

# SING: The Final Frontier For Japanese DTP

BY JOEL BRECKINRIDGE

**Adobe has introduced a new architecture that uses metadata to allow a single glyph to act as a mini font. It is designed to address long-standing font problems in the Japanese market, but the architecture has implications for Asian and Western markets as well.**

**O**f all the problems facing Japanese print production, none is more basic and troublesome than *gaiji*. It is an issue that we have written about before ([www.seyboldreports.com/SRPS/subs/3006/html/japanese.html](http://www.seyboldreports.com/SRPS/subs/3006/html/japanese.html)), but the *gaiji* problem remains the reason the Japanese desktop publishing (DTP) adoption rate has been mired near the 50% mark since the mid-1990's.

*Gaiji* are characters that are outside the boundaries of many encoding standards. They are not typically used in everyday Japanese, but they are used in people and place names, literature, Buddhist texts, and academic and government documents. So what's the big deal?

When a print job needs a *gaiji* character not contained in a standard Japanese font, the desktop publishing process quickly bogs down. The required *gaiji* has to be added by purchasing special fonts that both the printer and designer absolutely have to have, or it has to be created by hand. This is more difficult, time consuming and expensive than you might imagine.

An experienced font designer can create a character from scratch in Illustrator that matches a regular font design in less than a working day. A less-experienced designer will probably take longer and the result

will be less pleasing. After the print job is finished, the Illustrator file is copied onto a disk and likely forgotten or lost until the next time that character is needed, and the show starts all over again. Every Japanese printer no matter how small has to deal with *gaiji*.

The first Japanese PostScript fonts that arrived in 1989 (known as OCF fonts) had a relatively limited character set of some 8,000 glyphs and primitive typography. High-end proprietary digital typesetting systems from Shaken, an industry standard, has much larger character sets and fast, excellent typography to meet the demands of high-end publishing. Many customers stayed with them; the book-publishing industry, for example, never bothered with clumsy DTP production.

Over the years, Adobe addressed the shortcoming of PostScript fonts, albeit very slowly. It helped that in the early 1990s, Apple made big noises about fixing all the problems with QuickDraw GX. A longtime Apple engineer once told me, "It's because of GX that we have OpenType." GX lit a fire under Adobe.

Expanding character sets and adding *gaiji* was the first priority. The Adobe-Japan (AJ) 1-3 character set that arrived in the mid-1990s bumped the collection to 9,354. The AJ 1-4 specification for Japanese OpenType has a total of 15,444 characters and AJ 1-5, released in 2002 in response to Apple's MacOS X extended character Hiragino Japanese fonts, has 20,317. This year we have yet another new specification, AJ 1-6, which adds even more *gaiji* used in newspaper production.

Specifications are important, and Adobe's Ken Lunde is doing everyone a great service by putting them together. However, they present font vendors and customers with a big problem: what to do about the installed base. Should vendors go to the expense and trouble of creating and marketing ever-larger font sets? Will customers bother to upgrade their old fonts?

The experience so far is not encouraging. In 1997, Japan's leading font vendor, Morisawa, released an important PostScript font upgrade called CID, which laid the foundation for embedding Japanese text in PDF documents. The upgrade package was ill con-

SING's Glyphlet management window. Here users add (turn on) or delete (turn off) glyphs. InDesign CS documents containing glyphlets can be opened and printed even if glyphlets are turned off.



ceived because Morisawa changed the font metrics, and older data did not display properly with the upgraded fonts. It was also poorly marketed and expensive. Customers stayed away in droves, and Morisawa was forced to release another upgrade one year later just to address the complaints and lost lots of face. Ever since, Japanese customers see font upgrades as a nuisance, not a necessity.

Japanese OpenType upgrades started arriving a year ago from Morisawa and Fontworks Japan with advanced layout features and the larger *gaiji* character sets of AJ 1-4, but customer reaction has been tepid. The majority of prepress files still use older OCF and CID fonts. Everybody knows that OpenType is the future, but in addition to 15 years of legacy data made with OCF fonts, the printing industry has been in a 10-year slump and is only beginning to show signs of recovery. Budgets are tight and companies would rather spend money upgrading to CTP systems that give the bottom line a bigger bang for the buck than font upgrades that still don't solve the *gaiji* problem.

With no apparent way out of the eternal *gaiji* problem, what's a printing company to do? One option might be to hope for the best and invest in OpenType Pro now. Morisawa will come out with yet another big upgrade containing the AJ 1-5 *gaiji* set next year. Yet font vendors don't want to go to the expense of making and releasing larger font sets when customers don't want to invest in them. What if a customer wants to use only a few characters from those super-size font sets? Couldn't they just order the ones they want as they could back in the old analog typesetter days?

