

Legacy Gaiji Solutions & SING

Dr. Ken Lunde—lunde@adobe.com

Senior Computer Scientist, CJKV Type Development

Adobe Systems Incorporated

September 9, 2008—IUC32 @ San Jose, CA, USA, Earth

The Gaiji Problem

- **Defined as the absence of a desired glyph**
 - Fonts are static entities with limited glyph complements
 - Writing systems are open-ended and dynamic
- **New ideographs are being coined or discovered**
- **A myriad of variant forms**
 - Ideographic Variation Sequences (IVSes) can be used as a solution
- **Western languages use logos and new symbols**
 - New currency symbols represent a good example

What Is A Gaiji?

- Any glyph that is not accessible in a font
- Prototypical example: ideographs
- Generic symbols, logos, new currency symbols
- May or may not be in Unicode
 - May be a variant form of a Unicode character
- May be present in the font, but not accessible
 - Poor-quality input method
 - Unencoded
- The bottom line: *the desired glyph is not accessible*

Enter Legacy Gaiji Solutions...

- **Font-based, as inline graphics, or hand-written (*gasp!*)**
 - Font-based solutions are preferred because gaiji behave like characters
- **Legacy gaiji solutions can and do work *most of the time***
 - There is very strong emphasis on the phrase *most of the time*
 - The more closed a system or environment is, the better they work
 - However, today's systems and environments are becoming more open
- **Portability and document interchange**
 - These wreak havoc on legacy gaiji solutions

Legacy Gaiji Solutions—Encoding

- Strong motivation or tendency to encode gaiji
- Early solutions: *single-byte fonts with custom encoding*
- Shift-JIS user-defined region
 - 1,880 code points
- Unicode PUA (*Private Use Area*)
 - 6,400 code points in BMP
 - 131,068 code points in Planes 15 and 16

Legacy Gaiji Solutions–Code Point Poaching

- **Code point poaching is bad**
 - It is not illegal, but nothing good comes of it
- **Existing code points are used or repurposed**
- **The results can be unpredictable or unexpected**
 - Line breaking, word spacing, or other line-layout behaviors

“All Your PUA Are Belong To Us”

- **PUA code points have no intrinsic properties**
 - Not good, but more predictable than code point poaching
- **PUA code points do not exchange well**
 - Multiple entities may use the same PUA code points for different purposes
- **PUA code points are volatile**
 - Conflicting uses

Legacy Gaiji Solutions—Entering Gaiji

- **By code point, palette, or panel**
 - Legacy gaiji solutions carry no metadata, such as readings
- **By reading, for larger or standard gaiji collections**
 - Supplied in the form of supplemental input method dictionaries
- **Ultimately, an encoded value goes into the document**
 - And, that encoded value is either a poached code point or PUA code point
- **In summary...**
 - *Entering gaiji using legacy gaiji solutions is far from ideal*

Legacy Gaiji Solutions—Why Do They Fail?

- **Not portable**
 - PUA nor poached code points are portable
- **Document interchange is a hurdle or barrier**
 - All font components must be available, even for closed systems

Enter SING—Smart INdependent Glyphlets

- **Adobe carefully explored and studied legacy solutions**
- **SING provides portability and document interchange**
 - These issues plague legacy gaiji solutions
- **SING does not require that gaiji be encoded**
 - Instead, rich metadata in the 'META' table serves to identify the glyph
- **SING glyphlets are sticky to documents that use them**
- **The creation, distribution, and use of SING glyphlets**
 - Small font-like objects
 - Libraries and applications to manage and use them

SING—What Is A SING Glyphlet?

- **Small OpenType-like font object that lacks key tables**
 - No 'name' nor 'OS/2' tables
 - Does not appear in applications' font menus
- **Includes one meaningful glyph at GID+1**
 - Additional functional glyphs at GID+2 and greater
 - Vertical variants or other contextual variants
- **Small and lightweight—*travels easily and quickly***
- **Sticky to documents that use them—*portable***
- **Easily distinguished from standard fonts**
 - Legacy gaiji solutions use standard font formats

SING—How Does It Work?

- **Deployed in CCJK versions of InDesign CS2 and later**
- **The SING Library**
 - Organizes SING glyphlets
 - Makes SING glyphlets available to SING-*savvy* applications
- **The Tin Library**
 - Used by SING-*savvy* applications
 - Augment installed fonts with SING glyphlets that refer to them
- **SING glyphlets behave as standard glyphs in fonts**
 - Appear in Glyph Panel
 - Interact with other glyphs in the augmented font

SING—How Are SING Glyphlets Created?

- SING glyphlets are made by users and developers
- Many SING glyphlet development tools are available
 - *cvt2sing*—command-line tool included in GDK
 - *GCT (Glyphlet Creation Tool)*—Adobe Illustrator plugin
 - When purchased with a CCJK Creative Suite that includes Adobe InDesign
 - *SigMaker*—Version 3.0 or later by FontLab
 - *SINGEdit*—by Musashi Systems

SING—Better Than Legacy Gaiji Solutions?


- **Legacy solutions work better in closed environments**
- **SING was designed to work in open environments**
 - It actually doesn't matter how closed or open an environment is
- **SING does not require that gaiji be encoded**
 - In fact, PUA usage is not allowed
- **SING glyphlets are small, lightweight, self-contained**
- **SING glyphlets are sticky to documents that use them**
- **SING glyphlets are thus portable**
 - Document interchange is enabled

SING—What Else Is Needed?

- **Broader support in applications**
 - Adobe Systems and third-party applications
- **Support in OSes**
- **Performance enhancements**
 - Two orders of magnitude performance increase in InDesign CS4

For More Information...

- **White Paper**
 - Entitled *White Paper: Legacy Gaiji Solutions & SING*
- **Adobe Tech Note #5148**
 - See: http://www.adobe.com/devnet/font/pdfs/5148.SING_Tutorial.pdf
- **GDK (*Glyphlet Development Kit*)**
 - Includes the *cvt2sing* command-line tool
 - See: <http://www.adobe.com/devnet/opentype/gdk/topic.html>



Revolutionizing
how the world engages
with ideas and information