The Power of "Plain Text" & the Importance of Meaningful Content

Dr. Ken Lunde | Senior Computer Scientist | Adobe Systems Incorporated

IUC34, Santa Clara, CA, USA, Earth

What Gives “Plain Text” Its Power?

- “Plain text” represents raw text data
- “Plain text” can endure in more environments
- “Plain text” survives transcoding operations
  - Between UTFs: 100%
  - Between UTFs and legacy encodings: it depends on the legacy encoding
- “Plain text” persists throughout a document workflow
- “Plain text” can be edited, searched, copied, pasted, imported, and exported
- “Plain text” can be repurposed, mined, analyzed, and transliterated
- “Plain text” can be stylized, made rich, or marked-up
- “Plain text” serves as the foundation for “meaningful content”
- Unicode “plain text” is far more usable than that based on legacy encodings

Why Is "Meaningful Content" Important?

- Using characters correctly—as intended—results in meaningful content
  - Playing "by the book" is key
  - The book: *The Unicode Standard*
  - Data is inaccessible and unintelligible unless its content is meaningful
- Unicode characters have Properties
  - See the *Unicode Character Database* (UCD) for more details
- Presentation layer versus content layer—directly applicable to PDF
  - Some applications can preserve both layers—Adobe InDesign via PDFLib
  - Some cannot preserve the content layer—MS Word via PDF Maker
- Content is equally important as presentation
  - For some environments—mobile and cloud computing—content is more important
- "Meaningful content" is not possible without a "plain text" representation
- Beware of pitfalls!

“It Looks Like Double-Byte.” — Unidentified Apple Employee

- What is wrong with this statement?
- Content versus presentation
  - Which is more important?
  - Correct answer: *both!*

Pitfalls Serve As "Unicode Test Cases"

- An important part of software development is testing
  - The more thorough the testing, the more robust the software
- The best way to confirm "by the book" Unicode support is through testing
- Consider how to develop "Unicode Test Cases" based on the pitfalls that follow
  - Some pitfalls are more difficult to detect than others

Pitfall #1: Code-Point Poaching

- Wildlife poaching is illegal
- Code-point poaching is not illegal, but inappropriate and a bad practice
  - The act of assigning an inappropriate glyph to the code point of an existing character
  - Imagine copying "かなと漢字" then "jqnBW" gets pasted into another document
- Can easily result in inappropriate or incorrect Unicode character properties
- Consider U+005C (REVERSE SOLIDUS)
  - Is it a Backslash (\), Yen (¥; U+00A9), or Won (₩; U+20A9)?
    - It depends…
  - Residual effects or influence from legacy standards, such as JIS X 0201 and KS X 1003
- Code-point poaching was a necessary evil for legacy encodings
- Code-point poaching sacrifices long-term stability for short-term benefits
- Somewhat difficult to detect code-point poaching
- PUA code point usage is a lesser evil

Pitfall #2: PUA Code Point Usage

- Unicode includes 137,468 Private Use Area (PUA) code points
  - 6,400 in the BMP—U+E000 through U+F8FF
  - 65,534 in Plane 15—U+F0000 through U+FFFFD
  - 65,534 in Plane 16—U+100000 through 10FFFD
- No inherent or useful Unicode character properties
- There are absolutely no guarantees
- Must be treated as an absolute last-resort method of encoding glyphs
- Reliable only in closed environments
- Some BMP-only environments use PUA code points for non-BMP characters
  - U-PRESS is an example of such an implementation
- Very easy to detect PUA code point usage

Pitfall #3: Normalization

- Normalization itself is not inherently bad
  - Defines a common form for multiple representations of the same character/sequence
- Depending on distinctions that are erased by Normalization is bad practice
  - An excellent example: CJK Compatibility Ideographs
    - U+FA47 (漢) becomes U+6F22 (漢)
  - 57 of the 985 Jinmei-yō Kanji (人名用漢字) map to CJK Compatibility Ideographs
  - 75 kanji in JIS X 0213:2004 map to CJK Compatibility Ideographs
  - Do not forget about the twelve CJK Unified Ideographs among them!
    - They are not subject to Normalization
- Normalization can be applied at any time, by any client that acts on text data
  - Bottom line: *Do not depend of distinctions that are erased by Normalization*

Pitfall #4: Unassigned/Reserved/Noncharacter Code Point Usage

- No Unicode character properties
  - Other than being unassigned, reserved, or noncharacter code points
- Unassigned code points may become assigned in the future
  - Possible Unicode character property conflict
  - Guaranteed glyph/character mismatch
- Reserved and noncharacter code points should simply not be used

Pitfall #5: Characters That "Fall Between The Proverbial Cracks"

- The URO (Unified Repertoire & Ordering) has more than 20,902 characters
  - Unicode Version 4.1 appended 22 characters
  - Unicode Version 5.1 appended 8 more characters
  - Unicode Version 5.2 appended 8 more characters
  - *One more character was approved on 08/11/2010!*
- The twelve CJK Unified Ideographs among the CJK Compatibility Ideographs
  - U+FA0E, U+FA0F, U+FA11, U+FA13, U+FA14, U+FA1F, U+FA21, U+FA23, U+FA24, and U+FA27 through U+FA29.
  - Not subject to Normalization
- CJK Unified Ideograph "Extensions"
  - Extensions A, B, C, and D—Extension E in development—*more Extensions to follow*
- Stay up-to-date and familiar with Unicode

