DR 09-0037 — PML, Presentation: Clarify description of properties in embeddedFontLst

Status: Closed; in AMD3

Subject: PML, Presentation: Clarify description of properties in embeddedFontLst

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00010

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §19.2.1.10, "font (Embedded Font Name)", p. 2766

Related DR(s): none

Nature of the Defect:

The entry of the referred to font in the embeddedFontLst, "font (Embedded Font Name)" (19.2.1.13) includes the properties like charset, panose, and pitchFamily, but they are already specified in the referencing part by the feature of DrawingML, or it should be described how the inconsistency should be handled when they are different between the referencing part and the referred to part. In the case of WordprocessingML, this information is not specified in the referencing part "rFont"; they are collected in "Font Table".

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

No

Editor's Response:

None

2009-06-17 Shawn Villaron:

We have two choices in terms of how we handle this defect report:

1. We can specify that the charset, panose and pitchFamily attributes are to be ignored if otherwise specified; or,
2. We can remove them from the standard altogether.

I'm looking for guidance here. I'm normally not inclined to remove things from the standard as I always worry that some implementer that I'm not aware of is writing this information. That said, based on my limited research, I do not know of anyone writing out these attributes.

Here is the markup for choice 1:

19.2.1.13 font (Embedded Font Name)

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td>charset (Similar Character Set)

Namespace: .../drawingml/2006/main</td><td>…

<table><tr><th>Value</th><th>Description</th></tr><tr><td>0x00</td><td>Specifies the ANSI character set. (IANA name iso-8859-1)</td></tr><tr><td>…</td><td>…</td></tr><tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr></table>
If a charset is specified elsewhere, this attribute should be ignored.

…</td></tr>
<tr><td>panose (Panose Setting)</td><td>Specifies the Panose-1 classification number for the current font using the mechanism defined in §4.2.7.17 of ISO/IEC 14496-22:2007.

If a charset is specified elsewhere, this attribute should be ignored.</td></tr>
</table>

Attributes	Description
Namespace: .../drawingml/2006/main	The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).
pitchFamily (Similar Font Family) Namespace: .../drawingml/2006/main	... This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. If a charset is specified elsewhere, this attribute should be ignored. The possible values for this attribute are defined by the W3C XML Schema byte datatype.

[*Note*: The W3C XML Schema definition of this element's content model (CT_TextFont) is located in §A.4.1. *end note*]

And here is the markup for choice 2:

19.2.1.13 font (Embedded Font Name)

Attributes	Description
~~charset (Similar Character Set)~~ ~~Namespace: .../drawingml/2006/main~~	~~Specifies the character set which is supported by the parent font. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.~~ ~~The value of this attribute shall be interpreted as follows:~~ ~~Value~~ / ~~Description~~

Attributes	Description	
	0x00	Specifies the ANSI character set. (IANA name iso-8859-1)
	0x01	Specifies the default character set.
	0x02	Specifies the Symbol character set. This value specifies that the characters in the Unicode private use area (U+FF00 to U+FFFF) of the font should be used to display characters in the range U+0000 to U+00FF.
	0x4D	Specifies a Macintosh (Standard Roman) character set. (IANA name macintosh)
	0x80	Specifies the JIS character set. (IANA name shift_jis)
	0x81	Specifies the Hangul character set. (IANA name ks_c_5601-1987)
	0x82	Specifies a Johab character set. (IANA name KS_C-5601-1992)
	0x86	Specifies the GB-2312 character set. (IANA name GBK)
	0x88	Specifies the Chinese Big Five character set. (IANA name Big5)
	0xA1	Specifies a Greek character set. (IANA name windows-1253)
	0xA2	Specifies a Turkish character set. (IANA name iso-8859-9)
	0xA3	Specifies a Vietnamese character set. (IANA name windows-1258)
	0xB1	Specifies a Hebrew character set. (IANA name windows-1255)

<table>
<tr><th>Attributes</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>0xB2</td><td>Specifies an Arabic character set. (IANA name windows-1256)</td></tr>
<tr><td>0xBA</td><td>Specifies a Baltic character set. (IANA name windows-1257)</td></tr>
<tr><td>0xCC</td><td>Specifies a Russian character set. (IANA name windows-1251)</td></tr>
<tr><td>0xDE</td><td>Specifies a Thai character set. (IANA name windows-874)</td></tr>
<tr><td>0xEE</td><td>Specifies an Eastern European character set. (IANA name windows-1250)</td></tr>
<tr><td>0xFF</td><td>Specifies an OEM character set not defined by ISO/IEC 29500.</td></tr>
<tr><td>Any other value</td><td>Application-defined, can be ignored.</td></tr>
</table>

The possible values for this attribute are defined by the W3C XML Schema byte datatype.
</td></tr>
<tr><td>panose (Panose Setting)

Namespace: …/drawingml/2006/main</td><td>Specifies the Panose-1 classification number for the current font using the mechanism defined in §4.2.7.17 of ISO/IEC 14496-22:2007.

The possible values for this attribute are defined by the ST_Panose simple type (§22.9.2.8).</td></tr>
<tr><td>pitchFamily (Similar Font Family)

Namespace: …/drawingml/2006/main</td><td>Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by a byte variable this value shall be interpreted as follows:</td></tr>
</table>

Attributes	Description
	~~Value~~ / ~~Description~~
	~~0x00~~ ~~DEFAULT PITCH + UNKNOWN FONT FAMILY~~
	~~0x01~~ ~~FIXED PITCH + UNKNOWN FONT FAMILY~~
	~~0x02~~ ~~VARIABLE PITCH + UNKNOWN FONT FAMILY~~
	~~0x10~~ ~~DEFAULT PITCH + ROMAN FONT FAMILY~~
	~~0x11~~ ~~FIXED PITCH + ROMAN FONT FAMILY~~
	~~0x12~~ ~~VARIABLE PITCH + ROMAN FONT FAMILY~~
	~~0x20~~ ~~DEFAULT PITCH + SWISS FONT FAMILY~~
	~~0x21~~ ~~FIXED PITCH + SWISS FONT FAMILY~~
	~~0x22~~ ~~VARIABLE PITCH + SWISS FONT FAMILY~~
	~~0x30~~ ~~DEFAULT PITCH + MODERN FONT FAMILY~~
	~~0x31~~ ~~FIXED PITCH + MODERN FONT FAMILY~~
	~~0x32~~ ~~VARIABLE PITCH + MODERN FONT FAMILY~~
	~~0x40~~ ~~DEFAULT PITCH + SCRIPT FONT FAMILY~~
	~~0x41~~ ~~FIXED PITCH + SCRIPT FONT FAMILY~~
	~~0x42~~ ~~VARIABLE PITCH + SCRIPT FONT FAMILY~~
	~~0x50~~ ~~DEFAULT PITCH + DECORATIVE FONT FAMILY~~
	~~0x51~~ ~~FIXED PITCH + DECORATIVE FONT FAMILY~~
	~~0x52~~ ~~VARIABLE PITCH + DECORATIVE FONT FAMILY~~
	~~This information is determined by querying the font when present~~

Attributes	Description
	~~and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available.~~ ~~The possible values for this attribute are defined by the W3C XML Schema byte datatype.~~

2010-03-25 Stockholm meeting:

Reviewed Shawn's email of 2010-03-18. We chose Choice 2, "Remove the attributes from the standard". Move to Last Call (AMD).

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0038 — WML/PML/DML: Panose-1 Typeface Classification

Status: Closed Without Action

Subject: WML/PML/DML: Panose-1 Typeface Classification

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00011

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §19.2.1.13, "font (Embedded Font Name)", p. 2766 §17.8.3.13, "panose1 (Panose-1 Typeface Classification Number)", p. 766§19.2.1.13, "font (Embedded Font Name)", p. 2768§21.1.2.3.1, "cs (Complex Script Font)", p. 3598§21.1.2.3.3, "ea (East Asian Font)", p. 3606§21.1.2.3.7, "latin (Latin Font)", p. 3614§21.1.2.3.10, "sym (Symbol Font)", p. 3623§21.1.2.3.7, "buFont (Specified)", p. 3638

Related DR(s): none

Nature of the Defect:

The element or attribute name used to represent Panose-1 Typeface Classification Number is sometimes "panose1" (§17.8.3.13 in WordprocessingML) and sometimes "panose" (§19.2.1.13 in PresentationML). Considering the fact that there was a white paper for Panose-2.0, which is not compatible with Panose-1.0: http://www.w3.org/Fonts/Panose/pan2.html "panose1" is better than "panose" to avoid the confusion.

The subclause using "panose1" instead of "panose" is:

- §17.8.3.13, panose1 (Panose-1 Typeface Classification Number), p. 766

The subclauses using "panose" instead of "panose1" are:

- §19.2.1.13, font (Embedded Font Name), p. 2768
- §21.1.2.3.1, cs (Complex Script Font), p. 3598
- §21.1.2.3.3, ea (East Asian Font), p. 3606
- §21.1.2.3.7, latin (Latin Font), p. 3614
- §21.1.2.3.10, sym (Symbol Font), p. 3623
- §21.1.2.3.7, buFont (Specified), p. 3638

Also the type names like "CT_Panose", "ST_Panose" (§22.9.2.8) should be changed to "CT_Panose1", "ST_Panose1".

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Request to add a note regarding changing "Panose" to "Panose1" in narrative.

2009-06-07 Shawn Villaron:

This defect report recommends renaming some of the simple and complex types to bring more clarity to the fact that we are using Panose 1 classification numbers, in light of a whitepaper existing regarding Panose 2. The recommendation in Prague was that we leave the simple and complex types as-is, and make a narrative note that we're indeed talking about Panose 1 classification numbers.

