Worldwide Developers Conference

Bringing Unicode to the Mac OS: I

Peter Edberg

Senior Software Engineer
International, Text,
and Graphics

What Is Unicode?

- Uniform, universal 16-bit character set (*)
  - No byte values are special
  - Inline 32-bit characters: UTF-16 (*)
- Characters for most languages, many symbols
- Specifies additional information
  - Character properties
  - Rendering behavior
- Parallel standard ISO 10646
  - Same code points, no properties or behavior
- Originated by Apple and Xerox in 1988

Who Is Using Unicode?

- Java
- Document charset for latest HTML spec
- LDAP, other Internet services
- UDF (Universal Disk Format)
- Rhapsody Text System
- Newton
- Windows NT

Why Do We Need Unicode?

- 50+ encodings used on the Internet
- Too much work for every application and platform to handle them all
- Unicode includes the characters in these encodings, so
  - Deal with a single encoding
  - Use as a hub for conversion
- Easier to handle than many encodings
- Note: Unicode is not a complete international solution

Unicode Design Principles

- Separation between character and glyph
  - Assumes modern display system, complex text-to-display mapping
  - Different groupings (text elements) for different processes
- Text in logical order (as spoken); some exceptions
- Dynamic composition of diacritics
- Encodes plain text; does not encode language
- Character unification

ㅍ+ㅡ+ㄹ
=플

a+´=á
A+´=Á

Unicode Transformation Formats

- UTF-8: 8-bit safe (for Web, UNIX)
  - All of ASCII range maps to ASCII
    - One-byte nulls
  - Other 16-bit characters use 2–3 bytes
- UTF-7:7-bit safe (for mail)
  - ‘+’ to shift in, ‘-’ to shift out, modified base 64 in between
- See RFC 1641

Unicode vs. WorldScript

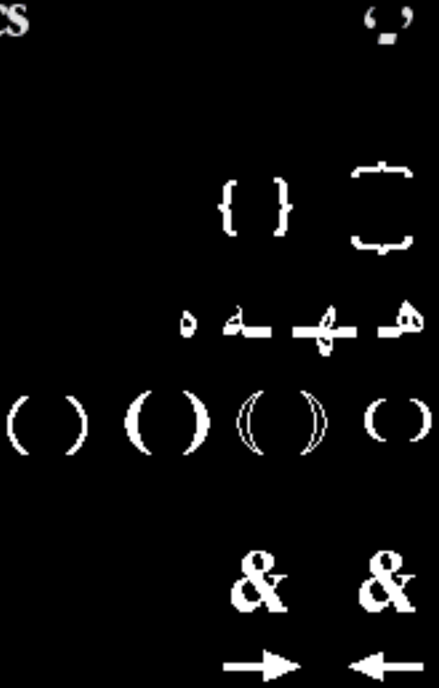
- Unicode:
  - Character encoding
- WorldScript:
  - Environment supporting multiple character encodings in the Mac OS
    - Certain assumptions about these encodings
  - Enhances QuickDraw to handle correct basic multilingual display
  - Provides text utilities
  - Provides locale information and related utilities

Character Sets and Encodings

- Coded character sets
  - Mapping from range of numbers to repertoire of characters
  - Fixed-width: 7-bit, 8-bit, 2-3 ¥ 7-bit
- Character encoding schemes
  - Include complex mappings: from sequence of bytes to sequence of characters
  - Multiple character sets (2-4) in single stream
  - Packing schemes (serial 8-bit): Shift-JIS, EUC
  - Switching schemes (serial 7-bit): ISO 2022…
- Internet “charset” designates character encoding

Character Set Features

- Multiple or ambiguous semantics
- Encoded presentation forms
  - Vertical forms
  - Contextual forms
  - Style variants
- Combining characters
- Direction clones
- Large repertoire for some
- User-defined characters, private or vendor additions



What Should an Encoding Converter Do? (Ours Does)

- All Mac OS encodings, top 50 Internet encodings
- Handle encoding features described above
- Round trip fidelity, especially
  Mac OS encoding ÆUnicode ÆMac OS encoding
- Minimize use of private Unicodes;
  maximize interoperability
- Auto-detection of encoding
- Map from Unicode to optimal series of runs in available target encodings
- Handle non-block-delimited conversion