The reality is that no digital font, even a super-size one, can cover all situations, and no font vendor would sell individual characters because there has been no architecture to support it. Until now.

Adobe has been quietly working to solve the *gaiji* problem. The first step was announced April 8 ([www.adobe.com/products/indesign/sing\\_gaiji.html](http://www.adobe.com/products/indesign/sing_gaiji.html)). The solution is called the SING *Gaiji* Architecture, a technology preview that is a plug-in for InDesign CS. The plug-in comes with 501 sample glyphlets based on Adobe's Kozuka Mincho font. Adobe said the architecture eventually will reach across all the applications of the Creative Suite. The SING architecture completely addresses *gaiji* management, from input and editing to typographic processing and output.

SING stands for Smart Independent Glyphlet. *Glyphlet* is a new word created by Adobe's font-engineering elves. It's helpful to explain the difference between a character and a glyph as defined in the Unicode world. In this model, a character is the fundamental concept, the letter L for example, and a glyph is the way the letter L is actually represented on the screen. Ligatures and other variations are glyphs that represent the character L. The most important point to remember is that a single character can be represented by many different glyphs.



Glyphlets are input with InDesign CS's character pallet. In the future, Japanese Mac users will be able to input glyphlets via the keyboard with a SING-compatible version of Ergo Soft's popular input module EG Bridge.

SING provides the roadmap for font vendors and other developers to break the *gaiji* deadlock with Adobe software. The basic concept is simple. Think of it as a return to the document-centered workflow of analog days: As long as all the necessary elements were on the page, it would print. This workflow model broke down with Japanese Postscript because of the limitations of OCF fonts. Even if all the elements are on the page, it would only print if the print company had the right fonts installed on its output devices. OpenType is slowly freeing Japanese DTP production from this for standard character sets, but SING will eventually liberate designers and printers once and for all.

When the SING plug-in is installed, the user simply loads the glyphlets with the SING management tool, then inputs the characters via the InDesign character pallet. Once the glyphlets are input into an InDesign CS file, they stay there even if the user turns off all the glyphlets in the SING management tool or moves the file to another computer that doesn't have the glyphlets installed. As long as the glyphlets are in the document, they display, print and output to PDF — wherever they go. The beauty of SING is its simplicity.

SING is preview software, so there are bound to be some limitations. It only works with InDesign CS, and if the glyphlets are not installed in the plug-in manager, they cannot be copied and pasted between different InDesign CS files.

PDF support is a little more complicated. SING works with PDF now, but SING glyphlets are stripped of their metadata, then converted to CFF or TrueType "dumb fonts" and embedded in the PDF document. "To the PDF file, it (SING glyphlets) looks like just a different font," said Jim DeLaHunt, Adobe's engineering manager of *gaiji* architectures. "It would be nice to have PDF treat glyphlets like full citizens." This suggests that the next major version of PDF will fully support SING metadata, but at this point DeLaHunt would only say, "It would be a nice thing to do and an obvious architectural direction."

How well SING will work with non-Adobe applications, such as Quark XPress, remains to be seen, but Japanese front-end processor and Macintosh developer ErgoSoft has pledged to add SING support to a future version of EG Bridge. (A front end processor, such as MacOS X's Kotoeri, translates alphabet keyboard input to Japanese characters.) At least some app developers will have reasonable access.

Technically, each SING glyphlet is an independent single file based on a subset of the OpenType format and contains vector data and metadata. The metadata contains the character stroke information, Unicode information, OpenType tables for text processing and layout, as well as a lot of other information.

"This (metadata) is what makes SING such a flexible architecture. The metadata allows the glyphlets to be independent," said Tomoo Yuri, Adobe Japan service provider manager for InDesign CS. Metadata also means better glyphlet management and faster searches. And metadata reduces the dependence on font encoding. Every single arcane character won't necessarily be part of Unicode, but it won't be necessary for smooth print workflow.

### Extended OpenType

DeLaHunt has been with Adobe from its very first foray into the Japanese market in 1989, when his job was to support Japanese developers. "One of the questions developers asked me was, 'How was PostScript going to handle the *gaiji* problem?'" It took Adobe 14 years to come up with the answer. DeLaHunt gave us a rundown on all the technical details.

"The glyphlet file format is based on OpenType," he said. "The outline data can be in CFF format or TrueType format. The metadata information is stored in tables that use OpenType conventions. The layout model that an application uses to do text processing using glyphlets is based on the OpenType layout

InDesign CS can output SING glyphlets to PDF.

model. OpenType is based on Unicode and SING is based on Unicode as well," he said.

