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): initially 29500:2011, then 29500:2012 Part 1, §19.2.1.13, “font (Embedded Font Name)”

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:

Yes

**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 §19.7, “PML Simple Types”, and then using that simple type in the definition of pitchFamily, as follows:

**Part 1, §19.2.1.13, “font (Embedded Font Name)”, pp. 2778–2779**

|  |  |
| --- | --- |
| 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:

|  |  |
| --- | --- |
| ~~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 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~~ST\_PitchFamily simple type (§19.7.56). |

**Part 1, §19.7.xx, “ST\_PitchFamily (Pitch Family)”, p. 3006, new subclause**

This simple type specifies a font pitch.

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

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

|  |  |
| --- | --- |
| Enumeration 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 |

[*Note*: The W3C XML Schema definition of this simple type’s content model (ST\_PitchFamily) is located in §A.3. *end note*]

**Part 1, §A.3, “PresentationML”, p. 4538, lines 1523–1535**

<xsd:simpleType name="ST\_ViewType">

 …

</xsd:simpleType>

<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>

<xsd:complexType name="CT\_NormalViewPortion">

**Part 1, §B.3, “PresentationML”, p. 4844, lines 1038–1047**

p\_ST\_ViewType =

 …

| "sldThumbnailView"

<<yet to come>>

p\_CT\_NormalViewPortion =

**2010-09-06/08 Tokyo Meeting:**

Agreed with proposal. Moved to “Last Call”.

**2010-11-25 Rex Jaeschke:**

The resolution for this DR was very much underspecified w.r.t the details although the intent is clear.

Below is my take at the details. While I think I'm on the right track, I have some questions and need some input from you (Chris) to complete this DR's write up. Please check thoroughly my edits.

1. You proposed the new simple type be called ST\_pitchFamily; however all other such types appear to start with an uppercase letter, so I've "corrected" that.

2. I've taken a shot as defining the narrative for the new simple type, but if you could supply an opening sentence that would be great.

3. I put "Pitch Family" inside the parens on the ST subclause header, which seemed the obvious thing to do.

4. It seems to me that the values of the new simple type should be moved to the simple type definition, and I have done that; however, in (all?) other cases of enumerated types with value lists, each enumeration has had a name.

But the values 00-82 aren't really names, are they? Is the table move correct, and if so, is it correct? I don't think so.

**2010-11-28 Makoto Murata:**

While trying to store this schema change to the SVN repository, I find some problems.

First, the proposal introduces a new type without referencing it from the attribute definition.

Second, why should this simple type definition appear in pml.xsd when this attribute occurs in dml-main.xsd as part of a complex type CT\_TextFont?

Third, if we introduce this simple type to dml-main.xsd and change the def of CT\_TextFont, will some use of this complex type be destroyed?

**2010-12-03 Chris Rae:**

I've made a couple of further changes (integrated above).

**2011-02-17 Makoto Murata: (in private mail)**

I am afraid that we really screwed up when we closed this DR in WG4.

First, I committed the required schema changes. They are parts of

<https://www.assembla.com/code/IS29500/subversion/nodes/branches/Part1DCOR2?rev=133>

and

<https://www.assembla.com/code/IS29500/subversion/nodes/branches/Part4DCOR2?rev=133>

If you know how to use subversion, this repository is extremely useful. My work required for schema maintenance is now extremely easier than before. The DCOR1 and FPDAM1 were really painful, but I now enjoy schema hacking for OOXML.

If you don't, here are the changes I did.

1) dml-main.rnc (for both Parts 1 and 4)

a\_ST\_PitchFamily =

xsd:byte "00" | xsd:byte "01" | xsd:byte "02" | xsd:byte "16" | xsd:byte "17" | xsd:byte "18" | xsd:byte "32" | xsd:byte "33" | xsd:byte "34" | xsd:byte "48" | xsd:byte "49" | xsd:byte "50" | xsd:byte "64" | xsd:byte "65" | xsd:byte "66" | xsd:byte "80" | xsd:byte "81" | xsd:byte "82"

is added immediately before the def of a\_CT\_TextFont

 attribute pitchFamily { xsd:byte }?,

in a\_CT\_TextFont is replaced by

 attribute pitchFamily { a\_ST\_PitchFamily }?

