Wolters Kluwer
Law & Business
Loislaw®
New features available

BALL IN YOUR COURT: All Wet

By Craig Ball

Send this to a colleague

Reprints & Permissions

I was once trial counsel for the water authority of a Mexican city seeking damages for delay in the mapping of a water system serving three million customers. I learned that most water entering the pipes never reached consumers because the patchwork system was riddled with leaks. The leaks were difficult to repair because the water authority didn't know where its pipes were buried!

Repair crews made Swiss cheese of streets, but the massive leakage limited water service to just a few hours a day. Those who could afford it erected tanks to hoard water. The rest suffered. Until Servicios de Agua y Drenaje learned where its pipes lay, staunched the leaks and addressed local hoarding, the system stayed broken. ¡Ay, caramba!

At the faucet, the thirsty señora didn't care how hard or costly it was to collect, filter and deliver the water. She couldn't tell the water company what reservoirs and wells to tap, purification techniques to employ or pipes to use to route the water. She certainly didn't want to hear that she didn't need the water or hadn't used the spigot correctly. She wanted a drink, and felt it should flow to her in a timely and adequate way.

A judge could have ordered the water company to pump, but the cost in terms of wasted agua would have been astronomical and unsustainable. Telling the consumer to, "Find your own water or do without," was likewise untenable.

An apt metaphor for e-discovery, don't you think?

Litigants harbor immense reservoirs of electronically stored information. Servers, like lakes and rivers, are evident and expansive. Databases and archives are vast subterranean aquifers. Information puddles in desktops, portable devices and online storage. It's costly to preserve, tap and process, and after all that effort, much is lost to leaky mains:

- We don't know where our pipes are buried (lax records management).
- We let sources evaporate and sour (poor preservation).
- We poison the well (spoliation).
- We use sieves to dip and dowsing rods to explore (careless collection and search).
- We fill the tub when a tumbler would do (over-broad requests for production).
- We bathe in Perrier (conversion of ESI to image formats for manual review).

Through education, cooperation and improved tools and techniques, these holes are slowly getting plugged. Good thing, too, because our thirst for electronic evidence is growing fast.

Still, there's a leak in the pipes that draws no attention. Sometimes it yields just a trickle, other times it's a gusher; but if we don't find and gauge the loss, how will it ever get fixed?

This leak is blind reliance on text extraction and indexing engines as principal tools of ESI search.

Many think of electronic search in linear terms — as something that surfs across the connected and collected sources of ESI comparing words and phrases to queries. Indeed, that's the way we search files on our computers and how computer forensic tools typically operate.

But most electronic data discovery search efforts aren't linear explorations. Instead, they run against an index of words extracted from the source data.

So, is that really different? Quite.

It may take hours or days to extract text and create the index, but once complete, searches run against indices are lightning fast compared to plodding linear search. That's the upside. But there's a noteworthy trade-off to using indices: you may not find what you seek, even though it's in the collection and you've chosen the right keyword.

Why? There are several reasons text extraction and indexing let data evaporate.

To start, text extraction tools parse data for sequences meeting the rules by which they define words. Is L33T a word? Is .doc a word? How about 3.14159?

A simple parser might define a word as, "more than four but less than 14 contiguous alphabetic characters flanked by a space or punctuation." Parsers also employ rules barring certain combinations. Numbers, most punctuation, and symbols are typically ignored, and common terms called "stop words" are sidelined, too. The very popular MySQL database excludes more than 500 common English words, and DTSearch excludes more than 120; so, Shakespeare buffs can forget about finding "to be or not to be."

A more insidious shortcoming flows from failure to include encoded text in the index.

ESI is encoded in many different ways, and encoded objects are often nested like Russian matryoshka dolls. Consider this frequent scenario: a Word document and a PowerPoint inside a Zip archive attached to an e-mail message within a compressed Outlook pst container file. Each nested object is encoded differently from its parent and child objects, and encoding may vary within the body of an object. Encoding is critical. In fact, next to metadata, encoding may be the most important thing many people don't understand about e-discovery.

When a parser processes encoded ESI, it must apply the appropriate filter to the data to convert it to plain text so it can be indexed. If the data is encoded in multiple ways, multiple filters must be applied in the correct sequence to cycle through all different forms of encoding to reach textual content. If no filter or the wrong filter is applied, the text isn't indexed. This occurs in various ways, e.g., the encoding isn't recognized, the tool doesn't support the encoding, the content isn't text or the file is corrupted, encrypted or password protected.

If a parser doesn't recognize the encoding, it may default to applying the most common textual encoding schemes to the unrecognized content in a last-ditch effort to find intelligible text. But that doesn't always work. Foreign alphabets employ many more than our paltry 26 letters. Ideographic languages like Chinese and Japanese don't separate words with spaces.

Even in English, you don't want to miss finding "résumé" when you search for "resume," so success hinges on whether the index is accent-sensitive or insensitive. Text parsers work around these challenges in various ways, but not all perform in the same way. There's many a slip 'twixt cup and lip.

Another reason data goes down the drain in text extraction and indexing is that failure is hard-coded into indexing applications. They're programmed to ignore file types deemed unlikely to hold text or apply only rudimentary text extraction methods. For example, Microsoft's Windows Search and Index Server have a limited capacity to index the contents of Access databases.

Finally, text extraction tools can't capture what they don't see as text. Facsimile or .tiff images are classic examples of text-laden documents not captured. These ESI sources, as well as documents storing text as vector graphics, must undergo optical character recognition to expose text. Although the same issue surfaces in linear search, you can subsequently run OCR against source data. You can't go back to the well with an index.

Maybe that's the ultimate failing of indices: they're just a shadow of the evidence. Because an index isn't the data, you can't apply new and better ways to wring out the truth.

Is it wrong to employ indexed searches in e-discovery? Certainly not, but it's all wet to select a tool to perform a task it can't accomplish. So, plumb the depths of your parser and indexer, then test them against representative samples of the data in the case and evaluate the search results for both data recovery and leaks. Be prepared to identify which encoded formats, file types and stop words are absent from the index.

In short, you need to know the capabilities and limits of the text extraction and indexing engines you deploy. Because if the index won't hold water, you're up a creek.

*Austin's **Craig Ball** is a trial lawyer and computer forensics/EDD special master.*
E-mail: craig@ball.net.

Law Technology News August 2009

Copyright 2009. Incisive Media US Properties, LLC. All rights reserved.