

Lecture 6 (1/2 hour)

Internationalization

Sang Shin

Java™ Technology Evangelist

sang.shin@sun.com

(You can use this material in any way you want,
but if you can drop me an email when you do,
that will be greatly appreciated.)

Topics



- Character set
- Encoding
- Unicode as a character set
- Unicode encodings
- ISO character sets
- Parser behavior

Terminology

- **Character set** maps characters to numbers
 - ◆ character “Z” is number 90, x5A
 - ◆ These numbers are called “**code points**”
- Character **encoding** determines how code points are represented in bytes
 - ◆ Code point 90 can be represented as
 - a signed byte
 - a little-endian unsigned short
 - 4 byte

Terminology

- A character set can have multiple encodings
- Unicode is a character set
- ISO has defined 14 “Latin” character sets (before Unicode gets Worldwide acceptance)
- Unicode has several encodings
- Most of other character sets have a single encoding scheme



Encoding Declaration in XML



- Every XML document should have an **encoding declaration**

```
<?xml version="1.0" encoding="US-ASCII"?>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

- Tells the parser what **character set** the document is written and **encoding**
 - ◆ **Parser** translates characters from document's native encoding into **Unicode**
 - ◆ Not all parsers knows all encodings but most parsers can handle most of the character sets you will encounter



Encoding Declaration

- Encoding declaration can be omitted if and only if the document is written in either **UTF-8** or **UTF-16** encodings of Unicode
- UTF-8 is a strict superset of ASCII
 - ◆ ASCII files are legal XML documents without encoding declaration

XML-Defined Character Set

- An XML parser **must handle** UTF-8 and UTF-16 encodings of Unicode
- Many XML parsers understand other legacy encodings
 - ◆ ISO-10646-UCS-2, ISO-10646-UCS-4, ISO-8859-1, ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7, ISO-8859-8, ISO-8859-9, ISO-2022-JP, Shift_JIS, EUC-JP

Unicode

- An international standard character set
- Is a character set large enough to include all living languages
- Unicode text editors are relatively uncommon yet
 - ◆ XML documents are typically written in other character sets

Unicode

- Current version 3.0.1 contains 49,194 characters
- Can represent all spoken languages
 - ◆ Latin alphabet, Ancient and modern Greek, Cyrillic, Chinese Han characters, Korean Hangeul, Japanese Katakana and Hiragana, Arabic, Hebrew, Indian Devanagari, Thai, Bengali, Tibetan

Unicode Encodings

- Unicode can be written in a variety of encodings
 - ◆ UCS-2
 - ◆ UCS-4
 - ◆ UTF-8
 - ◆ UTF-16

UCS-2

- Most natural encoding of Unicode
- Represents each character as a **two-byte unsigned integer**
 - ◆ 0 to 65,535
 - ◆ “A” is x0041
- Two variations
 - ◆ big-endian and little-endian
 - ◆ Distinguished by **“byte order mark”**
 - ◆ xFEFF or xFFFE

UCS-2

- Disadvantages
 - ◆ Files that contain Latin text can be twice the size
 - ◆ Is not backward for forward compatible with ASCII
 - Tools or programs written with ASCII in mind cannot handle UCS-2 document
 - ◆ Is limited to 65,536 characters

UCS-4

- Represents each character as a **four-byte unsigned integer**
- Practically no use at this point

UTF-8

- **Variable-length** encoding of Unicode
- ASCII character set (0 to 127)
 - ◆ One byte - same as in ASCII
 - “A” is x41 in ASCII
 - “A” is x41 in UTF-8
 - ◆ **Pure ASCII files are UTF-8 files**

UTF-8

- Most non-ideographic character set (128 to 1023)
 - ◆ Two bytes
- Chinese, Japanese, Korean characters (1024 and above)
 - ◆ Three bytes
- In the future, new characters can be added beyond 65,535
 - ◆ Four bytes

UTF-8

- ASCII files are the half the size of UCS-2 based files
- Files in Chinese, Japanese, Korean are 50% bigger
- Most broadly supported Unicode encoding
 - ◆ Java .class file store strings in UTF-8
- Default encoding of an XML parser

UTF-16

- Two bytes representation plus surrogate?
- Ordinary encoding scheme of Unicode
- Could be big-endian and little-endian
 - ◆ Have to have “byte order mark”
- Preferred for Chinese, Japanese, and Korean characters
- Internal representation?

ISO Character Sets

- Single byte character sets
- 14 Character sets
- Characters 0 to 127 are identical to ASCII character set
- Characters 128 to 255 vary
 - ◆ Cyrillic, Turkish
- **ISO-8859-1** (Latin-1)
 - ◆ ASCII plus most Latin-alphabet Western European languages

Which One to Use?

- UTF-8 or UTF-16 for all XML documents
- Increasing number of tools and programs supports UTF-8 or UTF-16
 - ◆ Word 2000 is Unicode editor
- Good XML and HTML editors will let you choose the character set

XML Parser

- First two bytes are xFEFF: big-endian, UCS-2 encoding of Unicode
- First two bytes are xFFFE: little-endian, UCS-2 encoding of Unicode
- First four bytes are x3C3F786D: ASCII “<?xm”, UTF-8 encoding of Unicode
 - ◆ Even if wrong assumption, it should be sufficient to read encoding declaration

Character References

- Predefined XML entity references
 - ◆ `<`, `>`, `&`, `"`, `'`
- **Number representation** of particular Unicode characters
 - ◆ `њ` (decimal representation)
 - ◆ `њ` (hexadecimal representation)
- Useful when only ASCII text editor is available
- To parser, they are the same

xml:lang

- XML document is a **multi-lingual** document
 - ◆ Arabic commentary on a Green text
- xml:lang identifies which language a particular section of text is written with two-letter language code

```
<maxim xml:lang="el">
```

```
&#x3C3;&#x3CC;
```

```
</maxim>
```

Summary

- XML is an excellent choice for multi-lingual documents
- Unicode is the choice of the character set of XML documents
- UTF-8 and UTF-16 are choice of encoding of Unicode

References

- “XML in a Nutshell” written by Elliotte Rusty Harold & W. Scott Means, O’Reilly, Jan. 2001(1st Edition), Chapter 5 “Internationalization”

