Request for Exemption from Certain ISO Style Rules

Rex Jaeschke, ISO/IEC 29500 Project Editor

(rex@RexJaeschke.com)

2012-04-24

Contents

1. Introduction

The standard ISO/IEC 29500 defines "Document description and processing languages — Office Open XML File Formats". It consists of approximately 6,500 pages contained in four Parts. I am the editor of that standard.

The initial edition of ISO/IEC 29500 resulted from a Fast Track by Ecma International. It did not—and did not need to—comply with the ISO/IEC Directives, Part 2, "Rules for the structure and drafting of International Standards". Subsequently, two further editions have been produced by ISO/IEC JTC 1/SC 34, as "consolidated reprints". That is, each was the result of a Technical Corrigendum and an Amendment being applied to the previous edition.

ISO/IEC JTC 1/SC 34 has now started work on a revision of one of the four Parts of ISO/IEC 29500, with CD Registration planned for early 2013. (The revision of another Part is expected to begin soon after.) As project editor, I am required to make revised editions comply with ISO/IEC Directives, Part 2. However, after having reviewed those requirements I've identified a few that will involve a lot of effort to reach compliance, but with no obvious improved readability or usability. This paper outlines those requirements for which I'd like ITTF to consider granting ISO/IEC 29500 an exemption.

2. Clause and Subclause Headings

Although I can find no written rule regarding the casing of clause and subclause titles, I deduce from the Directives that the intent is all words in such titles be spelled in lowercase except for the first letter of the first word and any uppercase letters required by the normal written form in running text (such as for proper nouns and acronyms).

ISO/IEC 29500 follows common USA/English publishing conventions in which almost all words in a title have a leading uppercase letter except for things like articles, conjunctions, and prepositions. Here is a representative set of titles taken from ISO/IEC 29500:

1. 10.1.3 Mapping Part Names to Physical Package Item Names
2. 10.2.3 Mapping Part Names to ZIP Item Names
3. 17.3.1.16 kinsoku (Use East Asian Typography Rules for First and Last Character per Line)
4. 17.3.1.10 divId (Associated HTML div ID)
5. 17.3.3.13 lastRenderedPageBreak (Position of Last Calculated Page Break)
6. 17.15.1.37 doNotHyphenateCaps (Do Not Hyphenate Words in ALL CAPITAL LETTERS)
7. 17.16.5.25 HYPERLINK
8. 17.18.55 ST_MailMergeOdsoFMDFieldType (Merge Field Mapping Types)
9. A.2 Creating a URI from an IRI

I estimate that in the 6,500 pages, the number of subclauses is (possibly many) more than 1,000. Most of them introduce XML vocabulary items, in an abbreviated short-form, followed by a long form.

Reviewing all these titles to see which could be "converted" would be very time consuming, and as the reader base for ISO/IEC 29500 (and its Ecma equivalent) well knows and understands them in their current form, I see no value in changing them.

3. Notes

As I mentioned above, the first edition of ISO/IEC 29500 was Fast Tracked from Ecma. I started editing programming language specifications for Ecma in 1999, and I borrowed the note style used in ISO/IEC 14882:1998, "Programming languages — C++". §1.5, "Structure of this International Standard", of that standard states, 'Each note is introduced by "[*Note*:" and terminated by "]". A representative example of such as note in that standard is, as follows:

> An object is a region of storage. [*Note*: A function is not an object, regardless of whether or not it occupies storage in the way that objects do.] An object is created by …

In this model, notes can be part of the paragraphs to which they pertain, and can even occur between normative sentences, two capabilities I found to be very attractive as well as space saving. However, I thought that having the note terminator simply be "]" made it hard to find visually, and no easier when searching electronically, especially in standards for programming languages and tools that support the "]" character. As such, I added the sentinel text "*end note*" as well (shown below in grey for expository purposes only), and removed the space before the "]" character. Using this approach the note example above becomes

> An object is a region of storage. [*Note*: A function is not an object, regardless of whether or not it occupies storage in the way that objects do. *end note*] An object is created by …

And that is what is used throughout ISO/IEC 29500. I also used this style for ECMA-334, "The C# programming language standard" (Fast Tracked as ISO/IEC 23270), ECMA-335, "Common Language Infrastructure (CLI) standard" (Fast Tracked as ISO/IEC 23271), ECMA-388, "The C++/CLI Language Specification", and ECMA-388, "Open XML Paper Specification [OpenXPS]", all of which are long and complex standards.

In ISO/IEC 29500, many subclauses are very long (e.g., 10–20 pages) as they contain tables of XML elements and attributes. In many cases rows within those tables have notes, which are often written at the end of, or in the middle of existing paragraphs.

In all my uses of notes, I've never needed to number or label them, as I have never needed to reference them by number or label; they simply are right at the point of the text to which they pertain.

Over the years, I have found two subcategories of notes to be useful, so much so that I've given them their own format, as follows:

1. [*Rationale:* … narrative … *end rationale*]
2. [*Guidance*: … narrative … *end guidance*]

and used them throughout the specifications I've edited.

Changing all existing notes to the ISO style will involve a lot of work for no obvious value. And it will almost certainly increase the page count as each note will have to be in its own paragraph and have indented margins. I also see the reduction of point size of note text to be problematic for those of us older folks with less-than-perfect eyesight.

4. Examples

The situation with examples is exactly like that for notes above, except that the notation used is, as follows:

[*Example*: … possible narrative

code fragment

possible narrative … *end example*]

Unlike notes, however, examples almost always involve (sometimes quite lengthy) XML or programming language fragments.