I have reviewed the current version of the 29500 text and we are already clarifying our use of Panose 1 via narrative notes in the documentation. Here is an example of how we do this (the emphasis is mine):

17.8.3.13 panose1 (**Panose-1 Typeface Classification Number**)

This element specifies the **Panose-1 classification number** for the current font using the mechanism defined in §4.2.7.17 of ISO/IEC 14496-22:2007. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then no **Panose-1 information** is available.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="Times New Roman">
  <w:panose1 w:val="02020603050405020304" />
  …
</w:font>
```

The panose1 element specifies its **Panose-1 number** via its val attribute value of `02020603050405020304`. *end example*]

Attributes	Description
val (Value)	Specifies the **Panose-1 classification number** for the font, stored as a series of two digit hexadecimal encodings of each digits of the Panose number. …

I have only found one Panose reference which did not have a nearly immediate clarification regarding Panose 1. Here is the text of that reference:

21.1.2.5 Font Substitution

If any DrawingML element references a font and an appropriate format of the font is not stored within the document, the process of finding a suitable alternative font is known as *font substitution*.

The following elements specify font information: buFont@panose (§21.1.2.4.6); cs@panose (§21.1.2.3.1); ea@panose (§21.1.2.3.3); font@panose (§**19.2.1.13**); latin@panose (§21.1.2.3.7); sym@panose (§21.1.2.3.10).

The exact algorithm which is used for font substitution is highly dependent on the characteristics which are most desirable when performing the substitution: similar appearance of each glyph (to maximize visual familiarity), similar physical characteristics (to minimize changes in line height and breaking), etc. ISO/IEC 29500 recommends that applications look for the closest match to the following attribute values (in descending priority) in order to determine a suitable alternative font: panose, charset, pitchFamily, and typeface; however, applications are free to apply higher-order logic in its place.

I do not believe that there is much value added by providing another narrative reference to Panose 1 classification numbers as this will become very evident to the reader when she looks into the definition of the therein-referenced entities.

It is based on this investigation that I recommend we CLOSE this defect report WITHOUT ACTION.

2009-06-11 Teleconference:

Agreed to Close Without Action.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0039 — Shared, Parts, Font Part: File format for "bitmapped font" is missing

Status: Further Consideration Required

Subject: Shared, Parts, Font Part: File format for "bitmapped font" is missing

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00012

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §15.2.13, "Font Part", p. 159

Related DR(s): none

Nature of the Defect:

The text "* application/x-fontdata specifies that the font shall be stored as a bitmapped font (each glyph is stored as a raster image)" provides no description about the file format of the "bitmapped font". It reminds of ".FNT" in MS-DOS or ".FON" file format introduced in Microsoft Windows 3.0, but there are too many mutually incompatible bitmap font formats.

Solution Proposed by the Submitter:

Clarify "the bitmapped font" file format specification or remove the permission to embed "the bitmapped font".

Schema Change(s) Needed:

No

Editor's Response:

2009-03-24 Prague meeting:

More investigation needed, but the problem is understood.

2009-09-13/15 Bellevue Meeting:

Although another media type for Embedded OpenType Format has been already registered at IANA, we continue to use application/x-fontdata for the compatibility with existing OOXML documents.

Shawn presented the following solution:

Part 1, §15.2.13, "Font Part"

Fonts stored in a Font part can be stored in one of two formats, identified by the associated content type:

- application/x-fontdata ~~specifies that the font shall be stored as a bitmapped font (each glyph is stored as a raster image)~~specifies that the font shall be stored in the Embedded OpenType Format of http://www.w3.org/Submission/2008/SUBM-EOT-20080305.
- application/x-font-ttf specifies that the font shall be stored in a format conforming to ISO/IEC 14496-22:2007.
- [definition for application/vnd.openxmlformats-officedocument.obfuscatedFont goes here if DR-0012 is accepted.]

Closed as proposed.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0040 — WML/DML: Complex scripts

Status: Further Consideration Required

Subject: WML/DML: Complex scripts

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00013, 08-00014, 08-00015, 08-00016, 08-00017, 08-00018, 08-00019

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.3.2.2, "bCs (Complex Script Bold)", p. 281Part 1, §17.3.2.7, "cs (Use Complex Script Formatting on Run)", p. 289Part 1, §17.3.2.17, "iCs (Complex Script Italics)", p. 304Part 1, §17.3.2.20, "lang (Languages for Run Content)", p. 304 Part 1, §17.3.2.39, "szCs (Complex Script Font Size)", p. 335 Part 1, §21.1.2.3.1, "cs (Complex Script Font)", p. 3596 Part 1, §21.1.2.3.3, "ea (East Asian Font)", p. 3605

Related DR(s): none

Nature of the Defect:

The coverage of "complex script", "east asian" "latin", "ascii", and "hansi" is unclear. At least, it should be made clear whether the ranges in Unicode are sufficient to determine whether a script is included in "complex script", or it is dependent on the consumers' text rendering systems and their font resource.

In some groups (e.g., sz, szCs), the scripts are classified into 2 groups: complex and non-complex script. In other groups (e.g., bidi/easiAsia/val attributes in lang element), the scripts are classified into 3 groups: complex script, eastAsia, and latin. Also there are a few groups that the scripts are classified into 4 groups: ascii, cs, eastAsia, hAnsi. In such group, hAnsi is defined as "which does not fall into one of the three categories defined above", so the definition of complex script is important in such group.

In the case of ISO/IEC 14496-4:2004/Amd.26 (the conformance levels of OFF in MPEG stream), all OpenType layout tables are classified as complex scripts, so vertical writing mode in east Asian scripts is also classified as complex scripts.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

We need to provide more documentation.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0041 — WML, Fonts: Font resource search

Status: Last Call

Subject: WML, Fonts: Font resource search

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00020

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.2, "Font Substitution", p. 751

Related DR(s): none

Nature of the Defect:

Regarding the priorities of the information to search an appropriate font resource for a document, there is the following note: "ISO/IEC 29500 recommends that applications looking for the closest match to the following pieces of information (in descending priority) in order to determine a suitable alternative font; however, applications are free to apply more sophisticated logic in its place:

- panose1 (§17.8.3.13)
- sig (§17.8.3.16)
- charset (§17.8.3.2)
- pitch (§17.8.3.14)
- family (§17.8.3.9)
- altName (§17.8.3.1)

This information come from Microsoft Windows GDI (the classifications in charset, pitch, family are designed to be identical with those in GDI), but the order of the priority is different from Microsoft Windows GDI's font substitution algorithm described in Microsoft GDI Technical Article "Windows Font Mapping"

(http://msdn.microsoft.com/en-us/library/ms969909.aspx). The remarkable difference is that "panose1" and "sig" are not used in GDI's font substitution. Therefore, the substituted fonts by ISO/IEC 29500-1 recommendation in §17.8.2 and that by existing Microsoft Office Word can be quite different. The order of the priority described in §17.8.2 should be improved to minimize the difference from GDI's font substitution algorithm.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

The character set is more important than Panose1. Potential bias against Asian character sets.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed.

2010-07-28 Chris Rae:

The above JISC DR represents a query about the ordering of substituted fonts, which seems to be inconsistent with other resources which have similar lists. The notes in the DR itself suggest that Shawn presented a solution (possibly back in Prague) but that further consideration is required. I have the solution Shawn proposed but unfortunately I don't know what the further consideration would have been.

The solution was simply to change the order of the substituted fonts by moving "charset" to the top (in Part 1, 17.8.2). Can we think again about accepting this solution?

2010-07-29 Makoto Murata:

I don't remember if this is good enough. I will see Suzuki-san (the original submitter) tomorrow.

2010-08-19 teleconference:

Accepted the proposal with the following change: Move sig to the first position, and charset to the second position. Moved to Last Call.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0042 — WML, Fonts: notTrueType attribute missing from list

Status: Last Call

Subject: WML, Fonts: notTrueType attribute missing from list

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00021

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.12, "notTrueType (Raster or Vector Font)", p. 766

Related DR(s): none

Nature of the Defect:

These attributes are described as: "This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available."

However, the notTrueType attribute is not in the list of attributes in §17.8.2.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

More investigation needed.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed.

2010-08-13 Chris Rae:

This looks like an easy one - the right thing to do seems to be just to add notTrueType to the list in 17.8.2. It seems a suspiciously old DR to have such an easy answer, though - did anyone have any other ideas?

2010-08-19 teleconference:

Adopted as proposed. Moved to Last Call.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0043 — WML, Fonts: notTrueType attribute and bitmap fonts

Status: Further Consideration Required

Subject: WML, Fonts: notTrueType attribute and bitmap fonts

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00022

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.12, "notTrueType (Raster or Vector Font)", p. 766

Related DR(s): none

Nature of the Defect:

ISO/IEC 14496-22 permits implementation of a font file including only bitmap and no outline (so-called "sbit font" is popular in Macintosh). It is not clear whether such bitmap-only OFF should be declared as notTrueType.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2009-09-13/15 Bellevue Meeting:

Shawn presented the solution shown below. However, more work is needed.

Agreed; this subclause is unclear with respect to its intent. The following changes will be made in §17.8.3.12 to clarify the meaning:

17.8.3.12 notTrueType (~~Raster or Vector~~Not a TrueType Font)

This element specifies that this font is not stored in a format conforming to ISO/IEC 14496-22:2007~~is not a TrueType or OpenType font, but is rather a raster or vector font~~. This information can be used ~~as defined~~ in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.
If this element is omitted, then the font shall be assumed to be stored in a format conforming to ISO/IEC 14496-22:2007~~a TrueType or OpenType font~~.

[*Example*: Consider the following information stored for a single font:

```
<w:font w:name="JonsFont">
 <w:notTrueType w:val="true" />
 …
</w:font>
```

The notTrueType element specifies via its val attribute value of `true` that this font is not a TrueType ~~a raster or vector~~ font. *end example*]

2010-07-28 Chris Rae:

This is another defect report to which Shawn presented a solution (this time in Bellevue in 2009) but which is marked as needing more discussion. It covers vague wording around the storage of raster font data (as opposed to TrueType/OpenType).

Does anyone recall what objections there were, if any? Again, I'd be happy if we can talk about this on the call tomorrow but I understand if the timing is a bit too close.

2010-07-29 Makoto Murata:

I do not remember. Will ask Suzuki-san.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0044 — WML, Fonts: Support for commas in font family/subfamily/full names

Status: Closed Without Action

Subject: WML, Fonts: Support for commas in font family/subfamily/full names

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00023

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.1, "altName (Alternate Names for Font)", p. 751

Related DR(s): REF _Ref226534183 \h DR 09-0050 — SML, Worksheets: comma delimiter between font name and type

Nature of the Defect:

ISO/IEC 14496-22 permits commas in font family/subfamily/full names. It should be made clear how names containing commas can be represented in a comma-delimited list.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Currently, this is not supported. Closed without change.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0045 — WML, Fonts: Character encodings of font names

Status: Further Consideration Required

Subject: WML, Fonts: Character encodings of font names

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00024

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.1, "altName (Alternate Names for Font)", p. 751

Related DR(s): none

Nature of the Defect:

The Open Font Format described in ISO/IEC 14496-22 can include various font names in various character encodings. For example, a name table of OFF can include 2 font family names, one coded by Unicode, the other coded by ShiftJIS. It should be made clear that OOXML relies on XML parsers for encodings.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed,

Two issues were identified during the discussion:

1. How do we determine the first font?
2. We need to change the word “parser” to “processor”

2010-08-13 Chris Rae:

This one looks fairly simple - I think it can be resolved by adopting the changes shown below.

“When an application cannot locate a font using the primary name stored on the font attribute of the font element (§17.8.3.10), it should use each alternate name in term to attempt to locate the font, and use the first font for which is locates a match.

Font names stored using this element shall be specified in the encoding specified by the Fonts part in its XML declaration; the name of the font will be interpreted by the XML parser.

If this element is omitted, then no alternate names are present for the parent font.”

2010-08-17 Jirka Kosek:

I understand to your intent, but result of XML parsing is always Unicode string. So speaking about encodings here seems quite strange to me.

If there is really need for clarifying this DR, what about: "Font names stored using this element shall consist of Unicode characters."

2010-08-17 Chris Rae:

Good point - I think Jirka's wording is much clearer.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0046 — WML, Fonts: Misleading example

Status: Further Consideration Required

Subject: WML, Fonts: Misleading example

Qualifier: Editorial defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00025

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.2, "charset (Character Set Supported By Font)", p. 753

Related DR(s): none

Nature of the Defect:

"The charset element specifies via its characterSet attribute value of GBK that this font uses the GB-2312 character set."

The example is misleading because GBK is not identical to GB 2312. See http://www.iana.org/assignments/charset-reg/GBK.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed. GBK is a proper superset of GB 2312.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution (see below). Closed as proposed.

2010-08-02 Chris Rae (in private mail):

This DR centered around a mismatch in wording inside an example regarding the mixed use of "GBK" and "GB-2312" character set descriptions. I believe Shawn's solution involved just changing "GB-2312" to "GBK".

Changes to Part 1: Y **Part 2: N Part 3: N Part 4: N**

DR 09-0047 — WML, Fonts: Identifying a face in an embedded font file

Status: Last Call

Subject: WML, Fonts: Identifying a face in an embedded font file

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00026, 01-00027, 01-00028, 01-00029

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.3, "embedBold (Bold Style Font Style Embedding)", p. 753Part 1, §17.8.3.4, "embedBoldItalic (Bold Italic Font Style Embedding)", p. 755Part 1, §17.8.3.5, "embedItalic (Italic Font Style Embedding)", p. 757Part 1, §17.8.3.6, "embedRegular (Regular Font Style Embedding)", p. 759

Related DR(s): none

Nature of the Defect:

Although the embedded font "file" in the package is specified by the attribute id, a "file" defined by ISO/IEC 14496-22 can include multiple faces by TrueType Collection (TTC) file format. So, the procedure to identify a face in an embedded font file in the package should be described.

Solution Proposed by the Submitter:

There are 3 scenarios:

1. Forbid the TTC file format; only TTF or OTF file formats including single face are permitted. A document producer has to convert TTC to TTF when it embeds a font into OOXML.
2. Use name or altName in the root element to select a face from an embedded TTC.

Add a new attribute to specify a face in TTC.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed.

2010-08-13 Chris Rae:

This is another relatively simple DR covering the way in which multiple font faces are stored. I think it can be addressed by following change to §15.2.13:

"Fonts stored in a Font part can be stored in one of two formats, identified by the associated content type:

- application/x-fontdata specifies that the font shall be stored as a bitmapped font (each glyph is stored as a raster image)
- application/x-font-ttf specifies that the font shall be stored in a format conforming to ISO/IEC 14496-22:2007

If a font is stored in the ISO/IEC 14496-22:2007 format, it shall only be used when stored as an individual font; font collections must be converted into individual fonts before they are embedded using this part.

A package shall contain zero or more Font parts, and for each that exists, that part shall be the target of an explicit relationship in the Font Table (§11.3.5), or Presentation (§13.3.6) part."

2010-08-19 teleconference:

If a font is stored in the ISO/IEC 14496-22:2007 format, it shall only be used when stored as an individual font. [*Note:* Font collections should be converted into individual fonts before they are embedded using this part. *end note*]"

Accepted the intent of the change proposed by Chris (see mail 2010-08-13), but rewritten as above. Moved to Last Call.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0048 — WML, Fonts: Panose-1 classification mechanism missing

Status: Closed; in COR1

Subject: WML, Fonts: Panose-1 classification mechanism missing

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00030

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.8.3.13, "panose1 (Panose-1 Typeface Classification Number)", p. 767

Related DR(s): none

Nature of the Defect:

The first sentence is: "This element specifies the Panose-1 classification number for the current font using the mechanism defined in 4.2.7.17 of ISO/IEC 14496-22:2007." However, 4.2.7.17 of ISO/IEC 14496-22 does not provide any such mechanism.

Solution Proposed by the Submitter:

Change this sentence to the following: "This element specifies the Panose-1 classification number shown in 4.2.7.17 of ISO/IEC 14496-22:2007." It might be a good idea to reference to the original Panose specification, available at http://www.panose.com.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2009-06-07 Shawn Villaron:

This DR suggests that our current reference to the Panose 1 classification number is invalid. We currently refer to §**4.2.7.17 of ISO/IEC 14496-22:2007**. I don't have access to this document. Does anyone, and if you do, would you mind looking this up for me? I'd like to know if we simply have the reference wrong (e.g., the wrong clause) or if the referenced document is in error.

2009-06-11 Teleconference:

We will *not* make a reference to Panose.com.

Shawn will update the text.

Agreed to move to Last Call.

2009-06-07 Shawn Villaron:

Part 1, §17.8.3.13, "panose1 (Panose-1 Typeface Classification Number)", p. 766:

This element specifies the Panose-1 classification number ~~for the current font using the mechanism defined~~ shown in §4.2.7.17 of ISO/IEC 14496-22:2007. This information can be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0049 — WML, Fields: SYMBOL switch proper charset name usage

Status: Closed; in AMD3

Subject: WML, Fields: SYMBOL switch proper charset name usage

Qualifier: Editorial defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00024

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.16.5.61, "SYMBOL", p. 1391.

Related DR(s): none

Nature of the Defect:

"`\j` Interprets text in field-argument as the value of a SHIFT-JIS character". "SHIFT-JIS" should be replaced by Windows-31J, which is an IANA charset name.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

None

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed.

2009-10-29 Makota Murata:

Japan proposed to replace shift_jis by Windows-31J. In Belleview, there was opposition to this proposal from Microsoft. However, Japanese experts (including Ishizaka-san of Microsoft Japan) argued that shift_jis has so many variations and that Windows-31J references to a particular variation used by Microsoft.

In this mail, I would like to present two supporting evidences. I would propose to accept the Japanese proposal and close this DR.

An authoritative document is the IANA charset registry, available at http://www.iana.org/assignments/character-sets. I was involved in the latest revision of the charsets shift_jis and windows-31j. I spoke with Microsoft Japan people and did this revision.

> Name: Windows-31J

> MIBenum: 2024

> Source: Windows Japanese. A further extension of Shift_JIS

> to include NEC special characters (Row 13), NEC

> selection of IBM extensions (Rows 89 to 92), and IBM

> extensions (Rows 115 to 119). The CCS's are

> JIS X0201:1997, JIS X0208:1997, and these extensions.

> This charset can be used for the top-level media type "text",

> but it is of limited or specialized use (see RFC2278).

> PCL Symbol Set id: 19K

> Alias: csWindows31J

...

> Name: Shift_JIS (preferred MIME name)

> MIBenum: 17

> Source: This charset is an extension of csHalfWidthKatakana by

> adding graphic characters in JIS X 0208. The CCS's are

> JIS X0201:1997 and JIS X0208:1997. The

> complete definition is shown in Appendix 1 of JIS

> X0208:1997.

> This charset can be used for the top-level media type "text".

> Alias: MS_Kanji

> Alias: csShiftJIS

Another important source of information is XML Japanese profile, available at: http://www.w3.org/TR/japanese-xml/#sjis. I am the editor of this document.

> 5.3 Shift-JIS

>

> This technical report and [XML] treat Shift-JIS, an ordinary

> Japanese charset, as a CES that represents Japanese characters and

> [US-ASCII] characters in [ISO/IEC10646 (all parts)] or [Unicode 3.2].

> For full interoperability in the Internet, migration from Shift-JIS to

> UTF-8/UTF-16 is highly recommended.

...

> There are four major conversion tables from Shift-JIS to [ISO/IEC10646

> (all parts)] or [Unicode 3.2]. This technical report names them

> x-sjis-unicode-0_9, x-sjis-jisx0221-1995, windows-31J, and

> x-sjis-jdk1_1_7, respectively. These conversion tables are not identical

> to each other.

2009-10-29 Teleconference:

WG4 agreed with JP's proposal. Closed; move to "Last Call (AMD)".

The exact changes are as follows:

Part 1, §17.16.5.61, "SYMBOL", p. 1391.

\h	…

\j	Interprets *text* in *field-argument* as the value of a ~~SHIFT-JIS~~Windows-31J character.
\s *field-argument*	…

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0050 — SML, Worksheets: comma delimiter between font name and type

Status: Closed; in COR1

Subject: SML, Worksheets: comma delimiter between font name and type

Qualifier: Technical Defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00032

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §18.3.1.39, "evenHeader", p. 1804

Related DR(s): none

Nature of the Defect:

Re &"font name,font type", since ISO/IEC 14496-22 permits commas in font family/subfamily/full names, it is not clear which "," is the delimiter between the font name and font type.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Change DR qualifier to "Technical Defect", write text to say that "the lexically first comma is the one recognized as the delimiter". Closed.

The solution to this DR should be published in a Technical Corrigendum.

The exact changes are as follows:

Part 1, §18.3.1.39, “evenHeader”, p. 1804

&"*font name*,*font type*" - code for "text font name" and "text font type", where *font name* and *font type* are strings specifying the name and type of the font, separated by a comma. When a hyphen appears in *font name*, it means "none specified". Both of *font name* and *font type* can be localized values. Although ISO/IEC 14496-22 permits commas in font family/subfamily/full names, name and font type, the lexically first comma in the string is the one recognized as the separating comma.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0051 — SML, Shared String Table: charset value duplicate

Status: ??

Subject: SML, Shared String Table: charset value duplicate

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00033

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §18.4.1, "charset (Character Set)", p. 1898

Related DR(s): none

Nature of the Defect:

There are two entries, namely "HANGUEL_CHARSET" and "HANGUL_CHARSET", for the integer value 129.

Solution Proposed by the Submitter:

Remove one of them.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Remove "HANGUEL_CHARSET". Closed.

The solution to this DR should be published in an Amendment.

The exact changes are as follows:

Part 1, §18.4.1, "charset (Character Set)", p. 1898

INT Value	Character Set
…	…
~~129~~	~~HANGEUL_CHARSET~~
129	HANGUL_CHARSET
…	…

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0052 — SML, Styles: family attribute range issue

Status: Closed; in AMD1

Subject: SML, Styles: family attribute range issue

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00034

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §18.8.18, "family (Font Family)", p. 1952

Related DR(s): none

Nature of the Defect:

The attribute description allows integer values ranging from 0–255, while the table allows values ranging from 0–5 only.

Ideally, as in WordprocessingML (§17.8.3.9), use the simple type ST_FontFamily, shown in §17.18.30. If this is not possible, the attribute value should be restricted.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-06-09/10 Shawn Villaron:

The attribute should be restricted in the schema to the values from 0–5.

Part 1, §18.8.18, "family (Font Family)", p. 1952 will be updated as follows:

The font family this font belongs to. ~~A font family is a set of fonts having common stroke width and serif characteristics. This is system level font information.~~ The font name overrides when there are conflicting values.

~~Value~~	~~Font Family~~
~~0~~	~~Not applicable.~~
~~1~~	~~Roman~~
~~2~~	~~Swiss~~
~~3~~	~~Modern~~
~~4~~	~~Script~~
~~5~~	~~Decorative~~

Attributes	Description
val (Value)	The font family this font belongs to ~~value of an integer, where each value corresponds to a different character set. This attribute is restricted to values ranging from 0 to 255.~~ The possible values for this attribute are defined by the ST_FontFamily simple type~~W3C XML Schema int datatype~~.

[*Note*: The W3C XML Schema definition of this element's content model (~~CT_IntProperty~~CT_FontFamily) is located in §xx. *end note*]

Part 1, §18.18.x, "ST_FontFamily (Font Family)", new subclause:

This simple type specifies a font family. A font family is a set of fonts having common stroke width and serif characteristics. This is system level font information.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type is restricted to the values listed in the following table:

Value	Font Family

Value	Font Family
0	Not applicable.
1	Roman
2	Swiss
3	Modern
4	Script
5	Decorative

Referenced By
fontFamily@val (§xx)

[*Note*: The W3C XML Schema definition of this simple type's content model (ST_ST_FontFamily) is located in §xx. *end note*]

Part 1, §A.2, "SpreadsheetML", p. 4481, lines 3792–3811