"We are not attempting to create a new font format," added DeLaHunt. "We took the OpenType philosophy, applied it to the requirements of *gaiji* and came up with extended OpenType."

### What Took So Long?

"Adobe has been aware of *gaiji* market requirements," DeLaHunt said. "We first had to get a font format adequate for the Japanese market. (OpenType) allows high-quality fonts with the proper metrics to do high-quality Japanese layout. Secondly, we needed to have an application with Japanese typography designed in so that we could do really high-quality layout using OpenType, and that is InDesign," he said.

"Once those two pieces were out of the way, the next issue we came up against was *gaiji*," said DeLaHunt. "About four years ago, we started saying, 'We need to figure out how we can build a *gaiji* architecture that builds on everything we learned about Japanese typography in 15 years in the Japanese market.'"

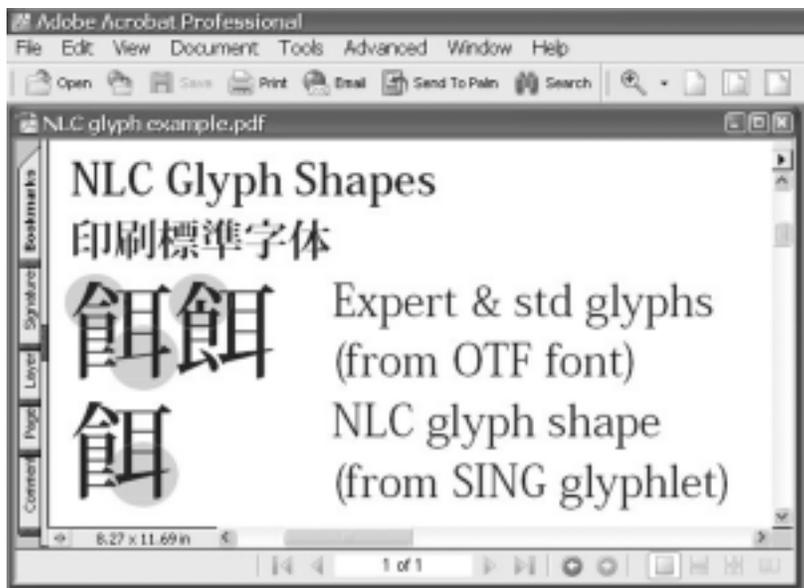
### Input from Morisawa

"I must give Morisawa some credit here that goes beyond joint development of fonts," said DeLaHunt. "They were our senpai (mentor), telling us about the market requirements and what their customers were telling them, and they took us to visit some of their customers. From that came the idea of each glyph bundled together with metadata and from that idea, we developed the architecture." (This is an interesting comment in light of the statement from a top Morisawa font engineer, who said: "Adobe kept asking us to build a *gaiji* solution, but we didn't want to. They eventually came to the conclusion they had to do it on their own.")

"If you step back and look at the architecture of an OpenType text layout system, it has character level data (which is a Unicode plain text string), and it has a bunch of formatting that is applied to the character level data (such as choice of fonts, and choice of glyphs via OpenType features, and ligatures, that sort of thing)," said DeLaHunt. "All of that formatting affects the mapping from character codes to glyph codes. Then you have text layout, which is based on font metrics. You can see how some character properties and glyph properties are pulled out of the font, but other data properties are pulled from elsewhere in the system," he said.

"Another example of the way the FEP [Front End Processor] is able to convert keystrokes to the Unicode character string is by using dictionaries of character code readings," he said. "Also, the way the layout system is able to decide where to break lines is by means of JIS 4051 (a Japanese typography standard)," said DeLaHunt

"The way glyph properties are stored, only some of them are in the font," he said. "Some of them are in



tables in the OS, the FEP or in the layout application. With SING, we tried to take all of those bits of data and put them in the glyphlet. And we also tried to make it so that the glyphlet would have enough data to describe itself in a lot of different situations,” he said.

“SING has a provision for glyphlet makers to put references to which page and entry in a Chinese or Japanese dictionary defines a glyph,” DeLaHunt said. “There’s a lot of data that we don’t see a way to use in InDesign now, but we figure we are building an architecture that lets glyphs be used in a lot of different environments. It’s important to give glyphlet makers a rich language for describing their glyph. There is nothing stopping you from making a glyph for a character that is not included in Unicode,” he said.

