DR 16-0016 — SML: Named Styles

Status: Further Consideration Required

Subject: SML: Named Styles

Qualifier: Request for clarification

Submitter: Charlie Clark Organization: Ecma/Clark Consulting & Research

Contact Information: [charlie.clark@clark-consulting.eu](file:///E:\Std\SC%2034\WG4\DRs\Log\2014\charlie.clark@clark-consulting.eu)

Submitter’s Defect Number: None

Supporting Document(s): None

Date Circulated by Secretariat: 2016-08-08

Deadline for Response from Editor: 2016-10-08

IS 29500 Reference(s): 29500:2016, Part 1, §18.8

Related DR(s): None

Nature of the Defect:

I've recently received an SML file that looks like it was created in LibreOffice and I'm having trouble working out how to handle the named styles these are defined as:

<cellStyles count="6">

<cellStyle name="Normal" xfId="0" builtinId="0" customBuiltin="false"/>

<cellStyle name="Comma" xfId="15" builtinId="3" customBuiltin="false"/>

<cellStyle name="Comma [0]" xfId="16" builtinId="6" customBuiltin="false"/>

<cellStyle name="Currency" xfId="17" builtinId="4" customBuiltin="false"/>

<cellStyle name="Currency [0]" xfId="18" builtinId="7" customBuiltin="false"/>

<cellStyle name="Percent" xfId="19" builtinId="5" customBuiltin="false"/>

</cellStyles>

So, there are only six named styles, but there are 20 "master style records". Is this kind of discrepancy intended? What about styles with indices of 1–14?

As with any kind of relationships between elements, it's impossible to write a constraint in the schema, but I understand the relationship between the two should be 1:1. This is important in any situation where you try and combine key-based referencing with positional and any kind of deduplication.

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

No

**Editor’s Response:**

**2016-08-08 Francis Cave:**

This DR relates to DR 16-0007. There are a number of other elements which have the attribute @count that, if either specified or taking its default value, can conflict with the content of the element. There are a large number of elements (>50 in Part 1) that have this attribute. I will check whether conflicts can arise in each of these cases.

**2016-08-09 Charlie Clark:**

I don't think this is the same issue. The problem with DR 16-006 is how to handle what are essentially denormalised relations such as between CellStyle and CellStyleXfs such as the following:

<cellStyles >

<cellStyle name="20% - Accent1" xfId="3" builtinId="30"/>

<cellStyle name="Heading 1" xfId="2" builtinId="16"/>

<cellStyle name="Normal" xfId="0" builtinId="0"/>

<cellStyle name="Title" xfId="1" builtinId="15"/> </cellStyles>

<cellStyleXfs>

<xf numFmtId="0" fontId="0" fillId="0" borderId="0"/>

<xf numFmtId="0" fontId="2" fillId="0" borderId="0" applyNumberFormat="0"

applyFill="0" applyBorder="0" applyAlignment="0" applyProtection="0"/> </cellStyleXfs>

As things stand I think this would be valid XML but likely to cause problems for applications.

There should be a constraint such that len(//cellsStyle/\*) = len(//cellStyleXfs/\*) I'm not so hot on XSL but maybe there should be relevant ref attributes in the schema?

<xsd:complexType name="CT\_CellStyle" ref="xf">

<xsd:sequence>

<xsd:element name="extLst" type="CT\_ExtensionList" minOccurs="0"

maxOccurs="1"/>

</xsd:sequence>

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

<xsd:attribute name="xfId" type="ST\_CellStyleXfId" use="required"/>

<xsd:attribute name="builtinId" type="xsd:unsignedInt" use="optional"/>

<xsd:attribute name="iLevel" type="xsd:unsignedInt" use="optional"/>

<xsd:attribute name="hidden" type="xsd:boolean" use="optional"/>

<xsd:attribute name="customBuiltin" type="xsd:boolean" use="optional"/> </xsd:complexType>

Unfortunately, a constraint the other way is not possible because the xf type is used in different contexts with different elements (I've submitted a DR in the past along these lines): a base type would help here.

**2016-09-26/29 Seoul F2F Meeting:**

After some discussion, Francis withdrew his comment of 2016-08-08, as it doesn’t help address the issue.

**2017-01-26 Aarti Nankani:**

Response from product team: CellStyles and CellStyleXfs need not be 1:1. All that’s required is that every CellStyle has an xfId connecting it to a CellStyleXf.  The example Charlie provides that show a CellStyle specifying a non-existent xfId is indeed schema-valid, but it violates the standard description of what xfId value is.  The requirement is clear even though it’s not schema enforced.

**2017-02-15 Charlie Clark:**

I don't think this really helps implementers. Elements in CellStyleXfs are referenced by cellStyle elements and xf elements where they refer to the same object. What other references are there to a CellStyleXf can exist?

Making the 1:1 relationship explicit cellStyle contains the name, cellStyleXf the formatting along with a diagram similar to those used in for r:Id attributes would make this section a lot easier to understand especially for anyone working with large files. I can understand that backfitting a constraint to avoid duplicates is not possible for compatibility reasons but at the very least a note that cellStyle@name should be unique should be added. I think this was agreed to elsewhere but I can't find the issue at the moment.

**2017-03-02 Seattle Meeting:**

We agreed with the comment of 2017-01-26: CellStyles and CellStyleXfs need not be 1:1, and we will change the spec to make this clear. Rex will write up this proposal, and then we’ll close this DR.

We note that the resolution to DR 14-0015 adds the sentence: “Individual cellStyle elements contained within a cellStyles element shall contain unique values for their name attributes.”

**2017-04-07 Rex Jaeschke:**

**Part 1: §18.8.8, “cellStyles (Cell Styles)”, p. xx**

This element contains the named cell styles, consisting of a sequence of named style records. A named cell style is a collection of direct or themed formatting (e.g., cell border, cell fill, and font type/size/style) grouped together into a single named style, and can be applied to a cell.

[Example: For example, "Normal", "Heading 1", "Title", and "20% - Accent1" are named cell styles expressed below. They have builtInId's associated with them, and use xfId to reference the specific formatting elements pertaining to the particular style. The xfId is a zero-based index, referencing an xf record in the cellStyleXfs collection.

<cellStyles count="4">

<cellStyle name="20% - Accent1" xfId="3" builtinId="30"/>

<cellStyle name="Heading 1" xfId="2" builtinId="16"/>

<cellStyle name="Normal" xfId="0" builtinId="0"/>

<cellStyle name="Title" xfId="1" builtinId="15"/>

</cellStyles>

end example]

CellStyles and CellStyleXfs need not have a 1:1 relationship; all that’s required is that every CellStyle has an xfId connecting it to a CellStyleXf.

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