2) dml-main.xsd (both Parts 1 and 4)

 <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>

is added immediately before the def of CT\_TextFont

 <xsd:attribute name="pitchFamily" type="xsd:byte" use="optional" default="0"/>

in CT\_TextFont is replaced by

 <xsd:attribute name="pitchFamily" type="ST\_PitchFamily" use="optional" default="0"/>

In the current DR log, “ST\_PitchFamily (Pitch Family)” is mistakenly added to 19.7.56. However, since it is part of dml-main.xsd and dml-main.rnc, it should be added to 20.1.10.

Accordingly, "Note: The W3C XML Schema definition of this simple type’s content model (ST\_PitchFamily) is located in §A.3. end note]" has to be revised by replace A.3 by A.4.1.

BTW, pitchFamily appears in many places, namely

19.2.1.13 font (Embedded Font Name)

21.1.2.3.1 cs (Complex Script Font)

21.1.2.3.3 ea (East Asian Font)

21.1.2.3.7 latin (Latin Font)

21.1.2.3.10 sym (Symbol Font)

21.1.2.4.6 buFont (Specified)

I think that we have to change all occurrences of the attribute pitchFamily in these places in the same manner, but the current wording changes 19.2.1.13 only.

**2011-03-28/30 Prague Meeting:**

We reviewed Murata-san’s mail of 2011-02-17.

**2011-05-16 Chris Rae:**

Murata-san, the edits look good; I think you're quite right about using §20.1.10 instead of §19.7.56. I also think it would be sensible to change all of the other pitchFamily instances as well.

If I understand this correctly these are schema-only changes?

**2011-05-23 Makoto Murata:**

I think that we have to change the description of @pitchFamily for the other subclauses (see below) as well. Since these subclauses rely on a single complex type CT\_TextFont, we only have to change it.

§21.1.2.3.1 cs (Complex Script Font)

§21.1.2.3.3 ea (East Asian Font)

§21.1.2.3.7 latin (Latin Font)

§21.1.2.3.10 sym (Symbol Font)

§21.1.2.4.6 buFont (Specified)

**2011-05-26 Teleconference:**

Chris reported that he agrees with Murata-san’s suggestion of making a broader change. Chris is currently working on the final resolution for this.

**2011-06-03 Chris Rae:**

I have removed the schema changes from my document, and added the modifications to each of the other sections Murata-san mentioned. The existing schema changes (revision 133) are not quite complete, as they don't utilize the new simple type in the extra subclauses. I'm going to see if I can check in the correct changes before Berlin.

The new version of the required changes to the standard is below.

**Part 1, §19.2.1.13, “font (Embedded Font Name)”, pp. 2777–2778, attribute pitchFamily**

| **Attributes** | **Description** |
| --- | --- |
| pitchFamily (Similar Font Family)Namespace: http://purl.oclc.org/ooxml/drawingml/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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.87)~~W3C XML Schema byte datatype~~. |

**Part 1, §20.1.10.87, “ST\_PitchFamily (Pitch Family)”, new subclause**

This simple type specifies a font pitch.

[*Note:* Although the type name is ST\_PitchFamily, the integer value of this attribute specifies the font family with the higher 4 bits nad the font pitch with the lower 4 bits. *end note*]

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

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

| **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 |

[Note: The W3C XML Schema definition of this simple type’s content model ([ST\_PitchFamily](#xsd_s_3dbeeb50-6fdd-490e-86c9-b7a5f36004)) is located in §A.3. end note]

**Part 1, §21.1.2.3.1, “cs (Complex Script Font)”, pp. 3607–3608, attribute pitchFamily**

| **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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.87)~~W3C XML Schema byte datatype~~. |

**Part 1, §21.1.2.3.3, “ea (East Asian Font)”, pp. 3615–3616, attribute pitchFamily**

| **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 a byte variable this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.87)~~W3C XML Schema byte datatype~~. |

**Part 1, §21.1.2.3.7, “latin (Latin Font)”, pp. 3623–3624, attribute pitchFamily**

| **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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.87)~~W3C XML Schema byte datatype~~. |

**Part 1, §21.1.2.3.10, “sym (Symbol Font)”, pp. 3632–3633, attribute pitchFamily**

| **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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.87)~~W3C XML Schema byte datatype~~. |

**Part 1, §21.1.2.4.6, “buFont (Specified)”, pp. 3647–3648, attribute pitchFamily**

| **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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.87)~~W3C XML Schema byte datatype~~. |

**2011-06-20/22 Berlin Meeting:**

There was general agreement with the proposal; however, when Rex asked if his (supposedly editorial) questions from 2010-11-25 had been factored into the current proposal, it appeared they had not.

