Globalization Gone Wild
Managing Foreign-Language Documents

by Gary Wiener of Liquid Litigation Management, Inc.

If you follow international news at all, then you already know how important the Far East has become to our global economy. It is said that China, for example, will soon pass the United States as the world's leading consumer nation. Japan's technological trailblazing has kept it a very powerful economic force for decades. South Korea, Vietnam and, of course, the Middle East are each becoming bigger players on the world stage, and all are creating a new generation of capitalists and consumers. And, as those of us in the United States well know, capitalists and consumers file lawsuits.

What we call "electronic discovery" is already posing quite a challenge to litigation practice. However, most in the legal profession have figured out that scanning and OCRing paper documents (or, more likely these days, extracting text from native ESI files) can be used to build an effective search index, which can significantly cull document review sets to a manageable, less-expensive size. But what about those evidentiary files that don't use the Latin alphabet that Western languages utilize? What about the 65,000 or so pictorial glyphs that make up Chinese, Japanese and Korean ("CJK") character sets? How about Middle Eastern languages, such as Arabic and Hebrew, that read from right to left? How does one build a search index when we're dealing with an entirely different method of written communication from what we're used to? And how can our English-language computers handle such a huge variety of characters and character sets?

More significantly for us in the legal profession, if we represent a client in litigation whose case involves millions of electronic documents in a foreign language, how the heck can we make sure that each document gets properly imaged and indexed? And how are we supposed to cull and search those documents?

ASCII and Ye Shall Receive

To create characters that we can read on our monitor, our computers read eight bits, or a "byte", together. In a string of eight binary numbers, there are 255 possible combinations of ones and zeros (in addition to all zeros, which equals "nothing"). For Western languages, each different binary sequence is assigned to a different character of the Latin alphabet. This easily provides enough sequences to cover 26 capital letters, 26 lower-case letters, ten numerals and a slew of special characters, accents, tildes and umlauts.

However, 255 combinations are nowhere near enough to handle the many thousands of characters used in CJK languages. A Chinese typewriter, for example, famously has well over 5,000 characters, and that's an incomplete character set used only for the Mandarin dialect. There are also two written forms of Chinese, Simplified (used in mainland China) and Traditional (used in Hong Kong and Taiwan). The Japanese use three different alphabets (including Chinese characters), all of which frequently appear in the same sentence. Similarly, Korean and Thai character sets have unique characteristics and many thousands of characters.

To get around the 255-character limitation, computers began to link two bytes together for each character, to create a "double-byte" sequence. These provide 256 times 256 options, or 65,536 possible combinations, an adequate number to handle the Far Eastern "double-byte" character sets.

Perhaps you've heard of Unicode, which is the attempt to represent all characters of all major languages in the world within a single character set. The current Unicode standard (called "UTF-8") strings four bytes together to represent a single character, which allows for 1,114,112 possible characters (the fourth byte is rarely used today, and when it is, only contains a "pointer" to tell the computer how to read the other three bytes).

Windows NT, XP and Vista, and Mac OS X are Unicode-compliant. These four operating systems are used by the overwhelming majority of law offices around the world today. So why does this still present us a problem for processing foreign-language documents electronically?

A Legacy of Headache

While the current major operating systems were written from the ground up to be Unicode-compliant, most document processing software packages were not. Most were written for previous versions of Windows or Mac (which were *not* fully Unicode-compliant), and were "ported" over to the current operating systems with minimal programming changes. Because the older operating systems had no trouble handling our Latin character set, these "legacy" software packages worked perfectly well with Western-language documents.

Try to extract a Japanese-language (for example) file into one of these legacy document processing programs, however, and the pitfalls of non-Unicode-compliant software become evident. Does the program recognize Japanese characters to begin with? Does the program acknowledge that the Japanese language can use several different characters to represent the same word? Does the program understand that Japanese writing occasionally uses two different Japanese "alphabets" and Chinese characters in the same sentence? And can the program switch character

sets on the fly to recognize those different character sets? Does the program know how to "tokenize" Japanese words to build a search index, without using the punctuation or spaces that English-language indexing uses? Can the program interpret a sentence in any direction other than left-to-right. and if it can, can the program properly interpret numbers or English words (left-to-right) within, for example, an Arabic (right-to-left) sentence?

These considerations are not limited to Japanese, of course. Chinese, Korean, Thai, Hebrew, Arabic, Cyrillic and other written languages all have their own rules for interpretation and tokenizing.

Perhaps you think it's up to your service bureau to resolve these issues; fair enough. So let's consider the two main issues that you, as a legal professional, are going to have to face when these documents come back from the service provider: Do the foreign-language characters show up within the documents just as they did in the original file? Can you run effective search queries on these documents?

The Wingdings Conundrum

As e-mail discovery continues to evolve as the backbone of evidentiary litigation practice, more and more of those e-mail messages are going to contain CJK characters. If you have the proper character sets installed on your computer, once those documents have been "petrified" (converted into a picture of the document as it would appear on your computer screen), all the characters should show up correctly, right? Consider this real-world example:

Perhaps, in the past, you have created a Word or WordPerfect document that contained some graphical fonts. Whether the font you used was called Wingdings, Dingbats or something else, the character set had lots of icons, pointers, small pictures and the like, in lieu of letters and numbers. When you inserted these characters into your document, everything looked just fine on your computer screen. If you e-mailed the document to someone else, though, some of those graphical characters might have been replaced on their screen by seemingly random letters or little boxes.

What happened? Quite simply, you had the proper character sets installed on your computer to create the "look" that you wanted but your recipient did not. The operating system could not find the character set that you used to create the document, so it substituted characters from your default font. Where the same binary sequence for the missing font was assigned to a character that existed within the default font, the computer substituted the corresponding character. Where there was no corresponding character, the computer added a "box" to show that a corresponding character couldn't be found within the character set.

