DR 14-0015 — SML: Cell Styles

Status: Further Consideration Required

Subject: SML: Cell 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: 2014-12-11

Deadline for Response from Editor: 2015-02-11

IS 29500 Reference(s): 29500:2016, Part 1, §18.8.8, “cellStyles (Cell Styles)”, p. 1761

Related DR(s): None

Nature of the Defect:

§18.8.8 cellStyles (Cell Styles) defines named styles’

Do these relate 1:1 to the styles defined in §18.8.9 cellStyleXfs (Formatting Records)?

Are there any constraints on the names for cellStyles? I notice that Excel enforces unique names for user-defined styles but I've seen files where styles are repeated with the same name, example enclosed. I'm not sure how to interpret such styles.

Looking at the definitions, and assuming a 1:1 relationship between CellStyle and CellStyleXfs, it looks that the same name is always is associated with the same definition even if the Ids differ. If this is true, and I've no idea how these styles can otherwise be referred to except by their name, a reduction to three styles should be possible. I'm not sure if it's possible to enforce a unique constraint on the attribute of a sequence but that would surely be desirable.

The enclosed file (see below) validates but according to the definition of the customBuiltin attribute:

“By default, built-in styles are not persisted when not in use. This flag indicates that a built-in style has been modified, and therefore should be saved with the workbook, even if not currently in use."

As these styles aren’t in use I would expect an application to remove them but that doesn't happen with Excel, at least. Should that be the behaviour for built-ins?



Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

No

**Editor’s Response:**

**2015-02-24/26 Seattle F2F Meeting:**

Chris will work on a proposal.

**2016-02-29/03-02 Barcelona Meeting:**

Response from Darrin’s team:

Add a new Implementer note. Some of the specific questions around when excel does/doesn’t remove unnecessary style specifications from the file are out-of-scope. The in-scope question here is regarding Excel’s requirements on the names of user-defined styles. Excel does require unique names, we should add a note for this.

Charlie wrote:

Thanks

Feedback from Caroline:

I had assumed that the feedback from the Excel team did mean that the requirement for unique names for user-defined styles should be added to the standard in a note.  However, if it really is a requirement, then it would be normative and therefore does not belong in a note.  There is no mention of user-defined styles in clause 18 for SML.  I think the requirement should probably go in the third paragraph of 18.8.7.  --- or in its own paragraph before the third para.

The Description for customBuiltin does indeed say, "By default built-in styles are not persisted when not in use."  If Excel does not do that as Charlie claims, then that statement should be removed or modified to suggest that applications can choose not to persist un-used built-in styles.

I agree with the Excel team that modifying the schema simply to be more elegant is out of scope.  I am also not sure that there is any reason to assume a 1-1 relationship between cellStyle and cellStyleXfs.  As long as each named style in cellStyles has a corresponding <xf> in cellStyleXfs, there is no harm in extra <xf> entries and could be benefits.

**2016-04-01 Charlie Clark:**

Here’s more information for this bug report, which I think helps clarify at least part of the issue that implementers face.

The following is part of a styles.xml that is perfectly valid:

<cellStyleXfs count="5">

<xf borderId="0" fillId="4" fontId="1" numFmtId="0"/>

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

<xf borderId="0" fillId="3" fontId="1" numFmtId="0"/>

<xf borderId="0" fillId="5" fontId="1" numFmtId="0"/>

<xf borderId="0" fillId="2" fontId="1" numFmtId="0"/>

</cellStyleXfs>

<cellXfs count="1">

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

</cellXfs>

<cellStyles count="5">

<cellStyle name="Custom 3" xfId="0"/>

<cellStyle builtinId="0" name="Normal" xfId="1"/>

<cellStyle name="Custom 1" xfId="2"/>

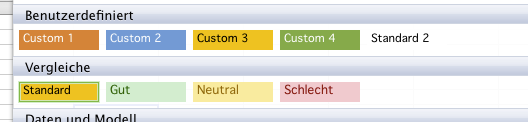
<cellStyle name="Custom 4" xfId="3"/>

<cellStyle name="Custom 2" xfId="4"/>

</cellStyles>

However, this has interesting effects when actually used (see screenshot below) in an Excel file. From Excel's behaviour I deduce the following:

* the order of named styles is determined by the order of the styles in the cellStyleXfs part
* the first style in the cellStyleXfs part is privileged and always refers to the default style for cells.



I would argue that both of these effects are another reason why there should not be any duplicates within the cellStyles part and I would suggest that, at the very least, the implementer notes be extended to cover this and suggest that the styles in cellStyles keep the same order as in cellStyleXfs even if it is not strictly necessary.

Here are the relevant bits from MS’s implementer notes: “(b) The standard states that both the cell style xf records and cell xf records must be read to understand the full set of formatting applied to a cell. In Office, only the cell xf record defines the formatting applied to a cell.”

Can we confirm that this is still the case even in standards compliant mode? This would sort of answer my question about the order of application: irrelevant because you must always write an individual cell style. But this would mean that the documentation should really be adjusted and that "commutative" styles are not possible, though this would be a real pity.

**2016-09-20 Rex Jaeschke:**

MS’s experts report, “We agree with Caroline’s recommendation to update the standard to resolve this.”

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

Darrin will forward this to his experts.

**2016-12-07 Rex Jaeschke:**

MS’s experts suggest adding the following: “Individual cellStyle elements contained within the cellStyles shall contain unique values for the name attribute”.

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