Regarding Rex's comments, Chris reported that they have all been taken into consideration by the latest proposal. Regarding having names for the enumerations, this is one of those instances where it's just a list of values rather than enumerated names. While it might have been a better idea to have names when it was designed, Chris proposed that we not change that now as it'll break existing documents and the benefit of changing it does not outweigh the cost.

Move to Closed in COR2.

**Part 1, §A.4.1, “DrawingML – Main”, p. 4593, lines 2807–2815**

<xsd:simpleType name="ST\_TextTypeface">

 <xsd:restriction base="xsd:string"/>

</xsd:simpleType>

<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>

<xsd:complexType name="CT\_TextFont">

 <xsd:attribute name="typeface" type="ST\_TextTypeface" use="required"/>

 <xsd:attribute name="panose" type="s:ST\_Panose" use="optional"/>

 <xsd:attribute name="pitchFamily" type="~~xsd:byte~~ST\_PitchFamily" use="optional" default="0"/>

 <xsd:attribute name="charset" type="xsd:byte" use="optional" default="1"/>

</xsd:complexType>

**Part 1, §B.4.1, “DrawingML – Main”, p. 4890, lines 2149–2160**

a\_ST\_TextFontSize =

xsd:int { minInclusive = "100" maxInclusive = "400000" }

a\_ST\_TextTypeface = xsd:string

a\_ST\_PitchFamily =

 xsd:byte "00" | xsd:byte "01" | xsd:byte "02" | xsd:byte "16" |

 xsd:byte "17" | xsd:byte "18" | xsd:byte "32" | xsd:byte "33" |

 xsd:byte "34" | xsd:byte "48" | xsd:byte "49" | xsd:byte "50" |

 xsd:byte "64" | xsd:byte "65" | xsd:byte "66" | xsd:byte "80" |

 xsd:byte "81" | xsd:byte "82"

a\_CT\_TextFont =

attribute typeface { a\_ST\_TextTypeface },

attribute panose { s\_ST\_Panose }?,

## default value: 0

attribute pitchFamily { ~~xsd:byte~~a\_ST\_PitchFamily }?,

## default value: 1

attribute charset { xsd:byte }?

**Part 4, §A.4.1, “DrawingML – Main”, p. 1145, lines 2842–2850**

<xsd:simpleType name="ST\_TextTypeface">

 <xsd:restriction base="xsd:string"/>

</xsd:simpleType>

<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>

<xsd:complexType name="CT\_TextFont">

 <xsd:attribute name="typeface" type="ST\_TextTypeface" use="required"/>

 <xsd:attribute name="panose" type="s:ST\_Panose" use="optional"/>

 <xsd:attribute name="pitchFamily" type="~~xsd:byte~~ST\_PitchFamily" use="optional" default="0"/>

 <xsd:attribute name="charset" type="xsd:byte" use="optional" default="1"/>

</xsd:complexType>

**Part 4, §B.4.1, “DrawingML – Main”, p. 1476, lines 2162–2173**

a\_ST\_TextFontSize =

xsd:int { minInclusive = "100" maxInclusive = "400000" }

a\_ST\_TextTypeface = xsd:string

a\_ST\_PitchFamily =

 xsd:byte "00" | xsd:byte "01" | xsd:byte "02" | xsd:byte "16" |

 xsd:byte "17" | xsd:byte "18" | xsd:byte "32" | xsd:byte "33" |

 xsd:byte "34" | xsd:byte "48" | xsd:byte "49" | xsd:byte "50" |

 xsd:byte "64" | xsd:byte "65" | xsd:byte "66" | xsd:byte "80" |

 xsd:byte "81" | xsd:byte "82"

a\_CT\_TextFont =

attribute typeface { a\_ST\_TextTypeface },

attribute panose { s\_ST\_Panose }?,

## default value: 0

attribute pitchFamily { ~~xsd:byte~~a\_ST\_PitchFamily }?,

## default value: 1

attribute charset { xsd:byte }?

