associated application and heads straight to each file's binary format. While the indexer needs to know the exact right file type to correctly parse each file, the indexer can figure this out from the binary format regardless of a file's extension. That way, a Word document with a .PDF extension or a PDF with a .DOCX extension won't gum up the works.

Binary formats further let the indexer parse both individual standalone files and nested files. Using binary formats, the indexer can parse an email with a ZIP or RAR attachment containing an Excel spreadsheet with a

Word document embedded inside the Excel file. Through binary formats, the indexer can even identify "image only" PDFs, letting you know that these files require an OCR program like Adobe Acrobat for full-text processing.

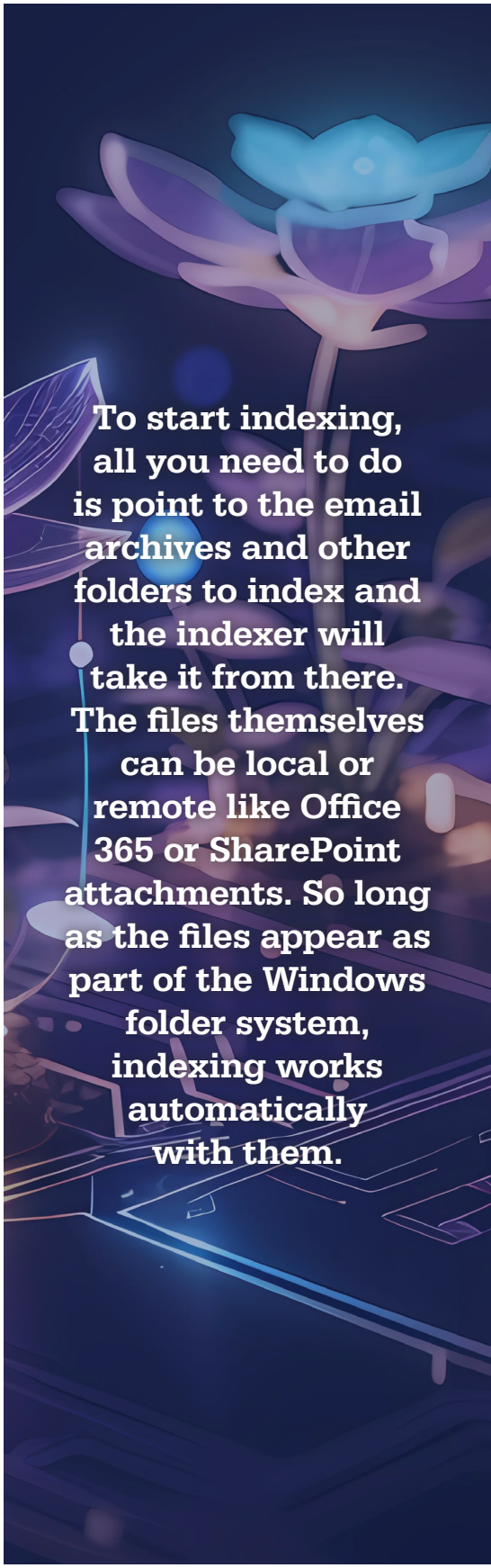
Binary formats also make available text that might be hard to spot in an associated application view. Some examples follow. You can click around a file extensively in its associated application and still miss key pieces of metadata. But all metadata is right there in the binary format for the indexer to parse. Also, certain text like white text against a white background or blue text against a blue background can hide in an associated application view. But all text is on the same footing in the binary format.

Some redaction programs can block out certain text, but the text itself can continue to exist invisibly under a black rectangle. Similarly, with tracked changes not fully accepted, text can appear deleted while actually remaining as part of the overall file. But in both the redaction and the tracked changes examples, all such text is fully apparent in the binary format.

Indexing is easy. To start indexing, all you need to do is point to the email archives and other folders to index and the indexer will take it from there. The files themselves can be local or remote like Office 365 or SharePoint attachments. So long as the files appear as part of the Windows folder system, indexing works automatically with them. A single dtSearch index can hold up to a terabyte and there are no limits on the number of indexes that the software can create and end-users instantly and simultaneously search. Concurrent searching can proceed in a classic network environment, from an on-premises web-based server, or from the cloud like AWS or Azure. And updating an index to reflect new, modified or deleted content does not halt instant concurrent searching.

A multitude of search options. Rummaging through a closet, you typically ask yourself a binary question: junk or not junk? By contrast, indexed searching has over 25 different full-text and metadata options ranging from "free form" natural language to high-precision phrase, Boolean (and/or/not) and proximity formulations. Concept searching extends a search term to similar

Article contributed by dtSearch®



concepts. Fuzzy searching adjusts from 1 to 10 to comb through typographical and OCR errors.

Unicode searching extends to hundreds of international languages that the Unicode standard covers, including right-to-left languages and double-byte Asian text. In addition to word-based searching, the software also supports number and numeric range searching, date and date range searching extending automatically to popular date formats, and hash value generation and search. dtSearch even has an option for flagging credit card numbers hiding in data.

File ranking and display. By default, dtSearch will relevancy rank retrieved files based on the density and rarity of search terms across indexed data. Take an "any words" query for *CorpABC acquisition of TargetXYZ*. If *CorpABC* and acquisition are all over indexed data but *TargetXYZ* appears in just a few places, *TargetXYZ* will get a higher relevancy rank. Files with the densest references to *TargetXYZ* will come out on top.

End-users can also add their own variable term weighting, assigning a positive or negative weight to search terms anywhere, or just to appearances in certain metadata or towards the top or bottom of files. For a fresh view, the software can instantly re-sort by a separate criterion like filename or file date. Whatever the sorting, dtSearch can display a full copy of retrieved files with highlighted hits for convenient browsing.

In sum, nix the spring clean and go straight to enterprise search for instant concurrent searching across terabytes of offline and online enterprise data. Get started at dtSearch.com with fully-functional 30-day evaluation downloads.

Article contributed by dtSearch®