

Indic Writing Systems on the Java™ 2 Platform

Naoto Sato

Member of Technical Staff, Java Internationalization
Sun Microsystems, Inc., California, USA

Summary of Slides

Agenda

- Java 2 platforms
- Java 2 platform internationalization
- Text rendering for Indic scripts
- Text input for Indic scripts
- Character encodings for Indic scripts
- Building web applications using Indic scripts
- Conclusion

Java 2 Platform

- Three editions: Standard, Enterprise, Micro
- Focus here: Standard Edition, release 1.4
- Primary target: desktop clients
- Also foundation for Enterprise Edition
- APIs designed through Java Community Process
- Implementations by many vendors for many different host systems. Examples:
- Sun for Windows, Solaris, Linux
- Apple for Mac OS, IBM for OS/2, AS 400, Windows, Linux

Text Rendering for Indic Scripts – Issues

- Unicode and Java are character based
- Easy to process, hard to render
- Complex mapping from characters to glyphs
- Mapping needs to look at entire syllable
- Dependent vowels are combining marks
- Complex ligatures
- Glyph reordering
- Modifying characters in some scripts: virama, joiner, non-joiner, nukta, bindus, svaras. Most host systems don't support it

Text Rendering of Indic Scripts – Solutions

- Java 2D now renders Indic text
- Independent of host OS
- Based on OpenType specification
- Can handle nine Indic writing systems
- Needs font with OpenType tables
- Host OS fonts on Windows 2000
- Lucida font in J2RE
- Third party fonts -
- Swing user interface toolkit uses 2D

Text Input for Indic Scripts – Issues

- Many different keyboard layouts
- Some layouts use precomposed glyphs
- Most host systems don't support it

Text Input for Indic Scripts – Solutions

- Implement keyboard layouts as simple input methods
- Use Java input method framework
- Independent of host OS
- Map English to Indic keyboard layouts
- May use contextual and many-to-many character mappings
- Sun provides layouts for nine Indic scripts
- Swing user interface toolkit supports input methods

Character Encoding for Indic Scripts

- Issues
- Existing data comes in many different encodings
- Many encode glyphs instead of characters
- Solutions
- Text must be converted to/from Unicode
- New plug-in interface for character converters
- ISCII converter

Building Web Applications using Indic Scripts

- Send HTML
- Leaves text input and rendering to browser
- Works only if browser/host OS support Indic
- If not UTF-8, requires user to set encoding
- Send text as images

- Java 2D rendering can be used on server
- Works with any browser
- Slow, doesn't solve input problem

Product Information

- Java 2 Standard Edition 1.4 is in beta now
- <http://java.sun.com/j2se/1.4>
- <http://java.sun.com/j2se/1.4/docs/guide/intl>
- Product level implementation of Hindi
- Font, input method, locale data, ISCII converter
- Technology supports additional scripts
- Tamil, Bengali, Gujarati, Gurmukhi, Kannada, Malayalam, Oriya, Telugu
- Thanks to IBM

Conclusion

- Java 2 provides technology for Indic scripts
- Host OS independent implementation
- Allows you to develop web applications using your customers' languages