Substituting character sets is easy for the computer to do when we're dealing with Western fonts of single-byte origin, like Arial or Helvetica or Wingdings. Odds are, all of the possible characters you would use are going to be somewhere within that 255-sequence assignment table. But if the correct character sets are missing from a computer that tries to open up a Unicode-based document, such as a Chinese-language e-mail, the computer cannot find characters within your default font to substitute. The viewer will be left looking at long strings of boxes and gibberish that bear no relation to the contents of the original document.

Let's apply this to the processing of electronic documents. When petrifying ESI into a readable form, the processing software literally "prints" the document into a graphic image, so that someone looking at the document should see it the way it appeared on the computer screen of the person who created it. As the term "petrification" suggests, however, once the document has been "printed," it is in its final form. The computer that petrifies the documents must therefore have all of the appropriate character sets installed in order to "print" an accurate representation of the document. That way, anybody who opens the petrified document — whether they have the foreign-language character sets installed on their computer or not — should be able to view the document as it was originally created.

A service bureau that does not use software that can properly handle Unicode-based electronic files (let alone that does not have all of the correct character sets installed on their processing computers) will return to you documents full of garbage. None of them will accurately represent the contents of the original documents, and none of them will be admissible in court. Neither you nor your opposing counsel will be pleased with the results; and, more significantly, neither will the judge.

Making a Token Effort

Running a search query on a Western-language document is fairly straightforward. Because written Western words are simply strings of characters bracketed on both sides by punctuation or a blank space, a computer can easily parse the words and look for matches in a full-text search. Indeed, most document management programs create an index, which (like a word index at the end of a deposition transcript) cross-references every word in the document universe to each file and location where it appears. Running searches on an index means that the search tool only has to search one document — the index — rather than comb through the full text of every document in the search universe.

Oh, if it were only that simple for Far Eastern languages.

CJK "characters" do not correspond to English "letters." Although both are glyphs (graphic representations of characters, sometimes called logograms or ideograms), CJK languages use a separate character for each syllable, but the syllables are based on written dialectic concepts rather than spoken sounds. These characters are further strung together without the use of spaces, usually ending a sentence (if at all) only with a punctuation mark. Consequently, there are no blank spaces that a search engine can use to parse and recognize individual words.

What we think of as a "word" is, for search purposes, more correctly called a "token," a combination of characters that represents a single word. Recognizing these tokens for inclusion within a search index is called "tokenization." So, in order to properly tokenize a CJK document, the tokenization engine must be able to recognize where one word ends and the next begins. Not only can the tokenizer not rely on the use of spaces, but consider that there are two forms of written

Chinese, three different alphabets used in Japanese and pre- and post-fixes in Korean to indicate whether each word is a subject or object of the sentence. (Thai, Arabic, Russian and Hebrew are four more common languages that present similar challenges.)

Not confusing enough? Consider that the CJK languages frequently have several different characters that can represent the same concept, and the meaning of a single character can change dramatically depending upon its context in a sentence.

For litigation purposes, the burning question is: How are we supposed to run a search query on a set of foreign-language documents with all these characteristics? Fortunately, the answer does not require a team of expert linguists translating all these thousands of documents into English. It merely requires that you (or your service bureau) use tokenizing software customized for each language that appears in the documents, which can find, parse and index all of the "words" that appear.

To search these indexed documents, it is critical to use processing software that can recognize different character sets within the same document (and frequently, within the same sentence) and can utilize bi-directional processing so that left-to-right strings of text embedded in right-to-left sentences (or top-to-bottom text strings, as we frequently see in CJK) will be recognized correctly.

With these methods in place, someone who is conversant in the appropriate language can enter a search term just like we do in English, and search or cull a document universe down to an easily manageable amount of documents. Without these methods, an army of translators will run up huge discovery costs as they slog through these foreign-language documents and files one at a time (if, as mentioned before, they were processed properly to begin with).

Maybe I'll Just Stick to English

These issues are not ones that a legal practitioner should have to delve into too deeply; after all, that's why you hire a service bureau to begin with. Without question, though, they are matters that your service bureau must master so that, when you get your processed foreign-language documents back, the documents will truly and accurately represent the content of the documents and files they came from and will be ready for searching. Without a basic awareness of these issues, you may find yourself with a very unpleasant surprise when the document production comes back to your desk.

If you use an in-house support team, and you frequently deal with documents from a particular foreign country, it may well be worth the investment to purchase tokenization engines for the languages that are likely to come into the firm. For most practitioners, however, this is not a viable option. Most of us send the files out and expect the documents to be properly extracted, tokenized and returned to them in an easily-searchable form.

Since you hired a service bureau (presumably) to do the complicated document processing so you wouldn't have to — and since you hired them to do it the "right" way, so that you could focus on your litigation practice without having to sweat the production details — you cannot afford to simply presume that your service bureau is up to the challenge of properly processing documents with foreign-language character sets. While you don't have to understand the nuances of how these documents are processed, it will ultimately be you, and not the service bureau, who is responsible to your client and to the court to make sure that these documents are processed correctly.

About our author :: :: ::

Gary Wiener, Esq., is Director of Litigation Services for Liquid Litigation Management, Inc. He is an attorney and electronic discovery consultant who focuses on developing and implementing effective strategies for managing electronic discovery. After a successful career as a trial lawyer, Gary decided to combine his passion for computer technology with his litigation experience. He is a member of the State Bar of Texas and a former Director of the Travis County Bar Association. He can be reached at gwiener@llminc.com.

This article was first published in ILTA's August, 2008 issue of Peer to Peer *and is reprinted here with permission. For more information about ILTA, visit their website at www.iltanet.org.*