

Still Working in a Hybrid Environment? Why It's Important to Consider Your Data from a Search Engine's Perspective

Working in a hybrid environment, you may feel like you want to go the “extra mile” to ensure maximum productivity. Of course, that mean leveraging a search engine to instantly find that critical nugget of information you need to get something done, whether that nugget resides somewhere in the entire enterprise dataset or simply in your own emails. But to the extent you are working remotely, you are also your own data administrator. For that reason, it is important to understand how a search engine sees your data and to use that knowledge to recognize possible data pitfalls.

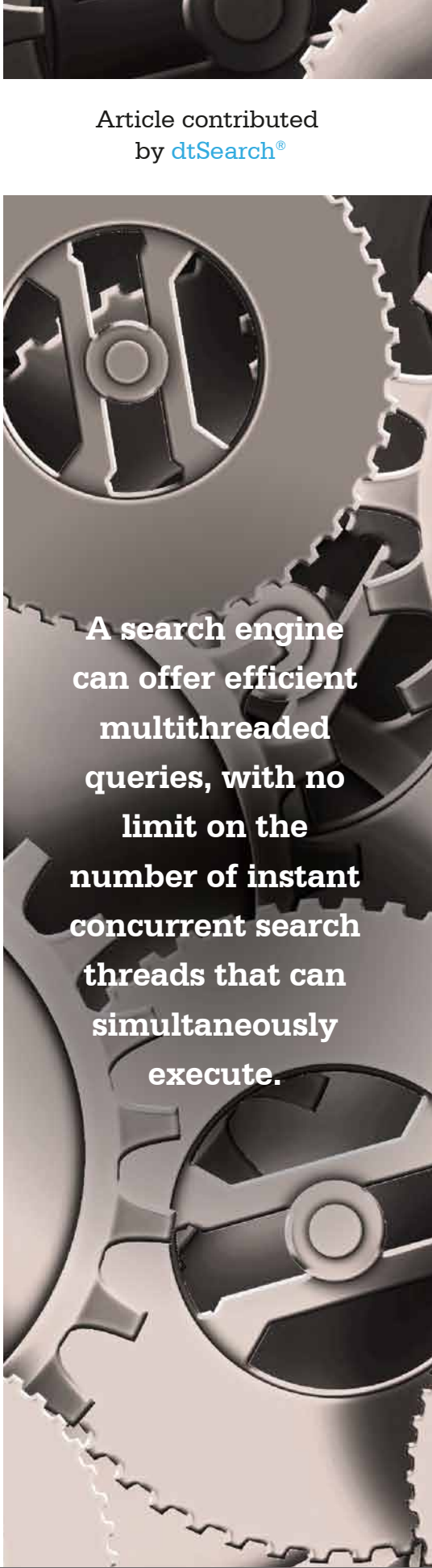
How a Search Engine Works

Concurrent vs. individual search. Your organization can use a search engine running in a web-based environment or a classic network environment to instantly search terabytes. (This can be done using an enterprise search – text retrieval software like dtSearch®, not “scouring the Internet” search like Google.) Running in an online or a network environment, such a search engine can offer efficient multithreaded queries, with no limit on the number of instant concurrent search threads that can simultaneously execute. But while enterprise search is the gold standard for organization efficiency, you can also run a search engine individually from any computer running under Windows.

Instant search across terabytes requires indexing. Whether a search engine is running enterprise-wide or just from an individual computer, the search engine essentially works the same way. The search engine can only instantly search terabytes after first indexing the data. Indexing pre-processes the text, identifying each unique word and number in the data and its location in the data.

To start indexing, just point to the folders, email repositories and the like to index and the search engine will do everything else. The indexer can cover local files or cloud-based data like OpenOffice files that appear inside the Windows folder system. Whether the data is cloud-based or local, a search engine will look at each item in its binary format.

Article contributed
by dtSearch®



A search engine can offer efficient multithreaded queries, with no limit on the number of instant concurrent search threads that can simultaneously execute.