These Requirements Imply…

- Map a source sequence of 1..m characters to 0..n characters in target
- Map many source sequences to one sequence in target
- Resolve character direction, use it in mappings
- Analyze contextual form, use it in mappings
- Support multiple tolerance levels

Text Encoding Converter Overview

- Extension containing three libraries (PPC and CFM-68K)
  - Text Common (general utilities)
  - Unicode Converter (low-level API)
  - Text Encoding Converter (high-level API)
- Text Encodings folder containing files with tables and/or plug-in code
- TEC 1.2 included with Tempo
- TEC 1.2 will also be available as SDK
- Mac OS clients: Cyberdog, MRJ, Data Detectors…

TextCommon

- TextEncoding type
- Functions to pack & unpack TextEncoding
- Functions to convert between old types and TextEncoding
- Get a localized user name for a TextEncoding

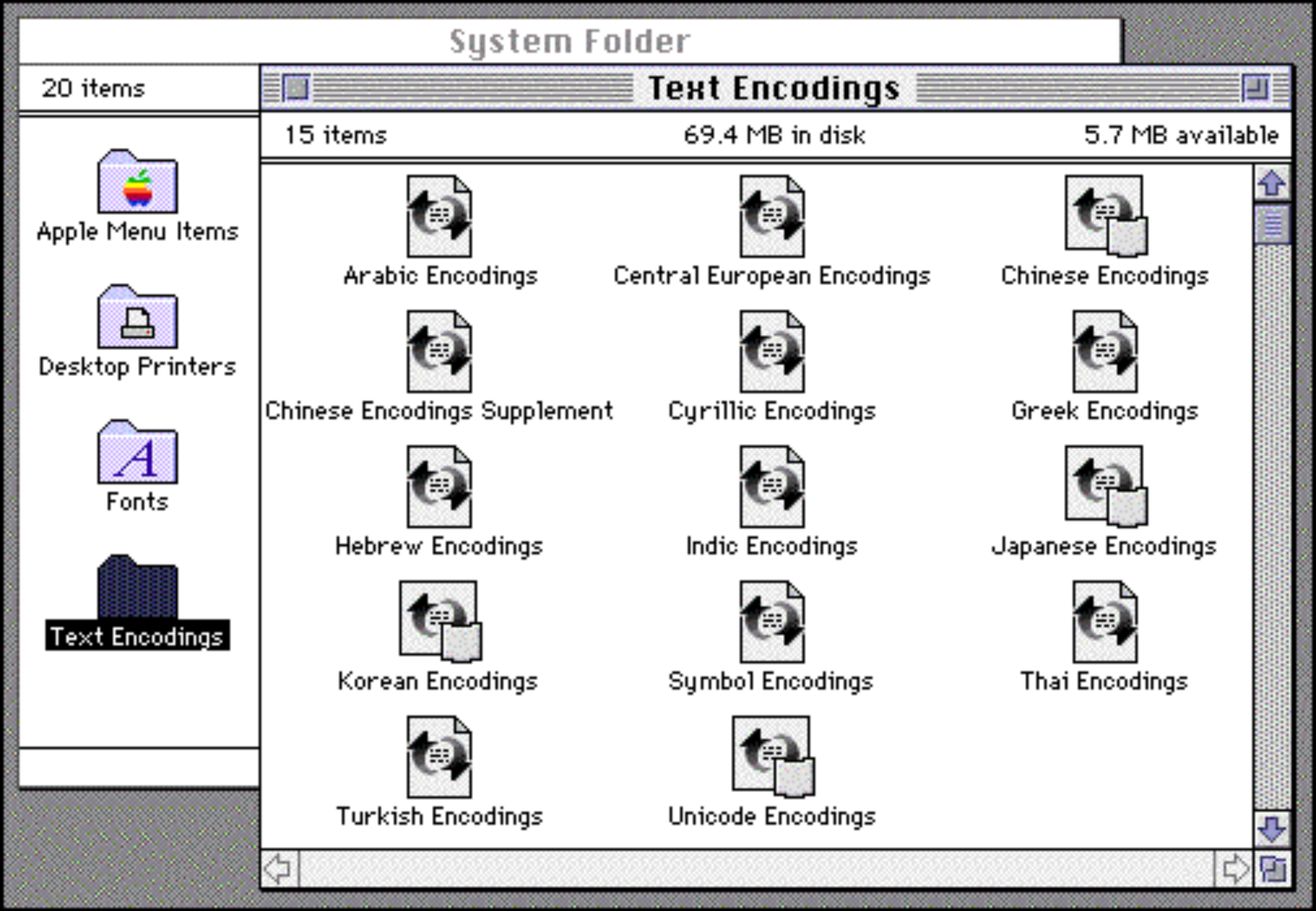
Low-Level API

- Table-based conversion to and from Unicode
- More control (so more setup required)
- Map style run offsets to target (for styled text)
- Optional caller fallback handling
- No code-switching schemes or algorithmic conversions
- Tables for all Mac OS encodings, common ISO encodings, many Windows encodings…
- Mapping strategy: Roundtrip fidelity with maximum interoperability

