DR 19-0012 — Part 4: Unprefixed and incorrectly-prefixed names in code fragments in examples

Status: Open

Subject: Part 4: Unprefixed and incorrectly-prefixed names in code fragments in examples

Qualifier: Editorial defect

Submitter: Francis Cave Organization: (BSI)

Contact Information: francis@franciscave.com

Submitter’s Defect Number: 20190225-1

Supporting Document(s): none

Date Circulated by Secretariat: 2019-02-26

Deadline for Response from Editor: 2019-04-26

IS 29500 Reference(s): Part 4, §5, §19

Related DR(s): DR 17-0010, DR 19-0006

Nature of the Defect:

Many code examples in Part 4 contain unprefixed element (tag) names. A review of two implementations of OOXML (MS Office and LibreOffice) suggests that, with the exception of documents where the root element is in the SpreadsheetML namespace\*, OOXML documents use prefixed names.

\* Names in the namespace “http://purl.oclc.org/ooxml/spreadsheetml/main” are unprefixed in these implementations. Names the Extended Properties and Custom Properties namespaces (see Part 1 §22.2 and §22.3) are also unprefixed in known implementations.

For consistency, names in examples that are in namespaces listed in Annex C, other those in the SpreadsheetML namespace, should be prefixed names. A Note is needed in §5 Notation conventions to explain the convention of using prefixed element names in examples, except in the case of SpreadsheetML examples.

The use of prefixed attribute names in WordprocessingML, where the attribute name is in the same namespace as its parent element, is poor XML practice. However, in the other MLs OOXML generally follows XML practice by only using prefixed names for attributes that are in a specific namespace that is different to the namespace of the parent element name. Unprefixed attribute names are in no namespace (see §6.2 of Namespaces in XML 1.0). Part 4 contains some unprefixed attribute names in examples that are in specified namespaces. To be schema-valid these names must be prefixed.

In §19.2, in attribute descriptions of elements using content model CT\_StrokeChild, the examples are inappropriate, in that the code fragments are copied from §19.1 without any modification to suit the local context. In many cases (due to copy-and-paste errors) the entire attribute description is inappropriate, not only the code fragment in the example. Given that these attribute descriptions require extensive revision, it might be sensible to group these in a single attribute group, as has already been done in §19.1 (see DR 17-0010), to avoid having to make similar corrections in five separate places.

In examples in §19.3 the use of the prefix ‘wd’ for names in the VML WordprocessingML Drawing namespace is inconsistent with Annex C, and the prefix ‘w10’ should be used.

In investigating problems with prefixes, a number of other typos in examples have been found. These are detailed in the solutions proposed.

[*Note*: I am indebted to Murata-san for finding many of the errors reported in this DR. *end note*]

Solution Proposed by the Submitter:

**In §5 add at the bottom:**

[Note: Tag names in code fragments are expressed as prefixed names whenever the name is in any of the namespaces listed in Annex C where a qualifying prefix is specified. The main exception is that tag names in SpreadsheetML examples are prefixless (in §15). Attribute names in code fragments are prefixless unless the name is declared to be in a specific namespace that is different to that of the parent element’s tag name, except in WordprocessingML examples in which all attribute names are prefixed. end note]

**In §19.1.4.10:**

In the attribute descriptions for the attributes dgmlayout, dgmlayoutmru and dgmmodekind change the code fragments in the examples respectively as follows:

… o:dgmlayout="1">

...

… o:dgmlayoutmru="1">

...

… o:dgmnodekind="1">

**In each of §19.1.2.5 fill, §19.1.2.11 imagedata and §19.1.2.21 stroke:**

In the description of the attribute althref change respectively the code fragment in the example as follows:

<v:fill … o:althref="myimage.pcz" … >

</v:fill>

...

<v:imagedata … o:althref="myimage.pcz" … >

</v:imagedata>

...

<v:stroke … o:althref="myimage.pcz" … >

</v:stroke>

**In §19.1.2.2 background:**

In the description of the attribute bwnormal change the code fragment in the example as follows:

… o:bwmode="auto" o:bwnormal="lightgrayscale" … >

**In §19.1.2.2 background:**

In the description of the attribute bwpure change the code fragment in the example as follows:

… o:bwmode="auto" o:bwpure="highcontrast" … >

**In §19.1.2.14 path:**

In the description of the attribute extrusionok change the first code fragment in the example as follows:

<v:rect fillcolor="lime"

style="width:50;height:50">

<vo:extrusion on="true"/>

<v:path o:extrusionok="false"/>

</v:rect>

**In §19.1.2.19 shape:**

In the description of the attribute extrusionok change the second code fragment in the example as follows:

<v:shape style="position:absolute;top:50;left:20;width:50;height:50"

fillcolor="yellow" path="m 0,0 l 0,1000 1000,1000 1000,0 x e">

<vo:extrusion on="true" lightposition="0,-2000,10000"/>

</v:shape>

**Request VML experts to provide suitable, contextually-specific replacement text and examples for descriptions of the following attributes:**

Used in §19.2.2.1 bottom, §19.2.2.6 column, §19.2.2.16 left, §19.2.2.26 right and §19.2.2.32 top:

* althref
* color
* color2
* dashstyle
* endarrow
* endarrowlength
* endarrowwidth
* endcap
* filltype
* forcedash
* href
* imagealignshape
* imageaspect
* imagesize
* insetpen
* joinstyle
* linestyle
* miterlimit
* on
* opacity
* src
* startarrow
* startarrowlength
* startarrowwidth
* title

Consider creating an attribute group for the all attributes used in CT\_StrokeChild, so that they can be documented in prose in one place rather than five.

The code fragment in each case should include the element of the sub-section in question as well as the attribute being described.

**In §19.2.2.10:**

Change the code fragment as follows:

<v:shape>

…

<o:equationXxml contentType="mathml">

<mrow xmlns="…">

…

</mrow>

</o:equationXxml>

</v:shape>

**In §19.2.2.20 OLEObject:**

Cange the code fragment in the first example as follows:

<o:OLEObject r:id="rb1" ProgId="OAFIID:Bonobo\_Sample\_Calculator">

…

</o:OLEObject>

**For §19.2.2.20 OLEObject:**

Request VML experts to provide an example of the use of attribute ObjectID that actually demonstrates the use of the attribute.

**In §19.3 and all sub-sections in all code fragments:**

Change ‘wd:’ to ‘w10:’. For example, in the code fragment in the example in §19.3 as follows:

<v:shape … >

…

<wd10:wrap wd10:type="topAndBottom" />

</v:shape>

**In §19.4.2.43 MultiLine:**

Change the code fragment in the example as follows:

<x:ClientData> …

<x:MultilLine>True</x:MultilLine>

</x:ClientData>

Schema Change(s) Needed:

Yes/No

**Editor’s Response:**

None

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