DR 16-0009 — Shared ML: AppVersion Element

Status: Open

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.

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