

2022: RIP Organizing Enterprise Data to Find Things

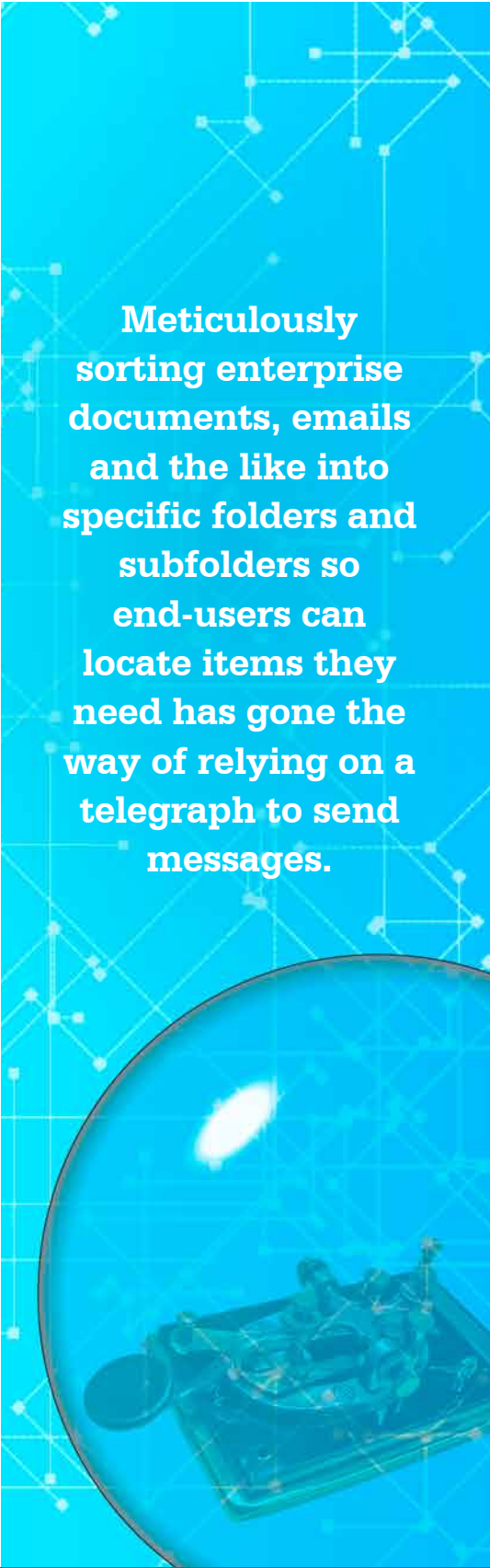
Meticulously sorting enterprise documents, emails and the like into specific folders and subfolders so end-users can locate items they need has gone the way of relying on a telegraph to send messages. A search engine enables multiple users to concurrently search across terabytes employing a wide range of instant search options - no prior human organizing required.

Note that these are not "search the Internet" search engines like Google. Rather, these are precision, enterprise search engines like dtSearch. An enterprise search engine instantly searches terabytes of data after indexing. Indexing takes no human effort: simply point to the high-level folders to cover, and the search engine automatically does the rest.

Some technical background on how a search engine operates might be helpful. When you retrieve a word processing document in Microsoft Word, the Word application takes the binary file and displays it through the lens of the Word application. In contrast, a search engine goes directly to the binary file, bypassing the application entirely.

A search engine first needs to figure out the correct parsing specification to apply to the binary file. .DOC has a completely separate parsing specification from .DOCX, and both are very different from the specifications for Outlook/Exchange, PDF, PowerPoint, Access, Excel, OneNote, etc. To figure out the parsing specification to apply, a search engine has to look inside the binary file contents, not at the file extension. (Saving a PDF with a Microsoft Word extension and an Access database with an Excel extension is all-too-easy.)

Article contributed
by [dtSearch®](#)



**Meticulously
sorting enterprise
documents, emails
and the like into
specific folders and
subfolders so
end-users can
locate items they
need has gone the
way of relying on a
telegraph to send
messages.**

But the advantage to a search engine of directly accessing binary files does not stop there. Metadata that may require extensive "clicking around" to see in an application view is immediately accessible in binary format. Black on black, white on white, or green on green text that may be hard to spot in an application view is "plain as day" in binary format. Even parsing multilayer nested files, like an email with a ZIP attachment containing an Access database with an Excel spreadsheet literally embedded inside, works better in binary format.

Organizing data in folders and subfolders typically relies on simple attributes like filename. By contrast, a search engine can instantly process over 25 different types of search requests. An end user can enter a "plain English" natural language search or craft a precision full-text or metadata search with expressions involving words or phrases in Boolean (and/or/not) or proximity relationships to other words or phrases.

Fuzzy searching can sift through spelling errors that can arise from OCR or simply mistyping in an email. Searching works not only with English, but any of the hundreds of international languages that Unicode supports. Beyond words, the search engine can search for numbers, number ranges or numeric expressions; dates or date ranges; and hash values. A search engine can even find credit card numbers in data.

The search engine has multiple options for relevancy ranking and other search results sorting (and immediate re-sorting), while displaying a full copy of retrieved items with highlighted hits. While a search engine can run on individual PCs, it can also enable concurrent searching across a network or an Internet or Intranet site. The site could be hosted "on premises" or hosted remotely in the cloud such as on Azure or AWS. Finally, an index can automatically update itself at specified intervals to accommodate content changes, without affecting continued concurrent searching.

So, as we enter 2022, RIP organizing enterprise data for the new year, and try a search engine.

Article contributed
by [dtSearch®](#)

**RIP organizing
enterprise data
for the new
year, and try a
search engine.**