“One thing that InDesign does is match glyphs when changing a font,” DeLaHunt said. “Most apps cannot do this and cannot match-find the correct *gaiji* when the font is changed. With SING, if the glyph has a CID or Unicode ID, in most cases the layout engine can figure out the corresponding glyph when changing fonts late in the editing process. If we can make that happen it should be an interesting step forward for the Japanese design market. The current technology preview does not implement font changing this way. But there is no technical reason why it can’t happen.”

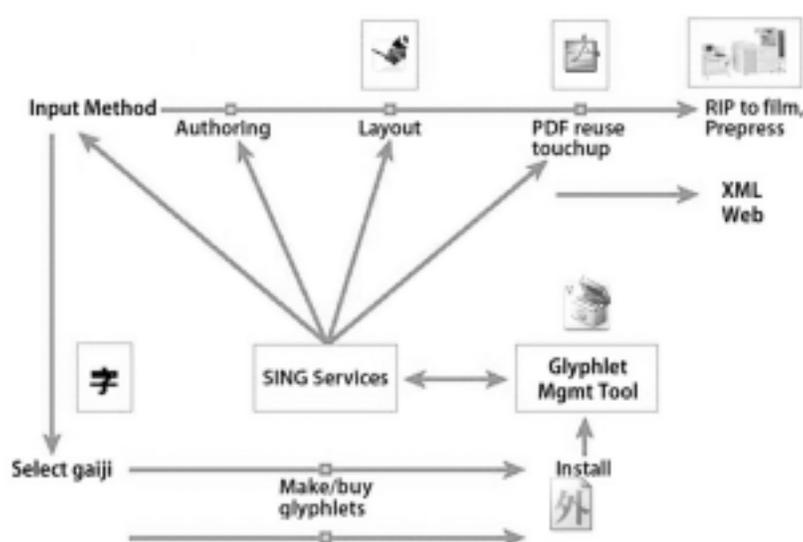
### A New Business Model for Fonts?

SING could help breathe new life into the Japanese font market and give vendors a chance to build a whole new business model. Instead of offering expensive super-size fonts, vendors could sell small glyphlet collections for online purchase and download so users can get just the *gaiji* they need in a hurry. It will be interesting to see what vendors come up with. Fontworks Japan likely will devise the most imaginative approach.

*Gaiji* creation tools should also get a lift. Until now, these tools have been hobbled with encoding problems and the same old “if the printer doesn’t have it I can use it” situation. SING-compatible *gaiji* tools will be free of encoding problems and should work seamlessly with Adobe products. Fontworks Japan has announced that its *Gaiji Master* ([www.seyboldreports.com/SRPS/subs/3006/html/stroke-fonts.html](http://www.seyboldreports.com/SRPS/subs/3006/html/stroke-fonts.html)) will support SING glyphlet creation in a future upgrade.

Adobe and DeLaHunt pointed out there are uses for *gaiji* in the West as well, such as with the euro symbol, for example. In the ’90s, every font and driver had to be updated to include the euro. “From the Japanese standpoint, that was nothing more than a *gaiji* being introduced. The 2004 version of this is the new currency symbol for Ukraine,” said DeLaHunt.

“There is also the possibility of adding alternate glyphs to Western fonts, such as alternate shapes of r’s



and p’s and so on. Imagine adding those using SING. In the West the concept of *gaiji* is much less common. Ultimately, what it all boils down to is, sometimes you need a new glyph, but you don’t need a new font. SING finally lets this become reality,” DeLaHunt said.

A simple overview of SING workflow.

### Big Changes for Japan and Beyond

SING, like OpenType or PostScript, is an Adobe technology that really puts a synergistic shine on Adobe applications. The advanced typography of OpenType, for example, is open for Window and Macintosh application developers to use, but nobody’s app uses them as well as an Adobe app does. SING provides a semi-open technological solution and boosts the bottom line.

It also will probably break the dominance of Quark XPress in the Japanese market and start a real switch over to InDesign, unless SING is made open enough for Quark to play and Quark moves to close the big Japanese feature gap with InDesign CS J.

It will take time for the impact of SING to reverberate throughout the Japanese market, but Adobe has done a great deal of thinking to solve the *gaiji* problem and has come up with an elegant solution that can push the market forward an important way.

The real test is, how open will the finalized architecture be? The more open and level Adobe makes SING, the more successful it will be. That’s in the industry’s best interest and, ultimately, Adobe’s.

### About the Author

Joel Breckinridge has lived in Japan for 20 years and has worked for several Japanese printing companies, including a stint as product manager for Heidelberg Japan. He works in Tokyo as a consultant and writer. After writing about Japanese font problems for eight years, he figures the release of SING will finally relieve him of that duty. He can be reached at [jbreckinridge@mac.com](mailto:jbreckinridge@mac.com).