```
<xsd:complexType name="CT_Font">
  <xsd:choice maxOccurs="unbounded">
    …
    <xsd:element name="family" type="~~CT_IntProperty~~CT_FontFamily" minOccurs="0"
          maxOccurs="1"/>
    …
  </xsd:choice>
</xsd:complexType>
```

Part 1, §B.2, "SpreadsheetML", p. 4788, lines 4031–4046

```
sml_CT_Font =
  …
  | element family { ~~sml_CT_IntProperty~~sml_CT_FontFamily }?
  …
  | element scheme { sml_CT_FontScheme }?)+
```

Part 4, §A.2, "SpreadsheetML", p. 951, lines 3820–3839

```
<xsd:complexType name="CT_Font">
  <xsd:choice maxOccurs="unbounded">
    …
    <xsd:element name="family" type="~~CT_IntProperty~~CT_FontFamily" minOccurs="0"
          maxOccurs="1"/>
    …
  </xsd:choice>
</xsd:complexType>
```

Part 4, §B.2, "SpreadsheetML", p. 1289–1290, lines 4057–4072

```
sml_CT_Font =
  …
  | element family { ~~sml_CT_IntProperty~~sml_CT_FontFamily }?
  …
  | element scheme { sml_CT_FontScheme }?)+
```

Part 1, §A.2, "SpreadsheetML", new type:

```
   <xsd:complexType name="CT_FontFamily">
       <xsd:attribute name="val" type="ST_FontFamily" use="required"/>
   </xsd:complexType>

   <xsd:simpleType name="ST_FontFamily">
       <xsd:restriction base="xsd:integer">
          <xsd:minInclusive value="0"/>
          <xsd:maxInclusive value="14"/>
       </xsd:restriction>
   </xsd:simpleType>
```

Part 1, §B.2, "SpreadsheetML", new type:

```
sml_CT_FontFamily = attribute val { sml_ST_FontFamily }
sml_ST_FontFamily =
  xsd:integer { minInclusive = "0" maxInclusive = "14" }
```

Part 4, §A.2, "SpreadsheetML", new type:

```
<xsd:complexType name="CT_FontFamily">
    <xsd:attribute name="val" type="ST_FontFamily" use="required"/>
</xsd:complexType>

<xsd:simpleType name="ST_FontFamily">
    <xsd:restriction base="xsd:integer">
       <xsd:minInclusive value="0"/>
       <xsd:maxInclusive value="14"/>
    </xsd:restriction>
</xsd:simpleType>
```

Part 4, §B.2, "SpreadsheetML", new type:

```
sml_CT_FontFamily = attribute val { sml_ST_FontFamily }
sml_ST_FontFamily =
  xsd:integer { minInclusive = "0" maxInclusive = "14" }
```

2009-06-11 Rick Jelliffe:

These styles are related to those now defined in ISO/IEC 14496-22:2007 Information technology -- Coding of audio-visual objects -- Part 22: Open Font Format

Open Font can be downloaded from http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html. Search 14496-22 and click through license to ZIP and PDF.

Annex B, p 347 gives the more recent list of font classes (and font subclasses). I presume this has the advantage of supporting Adobe/Apple fonts better, or of being more future proofed. I also presume, because of Microsoft's involvement in Open Font, that it is reasonable to expect that if future versions of Office supported more font classes, it would do so in conformance to these styles.

I also suggest that these font classes are the appropriate things to use, certainly for the strict schema, because we do not want to arbitrarily limit the capacity of an OOXML application to utilize as much of OpenFont as it needs for general application use.

In other words, the list in OOXML is obsolete, and suitable for transitional. We should align with Open Font.

I suggest the following (my additions in green):

18.18.x ST_FontFamily (Font Family Class)

This simple type specifies a font family class using the Class values specified in Annex B of of ISO/IEC 14496-22:2007. A font family class is a set of fonts having common stroke width and serif characteristics. This is system level font information.

This simple type's contents are a restriction of the W3C XML Schema unsignedInt datatype.

This simple type is restricted to the values listed in the following tabless .Note: the equivalent names used by Ecma 367 are given in parentheses.

Table 1.

Value	Font Family Class
0	No classification
1	Oldstyle serifs (Roman)
2	Transitional serifs (Swiss)
3	Modern serifs (Modern)
4	Clarenden serifs (Script)
5	Slab serifs (Decorative)

Table 2.

Value	Font Family Class
6,11, 13, 14	(Reserved)
7	Freeform serifs
8	Sans serifs
9	Ornamentals
10	Scripts
12	Symbolic

A producer or consumer may substitute values in table 1 for values of table 2 for compatibility with ECMA 367.

2009-06-11 Teleconference:

On 2009-06-11, in email titled “Alignment with ISO Open Font”, Rick Jelliffe suggested that we accommodate Open Font. After some discussion, Shawn proposed that we adopt Shawn’s proposed disposition moving this DR to Last Call , and that we encourage Rick to submit a separate DR to reconcile the differences between 29500 font family support and the Open Font standard. Agreed.

2009-06-12 Shawn Villaron:

At yesterday’s phone conference we discussed my original proposal and your response to it where you suggest alignment with ISO Open Font. The recommendation was that we accept the defect report as-is and ask that you log a separate defect report recommending improving 29500 such that we attain this alignment. I believe

that this is the right way to handle this, but I felt it was important for you to have an opportunity to provide feedback on this recommendation before we consider it official. What do you think?

If you agree that a separate defect report should be logged, I'd request that we look at 29500 comprehensively here. The proposed change would improve text support in SpreadsheetML, but I think the intent was to improve it for that, plus the other MLs.

I'd be happy to work with you on drafting the separate defect report if you're so interested.

I believe the consensus was that if you agree with the recommendation, we'd move this to LAST CALL and consider it for COR1.

2009-06-15 Rick Jelliffe:

I am happy with that approach.

However, I would still ask that the datatype allow maxInclusive=14, for possible forward compatibility, with the missing parts reserved and undocumented.

2009-06-15 Shawn Villaron:

That seems like the right compromise. The font family table above will be extended with values 6–14, as follows:

Value	Font Family
0	Not applicable.
1	Roman
2	Swiss
3	Modern
4	Script
5	Decorative
6	Reserved for future use
7	Reserved for future use
8	Reserved for future use
9	Reserved for future use

Value	Font Family
10	Reserved for future use
11	Reserved for future use
12	Reserved for future use
13	Reserved for future use
14	Reserved for future use

And the maxInclusive value="5" will be changed to 14 for the schemas in both Parts 1 and 4.

2009-06-26 Editor:

Goes into AMD1.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: Y

DR 09-0053 — PML, Presentation: Attribute name pitchFamily is misleading

Status: Closed; in COR1

Subject: PML, Presentation: Attribute name pitchFamily is misleading

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00035

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §19.2.1.13, "font (Embedded Font Name)", p. 2769

Related DR(s): none

Nature of the Defect:

The attribute name pitchFamily is misleading. It should be pitchAndFamily.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

A prose change explaining this attribute is preferred to changing the name of the attribute. Will propose wording.

2009-06-07 Shawn Villaron:

In reviewing this DR, a suggestion is made to improve the readability of the standard by changing the name of the pitchFamily attribute in PresentationML, to clarify that this attribute covers the pitch and the family for the font.

In Prague, the suggestion was to make a narrative fix to avoid a breaking file format change (which is what changing the attribute name would be). I've reviewed the current prose and am unconvinced that a narrative fix is required. Here is the current text of the at-issue clause (the red emphasis is mine):

19.2.1.13 font (Embedded Font Name)

This element specifies specific properties describing an embedded font. Once specified, this font is available for use within the presentation. Within a font specification there can be regular, bold, italic and boldItalic versions of the font specified. The actual font data for each of these is referenced using a relationships file that contains links to all available fonts. This font data contains font information for each of the characters to be made available in each version of the font.

…

Attributes	Description
pitchFamily (Similar Font Family) Namespace: …/drawingml/2006/main	**Specifies the font pitch as well as the font family** for the corresponding font. Because the value of this attribute is determined by a byte variable this value shall be interpreted as follows: …

Given the red bolded text above, I think that the documentation is pretty clear. And so it's my recommendation that we Close this Without Action.

2009-06-11 Teleconference:

Action: Murata-san will provide some text improvements.

2009-06-21 Makoto Murata:

I propose that we add the following note to the Description:

"Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits."

2009-06-22/24 Copenhagen Meeting:

Add the note proposed by Murata-san in email of 2009-06-21. Moved to Closed.

The exact changes are as follows:

Part 1, §19.2.1.13, "font (Embedded Font Name)", p. 2769

Attributes	Description
pitchFamily (Similar Font Family) Namespace: …/drawingml/2006/main	Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by an octet value ~~byte variable~~ this value shall be interpreted as follows: … This information is determined by querying the font when present and shall not be modified when the font is not available. This information can be used in font substitution logic to locate an appropriate substitute font when this font is not available. [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0054 — DML, run formatting: Attribute name pitchFamily is misleading

Status: Closed; in COR1

Subject: PML, run formatting: Attribute name pitchFamily is misleading

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00036, 08-00037, 08-00038, 08-00039, 08-00040

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §21.1.2.3.1, "cs (Complex Script Font", p. 3596Part 1, §21.1.2.3.3, "ea (East Asian Font)", p. 3605 Part 1, §21.1.2.3.7, "latin (Latin Font)", p. 3613Part 1, §21.1.2.3.10, "sym (Symbol Font)", p. 3622Part 1, §21.1.2.4.6, "buFont (Specified)", p. 3636

Related DR(s): none

Nature of the Defect:

The attribute name pitchFamily is misleading. It should be pitchAndFamily.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

A prose change explaining this attribute is preferred to changing the name of the attribute. Will propose wording.

2009-06-07 Shawn Villaron:

My research regarding this DR yielded the same results as those for エラー! 参照元が見つかりません。. As such, I'm making the same recommendation here as I am for DR 09-0053: we should Close this Without Action.

2009-06-11 Teleconference:

Action: Murata-san will provide some text improvements.

2009-06-21 Makoto Murata:

I propose that we add the following note to the Description:

"Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits."

2009-06-22 Copenhagen Meeting:

Add the note proposed by Murata-san in email of 2009-06-21. Moved to Closed.

The exact changes are as follows:

Part 1, §21.1.2.3.1, "cs (Complex Script Font)", pp. 3598–3599

Attributes	Description
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by an octet value ~~byte variable~~ this value shall be interpreted as follows: … [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.

Part 1, §21.1.2.3.3, "ea (East Asian Font)", pp. 3606–3607

Attributes	Description
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by an octet value ~~byte variable~~ this value shall be interpreted as follows: … [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.

Part 1, §21.1.2.3.7, "latin (Latin Font)", pp. 3614–3615

Attributes	Description
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by an octet value ~~byte variable~~ this value shall be interpreted as follows: … [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.

Part 1, §21.1.2.3.10, "sym (Symbol Font)", pp. 3623–3624

Attributes	Description
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by an octet value ~~byte variable~~ this value shall be

	interpreted as follows: … [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.

Part 1, §21.1.2.4.6, "buFont (Specified)", p. 3638–3639

Attributes	Description
pitchFamily (Similar Font Family)	Specifies the font pitch as well as the font family for the corresponding font. Because the value of this attribute is determined by an octet value ~~byte variable~~ this value shall be interpreted as follows: … [*Note*: Although the attribute name is pitchFamily, the integer value of this attribute specifies the font family with higher 4 bits and the font pitch with lower 4 bits. *end note*] The possible values for this attribute are defined by the W3C XML Schema byte datatype.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0055 — PML, Presentation: Type of the attribute pitchFamily is too loose

Status: Further Consideration Required

Subject: PML, Presentation: Type of the attribute pitchFamily is too loose

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00041

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §19.2.1.13, "font (Embedded Font Name)", p. 2769

Related DR(s): none

Nature of the Defect:

The type of the attribute pitchFamily is defined as "W3C XML Schema byte datatype", but that is too loose. Provide an enumerated list or the union of ranges 00–02, 10–12, 20–22, 30–32, 40–42, and 50–52. Observe that the higher 4 bits, which represents the typeface family (see §17.8.3.9 of WordprocessingML), must be 0x0–0x5, and that the lower 4 bits, which represents the pitch (see §17.8.3.14 of WordprocessingML), must be 0x0–0x2.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

No

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2010-08-13 Chris Rae:

This DR covers the loose specification of the pitchFamily attribute, and requests a tighter definition. I think we can effect this by creating a new simple type for pitchFamily in section 19.7 (PML Simple Types) with the definition below, and then using that simple type in the definition of pitchFamily.

The XSD changes follow:

```
<xsd:simpleType name="ST_pitchFamily">
  <xsd:restriction base="xsd:byte">
    <xsd:enumeration value="00"/>
    <xsd:enumeration value="01"/>
    <xsd:enumeration value="02"/>
    <xsd:enumeration value="16"/>
    <xsd:enumeration value="17"/>
    <xsd:enumeration value="18"/>
    <xsd:enumeration value="32"/>
    <xsd:enumeration value="33"/>
    <xsd:enumeration value="34"/>
    <xsd:enumeration value="48"/>
    <xsd:enumeration value="49"/>
    <xsd:enumeration value="50"/>
    <xsd:enumeration value="64"/>
    <xsd:enumeration value="65"/>
    <xsd:enumeration value="66"/>
    <xsd:enumeration value="80"/>
    <xsd:enumeration value="81"/>
    <xsd:enumeration value="82"/>
  </xsd:restriction>
</xsd:simpleType>
```

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0056 — DML, Main: Type of the attribute pitchFamily is too loose

Status: Further Consideration Required

Subject: DML, Main: Type of the attribute pitchFamily is too loose

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00042, 08-00043, 08-00044, 08-00045, 08-00046

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §21.1.2.3.1, "cs (Complex Script Font)", p. 3596 Part 1, §21.1.2.3.3, "ea (East Asian Font)", p. 3605 Part 1, §21.1.2.3.7, "font (Embedded Font Name)", p. 3613Part 1, §21.1.2.3.10, "sym (Symbol Font)", p. 3622 Part 1, §21.1.2.4.6, "buFont (Specified)", p. 3636

Related DR(s): none

Nature of the Defect:

The type of the attribute pitchFamily is defined as "W3C XML Schema byte datatype", but that is too loose. Provide an enumerated list or the union of ranges 00–02, 10–12, 20–22, 30–32, 40–42, and 50–52. Observe that the higher 4 bits, which represents the typeface family (see §17.8.3.9 of WordprocessingML), must be 0x0–0x5, and that the lower 4 bits, which represents the pitch (see §17.8.3.14 of WordprocessingML), must be 0x0–0x2.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2010-08-17 Chris Rae:

Like DR 09-0055, this DR covers the loose specification of the pitchFamily attribute, and requests a tighter definition. In the same way as I proposed for 09-0055, I think we can effect this by creating a new simple type for pitchFamily in §19.7 (PML Simple Types) with the definition below, and then using that simple type in the definition of pitchFamily.

2009-03-24 Prague meeting:

Understood; more work needed.

2010-08-13 Chris Rae:

This DR covers the loose specification of the pitchFamily attribute, and requests a tighter definition. I think we can effect this by creating a new simple type for pitchFamily in section 19.7 (PML Simple Types) with the definition below, and then using that simple type in the definition of pitchFamily.

The XSD changes follow:

```
<xsd:simpleType name="ST_pitchFamily">
  <xsd:restriction base="xsd:byte">
    <xsd:enumeration value="00"/>
    <xsd:enumeration value="01"/>
    <xsd:enumeration value="02"/>
    <xsd:enumeration value="16"/>
    <xsd:enumeration value="17"/>
    <xsd:enumeration value="18"/>
    <xsd:enumeration value="32"/>
    <xsd:enumeration value="33"/>
    <xsd:enumeration value="34"/>
    <xsd:enumeration value="48"/>
    <xsd:enumeration value="49"/>
    <xsd:enumeration value="50"/>
    <xsd:enumeration value="64"/>
    <xsd:enumeration value="65"/>
    <xsd:enumeration value="66"/>
    <xsd:enumeration value="80"/>
    <xsd:enumeration value="81"/>
    <xsd:enumeration value="82"/>
  </xsd:restriction>
</xsd:simpleType>
```

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0057 — PML, Presentation: Allow aliases as values for attribute typeface

Status: Closed Without Action

Subject: PML, Presentation: Allow aliases as values for attribute typeface

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00047

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §19.2.1.13, "font (Embedded Font Name)", p. 2769

Related DR(s): none

Nature of the Defect:

The attribute typeface can be used for selecting an alternate font. However, unlike the attribute altName of WordprocessingML (§17.8.3.1), this attribute cannot specify aliases as attribute values.

Solution Proposed by the Submitter:

Allow typeface to specify aliases.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2010-03-18 Shawn Villaron:

So this is not so much a defect report but rather a feature request. I think we need to consider whether or not we're going to use defect reports to add functionality to the standard.

I can say that the design of fonts in presentations was not intended to include a list of alternative fonts (so called "aliases"). So this was not an omission in documentation.

That said, I'm not against this suggested improvement. But I'm not sure that the recommended approach is the correct one. In the submission, the recommendation is made to allow the typeface attribute on the font element to specify alternative fonts. I don't believe this is the correct approach as this would break existing implementations which use this attribute (that is, the implementations would be expecting one typeface in the value and not be prepared to parse out multiple entries from a single string). So I firmly believe we should not take this approach.

There are other approaches we could take. We could consider adding an optional attribute to the font element (altName?) and have that specify a comma-delimited list of alternative font. We could also consider adding an optional child element to do the same thing.

Bottom line, I think we need to decide on our engineering approach to this problem, if we believe we should add this support, and if we believe it should be done via a defect report. Hopefully we can discuss this in Stockholm next week.

2010-03-25 Stockholm meeting:

We are interested in supporting this suggestion; however, it is a request for a new feature, not a defect. Close without action. Suggest the submitter propose this in a future amendment/revision.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0058 — DML, Main: Possible values of attribute script are unclear

Status: Further Consideration Required

Subject: DML, Main: Possible values of attribute script are unclear

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00048

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §20.1.4.1.16, "font (Font)", p. 3105

Related DR(s): none

Nature of the Defect:

The attribute script is very unclear. What is the value of this attribute? We can imagine four possibilities:

1. ISO 15924 script names. Also the extra attribute to interchange ISO 639 language code is expected.
2. Open Font Format 4-byte script tags defined in §5.4.1 of ISO/IEC 14496-22. Also the extra attribute to interchange Open Font Format 4-byte language tags (§5.4.2 of ISO/IEC 14496-22) is expected.
3. ST_Lang defined in ISO/IEC 29500-1, §22.9.2.6.
4. Unicode subrange specifications like ISO/IEC 29500-1, §17.8.3.16.

The examples shown in §20.1.4.1.24 and §20.1.4.1.25 suggest Option 1. However, Option 2 is most appropriate, since the value can easily be extracted from concrete font files, and we can search for compatible fonts from the given attribute.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2009-09-13/15 Bellevue Meeting:

<<solution to go here once I have it from Shawn>>

Shawn presented a solution. Closed as proposed.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0059 — DML, Main: Relationship between the symbol font and Symbol character set

Status: Further Consideration Required

Subject: DML, Main: Clarify relationship between the symbol font and Symbol character set

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00049

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §21.1.2.3.10, "sym (Symbol Font)", p. 3622

Related DR(s): none

Nature of the Defect:

The relationship between the symbol font and Symbol character set (used in the charset attribute description) is unclear. Are they expected to be identical?

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Understood; more work needed.

2009-09-13/15 Bellevue Meeting:

Shawn presented a solution. However, more work is needed.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0060 — WML, Fonts: Names should be registered at IANA

Status: Further Consideration Required

Subject: WML, Fonts: Names should be registered at IANA

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00050

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 4, §9.4.1.1, "Additional attribute for charset element", p. 28

Related DR(s): none

Nature of the Defect:

In comparison with IANA names listed in http://www.iana.org/assignments/character-sets, the table includes many names which are not registered in IANA. Also some entries are described with incorrect IANA names.

Solution Proposed by the Submitter:

1. 0x80: shift_jis should be replaced by Windows-31J.
2. 0x81: Microsoft Windows Codepage 949 should be registered in IANA, and use the name in the description.
3. 0x82: Microsoft Windows Codepage 1361 should be registered in IANA and use the name in the description.
4. 0x86: GB2312 should be replaced by GBK.
5. 0x88: Microsoft Windows Codepage 950 should be registered in IANA, and use the name in the description, because there are various Big5 diversions.

0xDE: windows-874 is not registered in IANA. It should be registered.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Apparently, this is clean-up from the BRM where it was agreed that code pages names should be replaced.

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0061 — Shared MLs, Shared Simple Types: Constrain ST_Panose value set

Status: Further Consideration Required

Subject: Shared MLs, Shared Simple Types: Constrain ST_Panose value set

Qualifier: Technical defect

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00051

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §22.9.2.8, "ST_Panose (Panose-1 Number)", p. 4326

Related DR(s): none

Nature of the Defect:

ST_Panose is declared as unrestricted 20 hexadecimal digits. According to the official definition of Panose-1 (http://www.panose.com/ProductsServices/pan1.aspx) the valid Panose number has a more restricted range (in most cases, 0–10).

Solution Proposed by the Submitter:

ST_Panose should be declared as the type which permits the valid number only.

Schema Change(s) Needed:

Yes, (which?)

Replace Panose-1 type definition with something like the following:

```
<xsd:simpleType name="Panose1">
  <xsd:restriction base="xsd:string">
```

```
    <xsd:pattern value="\s*[\s0]?2\s*[\s0]?[0-9A-Fa-f]\s*[\s0]?[0-
9ABab]\s*[\s0]?[0-9]\s*[\s0]?[0-9]\s*[\s0]?[0-9Aa]\s*0?[0-9ABab]\s*[\s0]?[0-9A-
Fa-f]\s*[\s0]?[0-9A-Da-d]\s*[\s0]?[0-7]\s*"/>
  </xsd:restriction>
</xsd:simpleType>
```

The interpretation of the numerical values of Panose-1 in Windows GDI is different from genuine Panose-1 definition. The definition in above restricts the scope to the values that genuine Panose-1 definition and Windows GDI interpretation are consistent.

Editor's Response:

2009-03-24 Prague meeting:

Can we still continue to use the xsd:hexBinary type and use the proposed regular expression?

2010-07-28 Chris Rae:

This is an interesting one – the only open issue on it is whether an item of type xsd:hexBinary can have a pattern-type RegExp restriction on it.

I think the answer to this is "yes" (going from our previous discussions around date types) – is that correct? If so, I think we can close this DR.

2010-07-28 Makoto Murata:

Yes, it can. See http://www.w3.org/TR/xmlschema-2/#hexBinary

2010-07-29 Makoto Murata:

Long time ago, Shawn said that the current regexp rejects some legal values. Suzuki-san requested an example. Since then, nothing has happened.

2010-08-25 Chris Rae:

I did a bit of investigation into this and it looks like almost all valid values are actually excluded by the current regexp. The regexp is:

\s*[\s0]?2\s*[\s0]?[0-9A-Fa-f]\s*[\s0]?[0-9ABab]\s*[\s0]?[0-9]\s*[\s0]?[0-9]\s*[\s0]?[0-9Aa]\s*0?[0-9ABab]\s*[\s0]?[0-9A-Fa-f]\s*[\s0]?[0-9A-Da-d]\s*[\s0]?[0-7]\s*

Some example values that are rejected: 1, 5, and 50.

Murata-san, will this be enough to go back to Suzuki-san with?

Changes to Part 1: N Part 2: N Part 3: N Part 4: N

DR 09-0062 — WML, Settings: Semantics of short file names

Status: Closed; in COR1

Subject: WML, Settings: Semantics of short file names

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00052, 08-00053

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.15.2, "Web Page Settings", p. 1219 Part 1, §17.15.2.13, "doNotUseLongFileNames", p. 1237Part 1, §17.15.2.46, "webSettings (Web Page Settings)", p. 1288

Related DR(s): none

Nature of the Defect:

The doNotUseLongFileNames element specifies that applications should ensure that all file names generated when this document is subsequently saved as a web page do not exceed the 8.3 character file name limitation.

Also it should be clarified if the file names are case sensitive or not.

Solution Proposed by the Submitter:

This feature is supposed to be introduced for the short file names on FAT file system; therefore, "8.3 characters" should be 8.3 octets.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Adopted the proposed solution and added text saying that short names are not case-sensitive. Closed.

The solution to this DR should be published in a Technical Corrigendum.

The exact changes are as follows:

Part 1, §17.15.2, "Web Page Settings", p. 1219

[*Example*: Consider the following WordprocessingML fragment for the web page settings in a WordprocessingML document:

…

The webSettings element contains all of the web page settings for this document. In this case, the web page settings specified for this document are: a frameset defined using the frameset element (§xx); and a setting specifying that when this file is saved as a web page, all resulting files must not exceed 8.3 ~~characters~~octets in length using the doNotUseLongFileNames element (§xx). *end example*]

Part 1, §17.15.2.13, "doNotUseLongFileNames", p. 1237

This element specifies that applications shall ensure that the file names for all files generated when saving this document as a web page do not exceed eight ~~characters~~octets with a three ~~character~~octet extension. This includes all supporting files (images which are part of this HTML web page, etc.). The file names generated are not case-sensitive.

[*Example*: Consider a WordprocessingML document which contains the following content within the web settings part:

…

The doNotUseLongFileNames element specifies that applications should ensure that all file names generated when this document is subsequently saved as a web page do not exceed the 8.3 ~~character~~octet file name limitation. *end example*]

Part 1, §17.15.2.46, "webSettings (Web Page Settings)", p. 1288

[*Example*: Consider the following WordprocessingML fragment for the web page settings in a WordprocessingML document:

…

The webSettings element contains all of the web page settings for this document. In this case, the web page settings specified for this document are: a frameset defined using the frameset element (§xx); and a setting specifying that when this file is saved as a web page, all resulting files must not exceed 8.3 ~~characters~~octets in length using the doNotUseLongFileNames element (§xx). *end example*]

2009-12-04 Paris meeting:

Add the following change in response to P1-GB-0008:

Replaced “8.3 characters in length” with “8 octets with 3 octet extension” in §17.15.2 and §17.15.2.46.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0063 — WML, Fields: Form Field Properties length

Status: Closed; in COR1

Subject: WML, Fields: Form Field Properties length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00055

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.16.17, "ffData (Form Field Properties)", p. 1412

Related DR(s): none

Nature of the Defect:

"… a maximum character length of 10 characters via the maxLength element"

If the length is counted by the octet instead of the Unicode character, it should be stated, because the length is quite short.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Still not sure of the units.

2009-06-09 Shawn Villaron:

I'm hoping that one of the XML experts on WG4 can help me determine the right approach for handling a set of open defect reports.

The following defect reports relate to the lengths of buffers for some of our entities:

DR 09-0063 — WML, Fields: Form Field Properties length

Part 1, §17.16.17, "ffData (Form Field Properties)", p. 1412

DR 09-0064 — WML, Simple Types: ST_FFHelpTextVal length

Part 1, §17.18.25, "ST_FFHelpTextVal", p. 1534

DR 09-0065 — WML, Simple Types: ST_FFName length

Part 1, §17.18.26, ST_FFName, p. 1535

DR 09-0066 — WML, Simple Types: ST_MacroName length

Part 1, §17.18.51, "ST_MacroName", p. 1563

DR 09-0068 — SML, Styles: name attribute length

Part 1, §18.8.29, "name (Font Name)", p. 1965

DR 09-0069 — SML, Pivot Tables: longText attribute length

Part 1, §18.10.1.90, "sharedItems (Shared Items)", p. 2164

We've currently defined each of these buffers in terms of number of characters. My notes say that there is WG4 interest in understanding the lengths of these in octets. But that's where I'm getting confused.

Since XML specifies a character encoding, shouldn't **characters** be the right unit of measurement here? I've verified that the buffers defined are correct, so if we're just waiting for verification, we can write this up and close them out; if we need to use a different unit of measurement, if you can help me figure out what the right unit of measurement is, I can get them written up tomorrow.

To be clear, I don't have a strong opinion here; rather, I'd prefer to tap into the XML expertise on WG4 to inform me on the best approach.

As you can imagine, I'm super interested in getting proposals to all of these out ASAP. Any help here would be greatly appreciated.

2009-06-10 Mohamed Zergaoui:

My understanding is the opposite. It looks like JISC was afraid that the lengths were in octets.

So the way to solve this is to make the word "character" point to a definition of "unicode character" in Terms and Definitions.

2009-06-10 Shawn Villaron:

Interesting; that makes me wonder if our resolution to DR-09-0070 is incorrect …

But, this does seem to make more sense from an XML perspective, so unless someone disagrees, I think we should go with using “unicode characters” as the metric.

I’d prefer to take the DRs I listed earlier, add DR-09-0070 to the list, and update their prose so that it clearly states Unicode characters. I looked at changing the Terms and Definitions (adding a definition for character, but with 2600+ instances of “character”, we could introduce more problems).

2009-06-10 Shawn Villaron:

Part 1, §17.16.17, “ffData (Form Field Properties)”, p. 1413:

The ffData element specifies the set of properties for this text box form field; in this example, a form field name of TestTextBox via the name element (§17.16.27), a disabled state via the enabled element (§17.16.14), and a maximum character length of 10 Unicode characters via the maxLength element (§17.16.26). *end example*]

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct “character” term to be used.

2009-06-20 Makoto Murata:

Here are some entries in Appendix G (Glossary) of the Unicode 5.0.0 standard.

++

Character. (1) The smallest component of written language that has semantic value; refers to the abstract meaning and/or shape, rather than a specific shape (see also glyph), though in code tables some form of visual representation is essential for the reader’s understanding. (2) Synonym for abstract character. (3) The basic unit of encoding for the Unicode character encoding. (4) The English name for the ideographic written elements of Chinese origin. [See ideograph (2).]

Abstract Character. A unit of information used for the organization, control, or representation of textual data. (See definition D7 in Section 3.4, Characters and Encoding.)

Code Point. Any value in the Unicode codespace; that is, the range of integers from 0 to 10FFFF16. (See definition D10 in Section 3.4, Characters and Encoding.)

Code Position. Synonym for code point. Used in ISO character encoding standards.

Code Unit. The minimal bit combination that can represent a unit of encoded text for processing or interchange. The Unicode Standard uses 8-bit code units in the UTF-8 encoding form, 16-bit code units in the UTF-16 encoding form, and 32-bit code units in the UTF-32 encoding form. (See definition D77 in Section 3.9, Unicode Encoding Forms.)

Code Value. Obsolete synonym for code unit.

Byte. (1) The minimal unit of addressable storage for a particular computer architecture.

(2) An octet. Note that many early computer architectures used bytes larger than 8 bits in size, but the industry has now standardized almost uniformly on 8-bit bytes. The Unicode Standard follows the current industry practice in equating the term byte with octet and using the more familiar term byte in all contexts. (See octet.)

Octet. An ordered sequence of eight bits considered as a unit. The Unicode Standard follows current industry practice in referring to an octet as a byte. (See byte.)

Unicode Scalar Value. Any Unicode code point except high-surrogate and low-surrogate code points. In other words, the ranges of integers 0 to D7FF16 and

E00016 to 10FFFF16,

inclusive. (See definition D76 in Section 3.9, Unicode Encoding Forms.)

++

I now think that "Unicode scalar value" is the right term *if* U+101D0 PHAISTOS DISC SIGN PEDESTRIAN, for example, is a single something.

DR 09-0070 should not be affected by this discussion, since DR 09-0070 is concerned about the representation given by UTF16LE. Meanwhile, other DRs do not choose and fix particular encodings.

2009-06-22/24 Copenhagen meeting:

Agreed to use the Unicode 5 term “Unicode Scalar Value”, as proposed in Murata-san’s mail of 2009-06-20. Closed.

2009-06-25 Mohamed Zergaoui:

+1 for Unicode Scalar Value

2009-06-25 Rick Jelliffe:

I would prefer "character (Unicode scalar value)" because character is a well-known term. I agree it is useful to be clear here, but raw "Unicode scalar value" will be irritating.

2009-06-25 Makoto Murata:

I do not think that a character is a Unicode scalar value, so I do not support your suggestion. But I agree that "Unicode scalar value" is irritating even for us. It would be nice if we can put a note for non-normatively explaining this term. Are you willing to provide a proposal? If lucky, WG4 can agree to incorporate in the DCOR in the upcoming phone conferences. If not, I will make sure that Japan will make a comment on this term and propose the note.

2009-06-25 Keld Jørn Simonsen:

I think the proper term in ISO is UCS rather than Unicode.

I also think there is a precise term for this in IS 10646.

I think the proper term in ISO is UCS rather than Unicode.

2009-06-25 Makoto Murata:

This issue has been discussed and resolved. It is now too late to consider such a big change. Moreover, XML depends on Unicode already. I see no reasons not to use Unicode terms.

> I also think there is a precise term for this in IS 10646.

I strongly doubt that.

The actual changes are:

Part 1, §17.16.17, "ffData (Form Field Properties)", p. 1413

[*Example*: Consider the following WordprocessingML fragment for a text box form field:

…

The ffData element specifies the set of properties for this text box form field; in this example, a form field name of TestTextBox via the name element (§17.16.27), a disabled state via the enabled element (§17.16.14), and a maximum character length of 10 Unicode scalar values~~characters~~ via the maxLength element (§17.16.26). *end example*]

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0064 — WML, Simple Types: ST_FFHelpTextVal length

Status: Closed; in COR1

Subject: WML, Simple Types: ST_FFHelpTextVal length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00056

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.18.25, "ST_FFHelpTextVal", p. 1534