Pitfall #6: Fonts With Glyphs That Map From More Than One Code Point

- The 'cmap' tables of many fonts map multiple code points to the same glyph
  - It is appropriate for many cases, to ensure that the same glyph is used
    - Consider the "Kangxi Radicals" (U+2F00 through U+2FD5)
    - U+2F00 (一) and U+4E00 (一) map to the same Adobe-Japan1-6 glyph: CID+1200
- Some implementations have no method to preserve the original content
  - PDF uses "ToUnicode" mapping resources to specify a glyph's preferred code point
    - All U+2F00 → CID+1200 and U+4E00 → CID+1200 instances become U+4E00
  - When a "ToUnicode" mapping resource is not available, heuristics must be used
    - All U+2F00 and U+4E00 instances can become U+2F00
- Some implementation are able to preserve the original content
  - Adobe InDesign preserves U+2F00 and U+4E00 in the PDF content layer

Pitfall #7: Supporting Only BMP Code Points

- The BMP is merely one of the 17 planes of Unicode
  - Arguably, the most frequently-used characters are in the BMP
- The BMP is effectively full
  - Any new block must be encoded outside the BMP
- The first beyond-BMP code points were assigned in Unicode Version 3.1
  - This ignores noncharacter and PUA code points that were assigned in Version 2.0
- As of Unicode 6.0, there are more beyond-BMP characters than BMP ones
  - 54,496 BMP “graphical” characters
  - 54,746 beyond-BMP “graphical” characters
- Today, there is no excuse for BMP-only implementations

Solutions to Code-Point Poaching & PUA Code Point Usage

- Check whether the latest version of Unicode includes the characters
  - Some environments support only BMP code points
- Take the time and make the effort to propose new characters
  - This is done via the appropriate National Body
  - For those in the US, the first step is to submit a proposal to the UTC
- If your application supports only BMP code points, you have work to do
  - Unicode is much more than the BMP

Solutions to Normalization of CJK Compatibility Ideographs

- Many CJK Compatibility Ideographs are thought to preserve glyph distinctions
  - This is an incorrect and dangerous assumption
  - Normalization removes such distinctions
- Ideographic Variation Sequences (IVSes) represent a viable solution
- IVS = Base Character + Variation Selector
  - A Base Character + Variation Selector sequence maps to a glyph
- IVSes are registered via Ideographic Variation Database (IVD) collections
  - IVD collections are independent by default
    - It is possible to share IVSes across IVD collection through mutual agreement
  - The "Adobe-Japan1" IVD Collection was registered on 12/14/2007
    - <U+8FBB, U+E0100> → 辻 (CID+3056) versus <U+8FBB, U+E0101> → 辻 (CID+8267)
  - The "Hanyo-Denshi" IVD Collection is in the process of being registered

Mobile & Cloud Computing Considerations

- Mobile is all about "plain text"
  - The notion of platform is blurred due to the large number of platforms
  - Interaction with other platforms is guaranteed
- Successful mobile implementations require meaningful content
  - Unicode serves as the foundation for platform-independent text data
- Unicode Version 6.0 begins to address the "emoji" issue
  - All legacy implementations of emoji are PUA-based

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—

06—

07—

08—

09—

10—

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—

06—

07—

08—

09—

10—**GB 18030 certification without PUA requires six Extension B ideographs**

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—

06—

07—

08—

09—**Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B and C**

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—

06—

07—

08—**JIS X 0213:2004 includes 303 ideographs in Extension B**

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B and C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—

06—

07—**The VSes that are required for IVS support are in Plane 14**

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B and C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—

06—**Many important-for-mobile emoji characters are in Plane 1**

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B and C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—

05—**CJK Unified Ideographs Extensions B, C & D are beyond the BMP**

06—Many important-for-mobile emoji characters are in Plane 1

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B & C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—

04—**Unicode Version 5.1 broke the 100,000-character barrier**

05—CJK Unified Ideographs Extensions B, C & D are beyond the BMP

06—Many important-for-mobile emoji characters are in Plane 1

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B & C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—

03—**Mac OS X and Windows OS support beyond-BMP code points**

04—Unicode Version 5.1 broke the 100,000-character barrier

05—CJK Unified Ideographs Extensions B, C & D are beyond the BMP

06—Many important-for-mobile emoji characters are in Plane 1

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B & C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—

02—**As of Unicode Version 6.0, there are more characters beyond the BMP**

03—Mac OS X and Windows OS support beyond-BMP code points

04—Unicode Version 5.1 broke the 100,000-character barrier

05—CJK Unified Ideographs Extensions B, C & D are beyond the BMP

06—Many important-for-mobile emoji characters are in Plane 1

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B & C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—**So you do not cripple other products that depend on your own product**

02—As of Unicode Version 6.0, there are more characters beyond the BMP

03—Mac OS X and Windows OS support beyond-BMP code points

04—Unicode Version 5.1 broke the 100,000-character barrier

05—CJK Unified Ideographs Extensions B, C & D are beyond the BMP

06—Many important-for-mobile emoji characters are in Plane 1

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B & C

10—GB 18030 certification without PUA requires six Extension B ideographs

Top Ten List: Why Support Beyond-BMP Code Points?

01—So you do not cripple other products that depend on your own product

02—As of Unicode Version 6.0, there are more characters beyond the BMP

03—Mac OS X and Windows OS support beyond-BMP code points

04—Unicode Version 5.1 broke the 100,000-character barrier

05—CJK Unified Ideographs Extensions B, C & D are beyond the BMP

06—Many important-for-mobile emoji characters are in Plane 1

07—The VSes that are required for IVS support are in Plane 14

08—JIS X 0213:2004 includes 303 ideographs in Extension B

09—Hong Kong SCS-2008 includes 1,713 ideographs in Extensions B & C

10—GB 18030 certification without PUA requires six Extension B ideographs

CJK Unified Ideographs “Extension B” Character Usage Example

It is important that U+20BB7…

…does not become this:


Adobe