DR 16-0009 — Shared ML: AppVersion Element

Status: Closed Without Action

Subject: Shared ML: AppVersion Element

Qualifier: Technical defect

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

Contact Information: [charlie.clark@clark-consulting.eu](file:///E%3A%5CStd%5CSC%2034%5CWG4%5CDRs%5CLog%5C2014%5Ccharlie.clark%40clark-consulting.eu)

Submitter’s Defect Number: None

Supporting Document(s): None

Date Circulated by Secretariat: 2016-04-11

Deadline for Response from Editor: 2016-06-11

IS 29500 Reference(s): 29500:2016, Part 1, §22.2.2.2, “AppVersion (Application Version)”. p. 3,722

Related DR(s): None

Nature of the Defect:

This issue is minor, but also very frustrating. The version element of a document's properties is a string. However, there is a restriction: “The content of this element shall be of the form XX.YYYY where X and Y represent numerical values, or the document shall be considered non-conformant.”

Either the schema should be updated to reflect this constraint ([0-9]?\.[0-9]? as a regex pattern, I think) , or it should be made a recommendation, or it should be removed as it is an unnecessarily restriction on versioning that hinders interoperability: it is often very helpful when working with files from another source to know exactly which version of which application or library produced the file. When using the common major.minor.patch scheme, Microsoft Excel will complain the file is broken, offer to repair it, but provide no report as to what is apparently broken. This really is something that Microsoft should resolve: either by being less fussy, or advising that the part is incorrect (though the validator won't flag it as it conforms to the schema).

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

No

**Editor’s Response:**

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

MS experts replied: We don’t think that enforcing the standard at a schema level is the right approach here. We offer no opinion on removing the restriction completely, as that decision lies with the standards body. As for how the app handles enforcement of the standard, that comes down to app-specific implementation of behavior. Our feeling is that changing our enforcement would result in Excel being overly prohibitive on non-conformant AppVersion. Overall, we propose no change though.

**2016-12-07 Charlie Clark:**

Then the resolution must be to remove the sentence, as requested because it indicates a normative definition that is not the case: "XX.YYYY" is **not** required by the specification. And Excel, in the spirit of interoperability, needs to be more forgiving, or at least more informative with other versioning schemes.

**2016-12-07 Teleconference:**

This might be a bad design, but we don’t consider it to be a defect, but rather a request for an extension. Closed without action.

**2016-12-14 Charlie Clark:**

I'm slightly confused by this. The suggestion was to remove an unrealistic normative assertion from the narrative description of the specification. I don't see how this can be interpreted as an extension. The important thing is that the descriptive and formal parts of the standard are in agreement wherever possible and here they most definitely are not.

The solutions I proposed:

1. update the schema to match the description
2. mark the current normative aspect of the description as exemplary
3. remove the normative aspect of the description as it is misleading

Can you explain how any of these can be considered as an extension?

**2016-12-14 Francis Cave:**

I suggest that we discuss this again on the next telecon. I may have misunderstood the argument.

**2016-12-14 Murata-san:**

Thank you very much for your contributions to WG4.  I hope that you continue to submit DRs even if our dispositions do not always make you happy.

First, I think that the schema is unnecessarily loose here.  Ideally, the schema should have been as tight as the prose restriction. (In general, schemas cannot capture all restrictions in prose.  But in this particular case, the schema can.)

Nevertheless, I do not think that this is a defect.   If a programmer read both the prose and schema, the only reasonable interpretation is that versions are of the form XX.YYYY.  I do not think that this (arguably poor) design choice will make two conformant implementations fail to interwork.  Fixing a poor design choice is an extension.  Not a defect correction.

**2016-12-14 Charlie Clark:**

I'm trying to engage in the spirit of an open specification. Thank you for taking the time to reply, especially most recently regarding the packaging issue. And thanks to you Rex, for tirelessly forwarding my e-mails.

> First, I think that the schema is unnecessarily loose here. Ideally,

> the schema should have been as tight as the prose restriction.

> (In general, schemas cannot capture all restrictions in prose. But in

> this particular case, the schema can.)

I agree that this should be the case wherever possible.

> Nevertheless, I do not think that this is a defect. If a programmer

> read both the prose and schema, the only reasonable interpretation is

> that versions are of the form XX.YYYY. I do not think that this

> (arguably poor) design choice will make two conformant implementations

> fail to interwork. Fixing a poor design choice is an extension. Not

> a defect correction.

Thanks for the clarification but I also proposed an alternative that would not require an extension: remove the restriction in the prose. Quoting MS on this:

"Our feeling is that changing our enforcement would result in Excel being overly prohibitive on non-conformant AppVersion"

This suggests that current practice allows for schema compliant numbering such as 6.7.5. But unfortunately, files like that are considered corrupt.

So, interoperability is already hindered due to an arbitrary preference for a versioning scheme. Microsoft already indicates that they don't with to be "overly prohibitive".

**2016-12-14 Murata-san:**

Removing this restriction in the prose is beyond the scope of technical corrigenda, since it clearly allows more documents as conformant.  In other words, even if there is an unreasonable restriction, lifting it is beyond the scope of technical corrigenda.  This requires an amendment.  The voting procedure for technical corrigenda is less heavier than that for amendments.

WG4 can discuss whether we need an amendment for removing this restriction.

**2016-12-14 Francis Cave:**

I do not see any new reasons to reopen this DR, but

I agree to consider possible removal of the restriction in prose as a request for an amendment rather than a COR.  Lifting existing restrictions is beyond the scope of CORs.

**2016-12-14 Charlie Clark:**

Ah, I think I understand. Yes, this would be a material change even if it is essentially a clarification.

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