The binary format world. You're typically used to how files appear in their associated applications, like Microsoft Word, Access, Excel, PowerPoint, OneNote, Outlook/Exchange; Adobe Acrobat Reader; or for web-based formats from inside a browser. But if you ran across a file in its binary format, you might have trouble reading any text at all amidst the binary codes. The search engine must correctly recognize each specific file format, and then apply the correct parsing standard to make out all text and metadata.

Parsing specifications can be hundreds of pages long, and applying a Word parsing specification to a PDF format would not be pretty. The search engine not only has to recognize and parse individual file formats, but also container formats, like emails with RAR or ZIP attachments. Fortunately, in today's world, multilingual text is no problem. Unicode encodings supporting hundreds of international languages directly carry over into the binary format version of files.

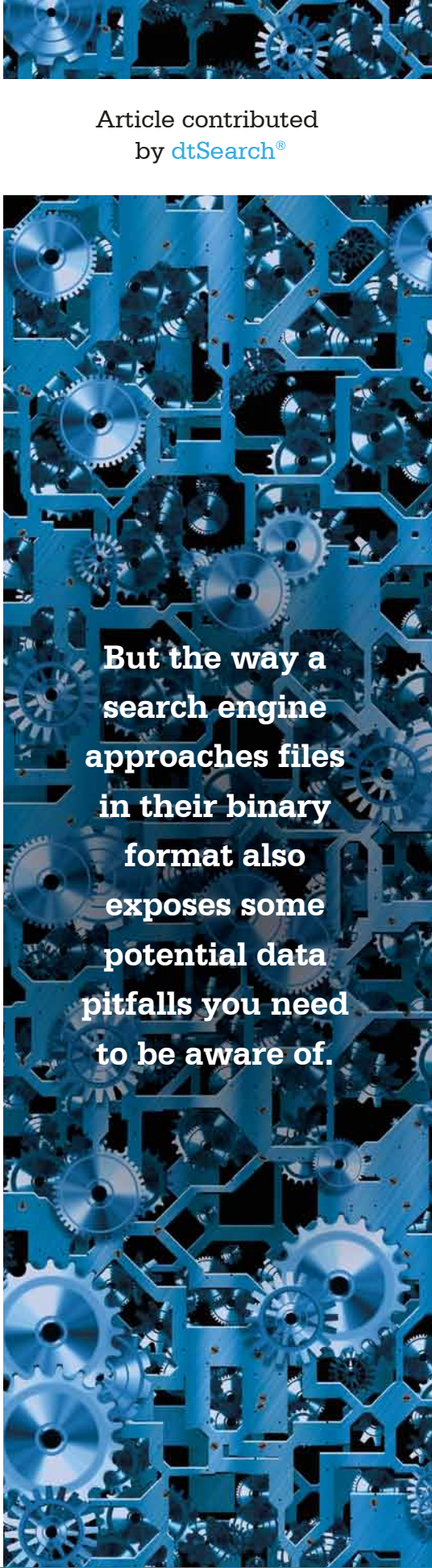
The end result. Once the search engine finishes indexing, it can instantly search terabytes. A search engine has multiple options for item ranking and sorting, as well as instant re-sorting of retrieved files. After a search, the search engine can display a full copy of retrieved files with highlighted hits for convenient browsing. As data changes, the search engine can automatically update its indexes without interfering with continuing individual or concurrent searching.

A search engine offers dozens of search options for precision data retrieval, ranging from unstructured natural language queries to highly structured Boolean (and/or/not), proximity, concept, etc. search requests. It can also further accommodate numeric search requests, like searches for specific numbers or numeric ranges, dates or data ranges (even automatically extending across different date formats). Lastly, search engines can even find numeric patterns, like identifying file hashes or finding any credit card numbers that might reside in data.

Potential Data Pitfalls

Understanding your data. It is easy to see how a search engine increases productivity by letting you instantly find whatever data you need to move forward. But the way a search engine approaches files in their binary format also exposes some potential data pitfalls you need to be aware of, particularly to the extent that you are working on your own.

Article contributed
by [dtSearch®](#)



But the way a search engine approaches files in their binary format also exposes some potential data pitfalls you need to be aware of.

Pitfall #1: Files inside other files. Current “Office” file formats let you nest one file inside another, like embedding an Excel spreadsheet inside a Microsoft Word file. When you look at the “outer” file in its associated application, only a portion of the embedded file might be immediately visible. But the whole nested structure is fully available in binary format and hence to a search engine.