**2015-09-21/24 Beijing Meeting:**

Re-opened based on issues found when proofing DCOR3B.

From John’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-09:

After digging into this, it seems DR 09-0037 may have been resolved incorrectly – implementations do write out those attributes.  The correct resolution seems to be the other option from that old e-mail.  Which would leave the attributes in CT\_TextFont (technically, re-add them) and DR 09-0055 can be applied as we planned and approved.  Shall I file a DR to change 09-0037, assuming we can incorporate that (and 09-0055) into the pending COR3b?

From Murata-san’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-09:

I don't think that we need a new DR.  We only have to reopen DR 09-0037 and add a new solution.  John, could you propose a solution?

FYI: Suzuki-san is aware of the issues, and is going to study them.

From John’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-10:

I'd probably suggest the first of the two options Shawn suggested back then - the one that wasn't chosen.

From Suzuki-san’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-14:

DR 09-0037:

The embeddedFont is an interface element to the embedded font; it has the attributes "charset", "panose" and "pitchFamily". But the font referrer in DrawngML (for example) can specify them independently.

Please think about a document with an embedded font; named "Arial", charset=ANSI, panose=020b0604020202020204, pitchFamily=Swiss/Variable.

If DrawingML tries to use a font named "Arial" but charset=Symbol && pitchFamily=Roman/Fixed, how the implementation should handle the request?

The implementation should take the requested font is not the embedded font even if the family name is same?

Or, as the referrer's attributes are described as the attributes for fallback, the family name matching is already sufficient to use the embedded "Arial"?

Considering that existing implementation write these attributes, I have no objection to these attributes in the specification as far as "how the implementation should use the values" is clarified.

DR 09-0055:

As far as I check the discussion, there is no rationale to keep current loose type definition for ST\_PitchFamily. I suggest to use clearer type defined by Murata-sensei.

From John’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-18:

DR 09-0037:

To ensure I'm looking at the same markup you are, do you mean there is potential confusion/conflict between a "font" element under the "embeddedFont" element in the presentation part and a font element ("latin", "ea", "cs", "sym") under a run properties element ("rPr", "endParaRPr") in a slide part?

Presentation part:

<p:embeddedFont>

 <p:font typeface="Abadi MT Condensed" panose="020B0506030101010103" pitchFamily="34" charset="0"/>

 <p:regular r:id="rId5"/>

</p:embeddedFont>

Slide part:

<a:rPr lang="en-US" dirty="0" smtClean="0">

 <a:latin typeface="Abadi MT Condensed" panose="020B0506030101010103" pitchFamily="34" charset="0"/> </a:rPr>

My initial thought is the same as your conclusion - the font in the rPr is not the same as the embedded font. But since the description for the charset and pitchFamily attributes of the latin/ea/cs/sym elements says those values may be used for font substitution, you raise a good question about whether they should be considered when determining whether the rPr font is the same as the embedded font. I will request an interpretation from the developers here.

From Suzuki-san’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-18: (attachments omitted)

Thank you for comment, just I made a preliminary testing PPTX.

a) a PPTX made by PowerPoint 2010, Title uses Century Gothic, Text uses Century

b) a PPTX with a "broken" embeddedFontLst, whose panose/pitchFamily/charset are exchanged;

\* "Century Gothic" entry keeps its name and its reference to embedded font stream, but its panose/pitchFamily/charset are replaced by those in original "Century" entry.

\* "Century" entry keeps its name and its reference to embedded font stream, but its panose/pitchFamily/charset are replaced by those in original "Century Gothic" entry.

c) a PPTX with "broken" font referrers, whose familyname "Century Gothic" and "Century" in slideMaster1.xml and slideLayout2.xml are exchanged.

Opening them by PowerPoint 2010, the results I see are:

b) looks same with a)

c) looks differently, uses "Century" on the title, and "Century Gothic" on the text.

According to this result, the font familyname matching seems to be sufficient to identify an embedded font resource, and the different panose/pitchFamily might not have serious impact.

