Confusion In Conversion

How Does Occasional Character Corruption Happen ?

Ken Glidden, Teruhiko Kurosaka, & Kazuo Hirai

25th Internationalization and Unicode Conference,
Washington, DC, March/April 2004

strategy h process h technology h results
www.basistech.com

BASIS TECHNOLOGY

Introductions

- Basis Technology – www.basistech.com

- Ken – manager of engineering services
- Kuro – principal software engineer
- Kazuo – principal software engineer
- Kazuo and Kuro: native Japanese speakers

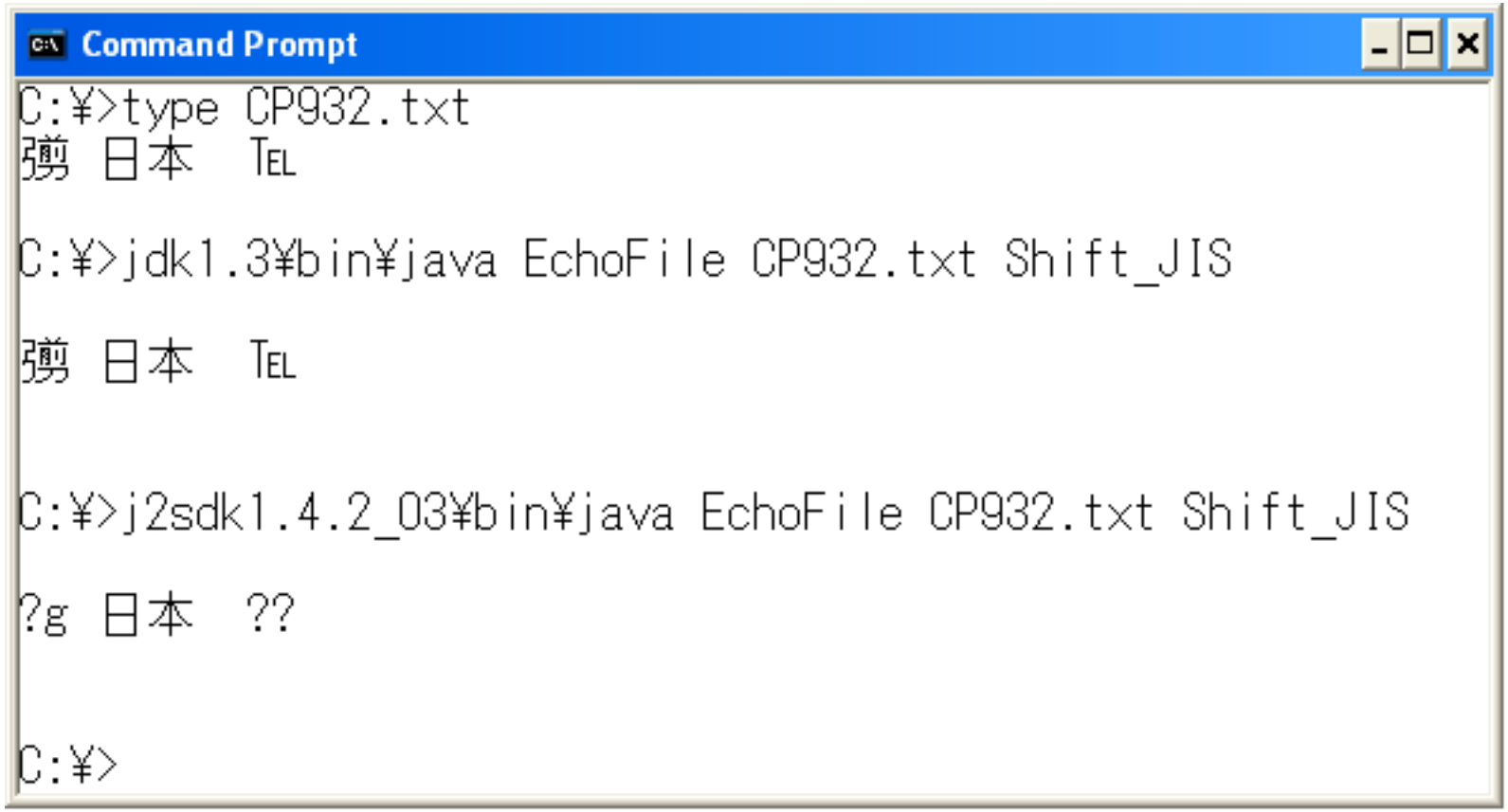
BASIS TECHNOLOGY

Outline

- Characterization
- Business impact
- Detection
- Prevention

BASIS TECHNOLOGY

Symptom of the Problem



BASIS TECHNOLOGY

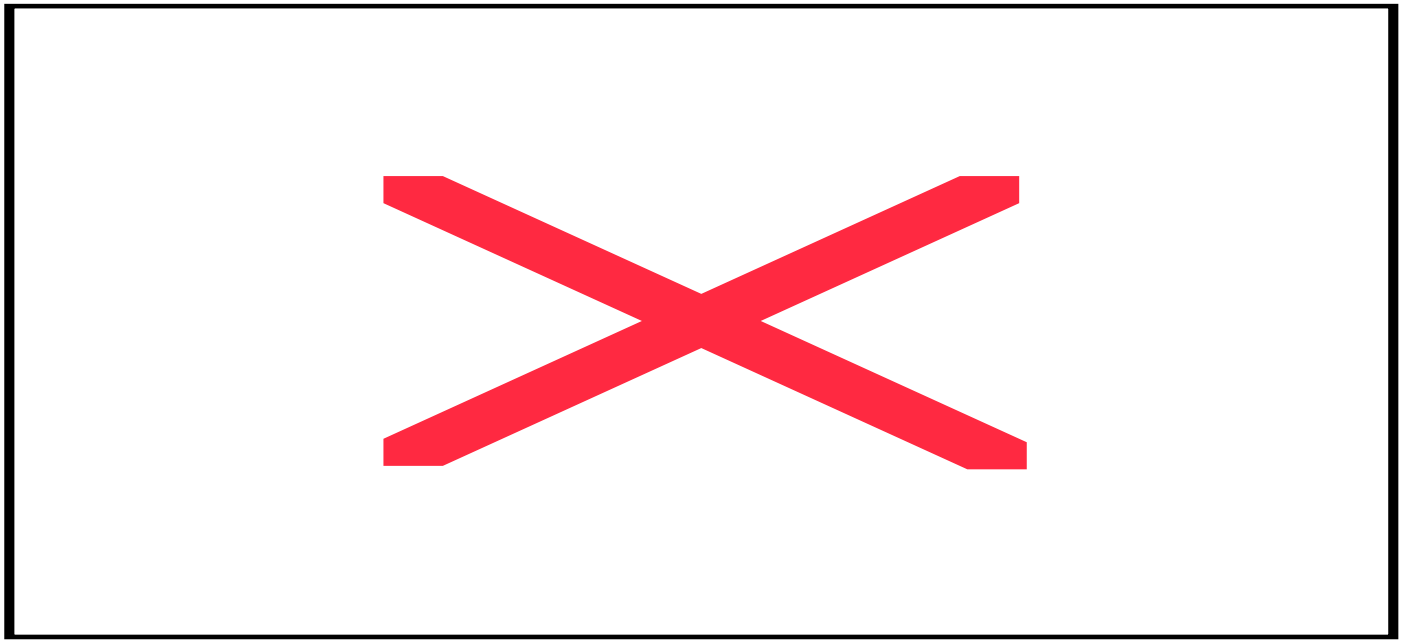
Problem Domain

- Some definitions
  - Confusion: 2 systems disagree
  - Character: implies readable text
  - Conversion: legacy - > Unicode -> legacy
  - Corruption: code points are wrong
  - Occasional: edge cases