Pitfall #2. Misplaced file extensions. As noted above, applying a Word parsing specification to a PDF format would not be pretty. Using that logic, it might seem that saving a Word file with a .PDF extension or a PDF with a .DOCX extension might obscure the text. But the search engine not only uses the binary format to discern the text and metadata, but also to determine the file format of the item itself. So while an associated application might “trip over” an erroneous file extension, such a mischaracterization will not affect a search engine.

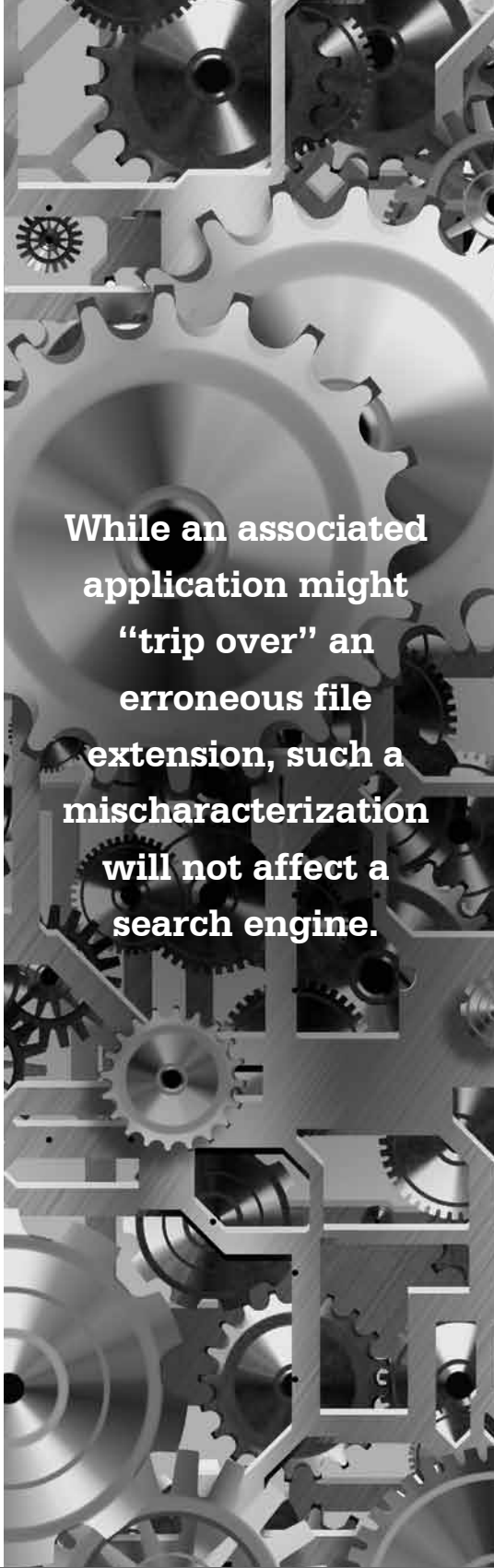
Pitfall #3. Text that blends in with the background color. White text against a white background, black text against a black background, orange text against an orange background, etc., can camouflage such writing in an associated application view. But all text is on the same footing in the binary format version that a search engine works with.

Pitfall #4. Obscure metadata. Certain metadata can hide in an associated application view of a file, potentially requiring tons of directed clicks before you even discover that it is there. But all metadata is equally accessible to a search engine from inside a binary format.

Pitfall #5. “Image only” PDFs. Have you ever run across a PDF that looks like any other PDF initially as it sits here in the file system, but when you try to copy and paste a passage from it no text appears? A search engine can identify such PDFs, letting you know you need to run them through an OCR application like Adobe Acrobat to turn them into “searchable image” PDFs.

Pitfall #6. Typos. Typographical errors in emails and in OCR'ed text are not uncommon. Fuzzy searching, adjustable from 0 to 10 can sift right through such typographiGal errors.

Article contributed
by [dtSearch®](#)



While an associated application might “trip over” an erroneous file extension, such a mischaracterization will not affect a search engine.