Related DR(s): none

Nature of the Defect:

This simple type's contents have a maximum length of 256 characters.

If the length is counted by the octet instead of the Unicode character, it should be stated, because the length is rather short for a help text.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Still not sure of the units.

2009-06-10 Shawn Villaron:

Part 1, §17.18.25, "ST_FFHelpTextVal (Help Text Value)", p. 1534:

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 256 Unicode characters.

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct "character" term to be used.

2009-06-22/24 Copenhagen meeting:

Agreed to use the Unicode 5 term "Unicode Scalar Value", as proposed in Murata-san's mail of 2009-06-20. Closed.

The actual changes are:

Part 1, §17.18.25, "ST_FFHelpTextVal", p. 1534

This simple type also specifies the following restrictions:

This simple type's contents have a maximum length of 256 Unicode scalar values~~characters~~.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0065 — WML, Simple Types: ST_FFName length

Status: Closed; in COR1

Subject: WML, Simple Types: ST_FFName length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00057

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.18.26, ST_FFName, p. 1535

Related DR(s): none

Nature of the Defect:

This simple type's contents have a maximum length of 65 characters.

If the length is counted by the octet instead of the Unicode character, it should be stated, because the length is rather short.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Still not sure of the units.

2009-06-10 Shawn Villaron:

Part 1, §17.18.26, "ST_FFName (Form Field Name Value)", p. 1535:

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 65 Unicode characters.

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct "character" term to be used.

2009-06-22/24 Copenhagen meeting:

Agreed to use the Unicode 5 term "Unicode Scalar Value", as proposed in Murata-san's mail of 2009-06-20. Closed.

The actual changes are:

Part 1, §17.18.26, ST_FFName, p. 1535

This simple type also specifies the following restrictions:

This simple type's contents have a maximum length of 65 Unicode scalar values~~characters~~.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0066 — WML, Simple Types: ST_MacroName length

Status: Closed; in COR1

Subject: WML, Simple Types: ST_MacroName length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00058

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §17.18.51, "ST_MacroName", p. 1563

Related DR(s): none

Nature of the Defect:

This simple type's contents have a maximum length of 33 characters.

If the length is counted by the octet instead of the Unicode character, it should be stated, because the length is rather short.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Still not sure of the units.

2009-06-10 Shawn Villaron:

Part 1, §17.18.51, "ST_MacroName (Script Subroutine Name Value)", p. 1563:

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 33 Unicode characters.

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct "character" term to be used.

2009-06-22/24 Copenhagen meeting:

Agreed to use the Unicode 5 term "Unicode Scalar Value", as proposed in Murata-san's mail of 2009-06-20. Closed.

The actual changes are:

Part 1, §17.18.51, "ST_MacroName", p. 1563

This simple type also specifies the following restrictions:

This simple type's contents have a maximum length of 33 Unicode scalar values~~characters~~.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0067 — SML, Workbook: Semantics of short file names

Status: Closed; in COR1

Subject: SML, Workbook: Semantics of short file names

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00059

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §18.2.24, "webPublishing", p. 1736

Related DR(s): none

Nature of the Defect:

longFileNames (Enable Long File Names) attribute: Specifies a boolean value that indicates whether the application allows file names longer than 8 characters for Web pages.

Also the case sensitive or not should be clarified.

Solution Proposed by the Submitter:

This feature is supposed to be introduced for the short file names on FAT file system; therefore, "8.3 characters" should be 8.3 octets.

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

Adopted the proposed solution and added text saying that short names are not case-sensitive. Closed.

The solution to this DR should be published in a Technical Corrigendum.

The exact changes are as follows:

Part 1, §18.2.24, "webPublishing", p. 1736

Attributes	Description
longFileNames (Enable Long File Names)	Specifies a boolean value that indicates whether the application allows file names longer than 8 ~~characters~~octets for Web pages. File names are not case-sensitive. The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

2009-12-04 Paris meeting:

Add the following change in response to P1-GB-0009:

Add "with a three octet extension" after "8 octets".

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0068 — SML, Styles: name attribute length

Status: Closed; in COR1

Subject: SML, Styles: name attribute length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00060

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §18.8.29, "name (Font Name)", p. 1965

Related DR(s): none

Nature of the Defect:

"The string length for this attribute shall be 0 to 31 characters."

If the length is counted by the octet instead of the Unicode character, it should be stated, because the length is quite short.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

62 octets.

2009-06-10 Shawn Villaron:

Here is my proposed response for this defect report.

Part 1, §18.8.29, "name (Font Name)"

Attributes	Description
val (String Value)	A string representing the name of the font. If the font doesn't exist (because it isn't installed on the system), or the charset not supported by that font, then another font should be substituted. The string length for this attribute shall be 0 to 31 Unicode characters. The possible values for this attribute are defined by the ST_Xstring simple type (§22.9.2.19).

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct "character" term to be used.

2009-06-22/24 Copenhagen meeting:

Agreed to use the Unicode 5 term "Unicode Scalar Value", as proposed in Murata-san's mail of 2009-06-20. Closed.

The actual changes are:

Part 1, §18.8.29, "name (Font Name)", p. 1965

Attributes	Description
val (String Value)	… The string length for this attribute shall be 0 to 31 Unicode scalar values~~characters~~. …

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0069 — SML, Pivot Tables: longText attribute length

Status: Closed; in COR1

Subject: SML, Pivot Tables: longText attribute length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00061

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §18.10.1.90, "sharedItems (Shared Items)", p. 2164

Related DR(s): none

Nature of the Defect:

"longText (Long Text) attribute: A string is considered long if it is over 255 characters. A value of 1 or true indicates the value contains more than 255 characters of text. A value of 0 or false indicates the value contains less than 255 characters. [Note: This is used as many legacy spreadsheet application support a limit of 255 characters for text values. end note]

If the length is counted by the octet instead of the Unicode character, it should be stated. Also it should be stated whether the assumed legacy spreadsheet applications can handle Unicode text.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-03-24 Prague meeting:

510 octets

2009-06-10 Shawn Villaron:

Part 1, §18.10.1.90, "sharedItems (Shared Items)", p. :

Attributes	Description
longText (Long Text)	Specifies a boolean value that indicates whether this field contains a long text value. A string is considered long if it is over 255 Unicode characters. A value of `1` or `true` indicates the value contains more than 255 Unicode characters of text. A value of `0` or `false` indicates the value contains less than 255 Unicode characters. [*Note*: This is used as many legacy spreadsheet application support a limit of 255 Unicode characters for text values. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct "character" term to be used.

2009-06-22/24 Copenhagen meeting:

Agreed to use the Unicode 5 term "Unicode Scalar Value", as proposed in Murata-san's mail of 2009-06-20. Closed.

The exact changes are:

Part 1, §18.10.1.90, "sharedItems (Shared Items)", p. 2167:

Attributes	Description

Attributes	Description
longText (Long Text)	Specifies a boolean value that indicates whether this field contains a long text value. A string is considered long if it is over 255 Unicode scalar values~~characters~~. A value of `1` or `true` indicates the value contains more than 255 Unicode scalar values~~characters~~ of text. A value of `0` or `false` indicates the value contains less than 255 Unicode scalar values~~characters~~. [*Note*: This is used as many legacy spreadsheet application support a limit of Unicode scalar values~~characters~~ for text values. *end note*] The possible values for this attribute are defined by the W3C XML Schema boolean datatype.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N

DR 09-0070 — PML, Presentation: modifyVerifier password length

Status: Closed; in COR1

Subject: PML, Presentation: modifyVerifier password length

Qualifier: Request for clarification

Submitter: JISC **Organization:** (JP)

Contact Information: eb2m-mrt@asahi-net.or.jp

Submitter's Defect Number: 08-00062

Supporting Document(s): none

Date Circulated by Secretariat: 2009-01-28

Deadline for Response from Editor: 2009-03-28

IS 29500 Reference(s): Part 1, §19.2.1.19, "modifyVerifier (Modification Verifier)", p. 2774

Related DR(s): none

Nature of the Defect:

"The password supplied to the algorithm is to be a UTF-16LE encoded string; strings longer than 255 characters are truncated to 255 characters. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation."

If the surrogate pair is counted as 2 characters, it should be stated.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Editor's Response:

2009-02-16 Shawn Villaron:

This issue deals with some ambiguity associated with the length of the supplied password in PresentationML. The current text states the following:

> The password supplied to the algorithm is to be a UTF-16LE encoded string; strings longer than 255 characters are truncated to 255 characters. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

The question relates to the meaning of “longer than 255 characters.” I believe that the intent of this limitation is to support up to 510 bytes.

I’m curious as to how everyone thinks we should rephrase things. Do we think that this updated text would suffice addressing this issue:

> The password supplied to the algorithm is to be a UTF-16LE encoded string; strings longer than 510 bytes are truncated to 510 bytes. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

Do we think we need more information here?

2009-02-16 MURATA Makoto:

I support the idea of replacing "255 characters" with "510 bytes". Although "255 characters" are actually correct, some people will think that a surrogate pair is a single character and thus misinterpret this sentence.

2009-02-17 Jesper Lund Stocholm:

I agree. The sentence is quite specific in terms of "UTF16LE"-encoding and using the (layman's) term "characters" is a bit confusing. Using 510 /bytes/ would be in line with the "level" of the rest of the paragraph.

2009-02-17 MURATA Makoto:

Oops, we are a part of ISO. We should use "octet" rather than "byte", since a byte is not always 8 bits. ISO/IEC 10646 uses "octet".

2009-02-18 Shawn Villaron:

Thanks. I was wondering if we should add an additional constraint regarding the number of octets. For example, what do we think about this text:

The password supplied to the algorithm is to be a UTF-16LE encoded string; strings longer than 510 octets are truncated to 510 octets. Further, the number of octets should be evenly divisible by the size of the octets of the character in the target octet set. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

Does this add any value?

2009-03-24 Prague meeting:

Adopted the proposed solution. Closed.

The solution to this DR should be published in a Technical Corrigendum.

2009-06-11 Teleconference:

The following DRs were discussed as a group: 09-0063, 09-0064, 09-0065, 09-0066, 09-0068, and 09-0069.

After a short discussion regarding bytes, octets, characters, Unicode characters, and Unicode code points, it was agreed to defer this discussion. A detailed off-line investigation of this is needed. *Depending on the outcome of this, we may need to revisit our response to DR 09-0070.*

Action: Shawn, Murata-san, Rex, and Mohamed will look at the correct "character" term to be used.

2009-06-22/24 Copenhagen meeting:

Previously, this DR was kept open just in case the resolution to DR's 09-0063, 64, 65, 66, 68, and 69 affected the resolution of DR 09-0070. As that resolution does not have any impact, we agreed to move this one to Closed.

The exact changes are:

Part 1, §19.2.1.19, "modifyVerifier (Modification Verifier)", p. 2774

The password supplied to the algorithm is to be a UTF-16LE encoded string; strings longer than 510 octets~~255 characters~~ are truncated to 510 octets~~255 characters~~. If there is a leading BOM character (U+FEFF) in the encoded password it is removed before hash calculation.

Changes to Part 1: Y Part 2: N Part 3: N Part 4: N