- Domain Summary
  - Multi-tier systems
  - Transcoders disagree for identical encodings

BASIS TECHNOLOGY

Location of the Problem



BASIS TECHNOLOGY

Impact of the Problem

- Legacy encodings are still used
- System may need to input/output in a legacy encoding
- Examples:
  - Email-based help desk
  - Document storage/retrieval system
  - Network monitoring programs
  - Mixture of Java and C++ subsystems
- These issues are deal breakers

BASIS TECHNOLOGY

Cause of the Problem

- Generic issues
  - Conflicting encoding definitions
  - Conflicting maps
  - Vendor extensions
  - Undefined Code Points
  - Duplicates
- Japanese examples
  - 0x5C Yen or backslash
  - Some JIS characters mapped differently
  - IBM/NEC extensions
  - Half-width katakana and ISO-2022
  - CP932

BASIS TECHNOLOGY

Conflicting Encoding Definitions

- ASCII or JIS-Roman?
  - 0x5C = \ or ¥ (half-width yen)
  - 0x7e = ~ or _

Vendor Platform	Source Encoding	ASCII / JIS-Roman
RedHat Linux	EUC-JP	ASCII
RedHat Linux	SJIS	JIS-Roman
Solaris	EUC-JP	ASCII
Windows	CP932	ASCII
DEC OSF	EUC-JP	JIS-Roman

BASIS TECHNOLOGY

Conflicting Maps

- Differences in JIS X 0208-1990 to Unicode maps for 7 characters. (nn-nn) = JIS code point
  - ＼ (01-32)
  - ～ (01-33)
  - ＿ (01-34)
  - ― (01-61)
  - ∥ (01-81)
  - － (01-82)
  - ￠ (02-44)
- e.g. 01-33
  - U+301C, ～, RedHat Linux
  - U+FF5E, ～, Microsoft
- Chinese problems, too.

BASIS TECHNOLOGY

Vendor Extensions

- IBM/NEC
- Microsoft's CP932 includes these
- Others do not
- Potential for OCC

BASIS TECHNOLOGY

Undefined Code Points

- Some transcoders handle this differently
- Example: Halfwidth Katakana (ｱ vs.ｱ)
  - Not defined in ISO-2022-JP
  - But, some transcoders convert halfwidth to fullwidth

BASIS TECHNOLOGY

Duplicates

- Microsoft's CP932 has duplicate and triplicate characters in the IBM/NEC extension range
- The CP932 code map does not allow these to roundtrip.
- Visually this is not a problem
- Bit-wise it is.

BASIS TECHNOLOGY

Detecting OCC

- White-box testing (QA)
- Black-box testing (development)

BASIS TECHNOLOGY

White-box Testing

- Casual tests are insufficient
- Carefully planned tests are needed:
  - Loopback tests
  - Use all characters
  - Ensure all transcoders are exercised
- Test all subsystems

BASIS TECHNOLOGY

White-box Testing Caveats

- User-Defined Characters.
- Duplicated code points in MS 932 extension blocks typically do not round trip; need special treatment in judging test result.

BASIS TECHNOLOGY

Black-box Testing

- Test each transcoding module
- Characterize each transcoder
  - Test every character
  - Roundtrip every character

BASIS TECHNOLOGY

Preventing OCC

- Use the same transcoder everywhere
  - ICU
  - Basis Tech's RCLU
  - Gnu's libiconv
- Use transcoder's exception handling features.
- Use roundtrip-enabled transcoders

BASIS TECHNOLOGY

Conclusion

- OCC is a real issue for Unicode-enabled systems
- Recommendations for application developers
  - Include OCC tests during QA
  - Use one cross-platform library everywhere
  - Characterize the transcoder and handle its anomalies
- Recommendations for vendors
  - Standardize maps (e.g. iconv)

BASIS TECHNOLOGY

Thanks for coming!

- Thanks for the opportunity to share.