High-level API

- Intended for plain text or inline formatting (html)
- Simpler API, does more for client, less control
- Code conversion between arbitrary encodings
  - Table based and algorithmic conversion
    - e.g., JIS to Shift JIS, Shift JIS to EUC
  - Code switching schemes
- Supports code plug-ins
- Multiple plug-in conversion modules, chained as necessary
- Unicode converter is one plug-in

The Text Encodings Folder and Its Contents



System Folder

20 items

Apple Menu Items

Desktop Printers

Fonts

Text Encodings

Text Encodings

15 items 69.4 MB in disk 5.7 MB available

Arabic Encodings

Central European Encodings

Chinese Encodings

Chinese Encodings Supplement

Cyrillic Encodings

Greek Encodings

Hebrew Encodings

Indic Encodings

Japanese Encodings

Korean Encodings

Symbol Encodings

Thai Encodings

Turkish Encodings

Unicode Encodings

Bringing Unicode to the Mac OS: II

John H. Jenkins

International Software Engineer

Unicode on the Macintosh Now

- WorldScript and the Text Encoding Converter
  - Complete set of international text utilities
    - Number formatting, collation, tokenization, etc.
  - Can be used to draw and manipulate many parts of Unicode
- QuickDraw GX
  - Fully featured and powerful drawing engine
  - Can be used to draw but not manipulate Unicode

WorldScript and the TEC

Advantages

- Top-of-the-line international support
- Full-fledged implementation of international utilities
  - Bidirectional text
  - Sorting
  - Line-breaking
- Comes with Mac OS

Gidi said, “אם אין אני לי מי לי”.

WorldScript and the TEC

Disadvantages

- Drawing not as full-featured as GX
- Performance
  - Conversion and drawing, not just drawing
- Limited coverage of Unicode
  - Not a problem if current Macintosh scripts cover your needs (and they probably do)
- Unexpected conversion glitches
  - e.g., Cyrillic with a Japanese font

русский vs. р у с с к и й

QuickDraw GX

Advantages

- Sophisticated, script-and encoding-neutral text drawing
  - No better text drawing engine anywhere!
- Correct handling of complex scripts
  - Arabic, South Asian scripts, etc.
- Numerous advanced features
  - Swashes, ligatures, kerning, contextual forms—the list goes on and on!
  - Can handle UTF-16

हिन्दी

Questions?

QuickDraw GX

Disadvantages

- Using QuickDraw GX means reworking some of your drawing code
- Getting QuickDraw and QuickDraw GX to work together can be difficult
- No line-breaking, making it harder to do multi-line text
- Printing has been a problem with GX
  - Solved in Tempo
- Fonts must be revised to support Unicode
  - Tools available at Apple's Web site

Transitioning to the Future

- Apple
  - Extending the QuickDraw API to include Unicode drawing
  - Being worked on for Allegro time-frame
- Developers
  - Convert to use Unicode internally
  - Use TEC to convert keyboard input and interchange WorldScript and Unicode text
  - Render through WorldScript or GX
    - Use WorldScript for basic text drawing
    - Use GX for high-end typography

Useful URLs

- http://unicode.org
  - The Unicode Consortium’s home page
  - Includes links to order the Unicode book
- http://fonts.apple.com/Tools/tools.html
  - Get the latest versions of Apple’s tools to create Unicode fonts for GX and the Newton

For International Types at WWDC…

- 192, Int’l Technologies Feedback Forum
  - Wed., 3:10–4:10, Hall J4
- 209, Rhapsody Text System & Localization
  - Wed., 4:30–5:30, Hall A1
- Lunch with Apple’s International Engineers
  - Thurs., 12:30–1:30, Hall 2, Find balloons!

Worldwide Developers Conference