Encodings and character sets in XML

Author: Mihai Rotaru

Date: 1 Dec 2011

The character set used throughout an XML document can be declared as part of the first line of the document, inside the xml declaration:

```
1 <?xml version="1.0"? encoding="UTF-8"?>
```

If no encoding is declared, parsers will generally try to guess which encoding is used, first by looking at the first bytes of the file. If characters outside the assumed encoding are detected, the parser will try to find an alternative encoding that will contain it. If the characters are not recognized, the parser should stop processing (http://www.w3.org/TR/REC-xml/#charencoding).

XML encodings can be divided into three categories, the most common encodings being Unicode/ISO/IEC10646 encodings and transformations: "UTF-8", "UTF-16", "ISO-10646-UCS-2", and "ISO-10646-UCS-4". UTF-8 is the most used encoding (http://w3techs.com/technologies/overview/character_encoding/all), accounting for 67% of websites which use a known character encoding.

The main advantage of UTF-8 is that it can encode any Unicode character, and any valid ASCII text file is also a valid UTF-8 document. This is due to the fact that UTF-8 characters do not have a fixed width (in bytes), but can occupy from one to six bytes; and since it was designed with backwards-compatibility with ASCII, the first 127 characters are the same for ASCII and UTF-8.

Among disadvantages, the major drawback of UTF-8 is that for certain languages, it will take up more space. Taking Romanian as an example - the word "ampul#" cannot be represented with ASCII because of the "#" character (http://www.fileformat.info/info/unicode/char/103/index.htm); however, the ISO/IEC 8859-16, informally known as "Latin-10" or "South-Eastern European" encoding defines the character I need as 0xE3, so I would use this encoding for my document. ISO-8859-16 is a single-byte encoding, so it is very efficient and suited for situations when I don't need other non-ASCII characters.

However, to represent "#" using UTF-8, two bytes are needed: 0xC4 and 0x83, because it is decomposed into LATIN SMALL LETTER A (0xC4) and COMBINING BREVE (0x83), hence occupying double the amount of space. Therefore, if space were a concern, I would use the ISO-8859-16 encoding for Romanian texts.

However, it is unpractical to represent Chinese writing with any encodings which don't support many characters. Single-byte encodings work for Romanian and other languages with a small number of characters, but not for languages which use ideograms, such as Chinese or Japanese. In this situation, UTF-8 becomes the better option.