Oh, my test is still insufficient; Century and Century Gothic have same "charset". I will do yet another experiment.

From Suzuki-san’s email thread, “COR3 issue: ST\_PitchFamily” from 2015-09-18: (attachments omitted)

> Oh, my test is still insufficient; Century and Century Gothic have

> same "charset". I will do yet another experiment.

The results of my preliminary experiments for mismatched charset seem to be complicated than mismatched panose/pitchFamily:

-----------------------------------------------------------------

Experiment 1)

charsets of all referring parts are changed from "0" (ANSI) to "177" (Hebrew), but charsets in embeddedfontlist are left as "0".

--> a dialog is shown "PowerPoint found a problem with content in <filename>, PowerPoint can attempt to repair the presentation"

-----------------------------------------------------------------

Experiment 2)

charsets of all referring parts and embeddedfontlist are changed from "0" (ANSI) to "177" (Hebrew).

--> a dialog is shown "PowerPoint found a problem with content in <filename>, PowerPoint can attempt to repair the presentation"

-----------------------------------------------------------------

There is a possibility that mismatched-charset is seriously handled than mismatched-panose and mismatched-pitchFamily.

In addition, there is a possibility that existing implementation checks the mismatch between charset declared by embeddedFontLst and the embedded binary data.

I think a note to encourage the consistency of charsets among the referring XML, referred XML and embedded font data would be safer, if it can cause visible warning in the existing implementation.

John produced a proposal, which he presented Thursday afternoon. There was general agreement.

Here are the proposed changes (adapted from John’s document “DR-09-0055 changes v4.docx”):

**Part 1: §19.2.1.13, “font (Embedded Font Name)”, attribute pitchFamily, p. 2,523**

*[This edit assumes that the edits to this subclause in DR 09-0037 have already been applied.]*

|  |  |
| --- | --- |
| Attributes | Description |
| pitchFamily (Similar Font Family)Namespace: http://purl.oclc.org/ooxml/drawingml/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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.xx)~~W3C XML Schema byte datatype~~. |

**Part 1, §20.1.10.xx, “ST\_PitchFamily (Pitch Family)”, new subclause, p. xx**

**20.1.10.xx ST\_PitchFamily (Pitch Family)**

This simple type specifies a font pitch.

[Note*:* Although the type name is ST\_PitchFamily, the integer value of this attribute specifies the font family with the higher 4 bits and the font pitch with the lower 4 bits. end note]

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

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

| **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 |

[Note: The W3C XML Schema definition of this simple type’s content model ([ST\_PitchFamily](#xsd_s_3dbeeb50-6fdd-490e-86c9-b7a5f36004)) is located in §A.3. end note]

**Part 1: §21.1.2.3.1, “cs (Complex Script Font)”, attribute pitchFamily, p. 3,220**

|  |  |
| --- | --- |
| 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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.xx)~~W3C XML Schema byte datatype~~. |

**Part 1, §21.1.2.3.3, “ea (East Asian Font)”, attribute pitchFamily, pp. 3,227–3,228**

|  |  |
| --- | --- |
| 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 a byte variable this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.xx)~~W3C XML Schema byte datatype~~. |

**Part 1: §21.1.2.3.7, “latin (Latin Font)”, attribute pitchFamily, pp. 3,234–3,235**

|  |  |
| --- | --- |
| 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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.xx)~~W3C XML Schema byte datatype~~. |

**Part 1: §21.1.2.3.10, “sym (Symbol Font)”, attribute pitchFamily, pp. 3,242–3,243**

|  |  |
| --- | --- |
| 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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.xx)~~W3C XML Schema byte datatype~~. |

**Part 1: §21.1.2.4.6, “buFont (Specified)”, attribute pitchFamily, pp. 3,254–3,255**

|  |  |
| --- | --- |
| 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 this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (§20.1.10.xx)~~W3C XML Schema byte datatype~~. |

**Part 4, §16.5.3, “Changed attribute for font element (Part 1, §19.2.1.13)”, pp. 239–240**

|  |  |
| --- | --- |
| 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 a byte variable this value shall be interpreted as follows:~~

| **~~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 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 ST\_PitchFamily simple type (Part 1, §20.1.10.xx)~~W3C XML Schema byte datatype~~. |

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