



## **Ecma's Proposed Disposition of Comments on OOXML: How we got here; What is missing; Why you should vote No**

### **I. Background – How We Got Here**

In early 2007 Ecma International (Ecma) submitted the Microsoft-authored document format specification for Office Open XML (OOXML) to ISO/IEC JTC1 for approval as an International Standard. This provocative submission occurred only three months after JTC1 published OpenDocument Format as a unanimously-approved International Standard (ISO/IEC 26300:2006).

ODF is the current ISO standard for XML-based word processing, spreadsheet and presentation documents. By using an open standard format like ODF, consumers avoid vendor lock-in and are able to have a choice of suppliers. ODF is widely supported by many vendors in both commercially-available and open-source software, and is seeing strong adoption world wide<sup>1</sup>.

OOXML has been widely criticized as a flawed standard, having been designed with only a single vendor's objectives in mind and to work fully only with that vendor's products. OOXML was submitted to JTC1 in an immature state, hastily written, insufficiently reviewed and full of errors<sup>2</sup>.

#### ***A. JTC1 September Vote - OOXML not approved as an International Standard***

On September 2, 2007, DIS 29500 Office Open XML failed to receive a sufficient number of favorable votes from national bodies (NBs) of ISO, IEC and JTC1 to be approved as an International Standard. Further, the balloting was tainted by many documented irregularities. Some numbers to consider:

- 5 months – the time NBs had to review the specification in the “fast track” process
- 6,045 pages – the length of specification to be reviewed
- 3,522 comments – the number of errors, ambiguities and omissions submitted by NBs

#### ***B. Disposition of Comments – These are just Draft Proposals***

The JTC1 Fast-Track procedure gives a proposer of a failed standard the opportunity to address the reported problems in hopes of persuading those NBs with comments to change their vote to approval. The proposer, Ecma in this case, writes a proposed Disposition of Comments report (“the proposal”)

<sup>1</sup> See <http://www.odfalliance.org/resources/Adoptions20Dec2007.pdf> for a list of over a dozen nations with ODF policies and a list of applications support.

<sup>2</sup> See <http://www.odfalliance.org/resources/TheCaseAgainstOOXML.pdf> for a more details on technical concerns with OOXML.

in which resolutions for each NB comment from the September 2 ballot are suggested. Note some rules:

- The proposals are non-binding. Ecma does not have the power to change OOXML, since the proposed standard is under JTC1 control.
- NB can submit proposals to resolve their own (or other NB) comments. Many NBs have done so already in proposals that accompanied their ballot comments.
- No changes are actually made until approved at the Ballot Resolution Meeting (BRM)<sup>3</sup>. NBs should understand that it is just a draft that has not been formally submitted to JTC1 yet.

## II. The Disposition of Comments – What is Missing from the Proposal

Below is a brief evaluation of Ecma's proposed Disposition of Comments with some technical details on seven key deficiencies. In short, the proposal does NOT address the critical need for: a.) review time; b.) harmonization, c.) a clear name; d.) a sound standard with no (new or old) technical errors; e.) interoperability; f.) support for legacy documents; and g.) consistency of “fixes.”

### A. What is Missing? Time. Legitimate defects acknowledged by Ecma, why rush?

Because this review was rushed (as shared above), we believe that the number of undetected and unreported flaws exceeds those reported with the ballot. **Ecma, in its proposed Disposition of Comments, has acknowledged the legitimacy and seriousness of the reported defects** and in many cases they have proposed changes to the proposal to fix these errors. However, a look at the numbers again reveals that it is unwise to rush the review and approval of these fixes.

- Over 1,000 changes – total number of “fixes” proposed by Ecma
- 2,293 pages – length of response explaining these “fixes.” Many of the responses are mere rough outlines of what should be done and not full solutions.
- 200 misses – number of “fixes” (at least 20 percent of those submitted) where Ecma missed the opportunity to address serious concerns (e.g., harmonization, interoperability) by providing an inadequate response or introducing additional errors (discussed further below).

The rushed preparation of the standard has already resulted in quality problems which prevented approval in September. Further rushing will only serve to make matters worse.

### B. What is Missing? Harmonization. Why not move toward one standard?

As Tim Bray (co-editor of the W3C XML standard) pointed out in 2005, “The world does not need two ways to say, ‘This paragraph is in 12-point Arial with 1.2em leading and ragged-right justification.’” Yet, the Proposal promotes “translation” and denies the ability to add harmonizing features, which as

---

<sup>3</sup> For more on the BRM and next steps, see Annex A or the BRM Fact Sheet - [http://www.odfalliance.org/resources/BRM%20Voting%20Guide\\_final.pdf](http://www.odfalliance.org/resources/BRM%20Voting%20Guide_final.pdf)

explained below, is not acceptable.

**Translation is a poor substitute for having a single format.** If the translation as proposed by Ecma can be done perfectly, then the necessity of two formats is questionable. If translation introduces errors, it is clearly inferior to having vendors work to harmonize their formats.

Harmonization starts from looking at where the two formats overlap – and there is a significant, perhaps 90 percent or more, area where OOXML and ODF do overlap – and expressing this functional overlap identically. This common functionality between ODF and OOXML would also include a common extensibility mechanism. The remaining 10 percent of the functionality, where these standards do not overlap, would represent the focus of the harmonization effort. That portion of it which represents a widespread need could be brought into the core of ODF. That remaining portion which only serves one vendor's needs, such as flags for deprecated legacy formatting options, could be represented using the common extensibility mechanism.

**The Proposal refuses to add ODF features to OOXML, but Microsoft has acknowledged that OOXML can be built on top of ODF.** The Proposal rejects the reasonable request (see for example DK-004 and DK-005) to add specific features from ISO ODF to OOXML, which is doable. Further, the Proposal is wrong because there is no reason why, by a harmonization process, all of the functionality of Microsoft Office cannot be represented on a base of ISO ODF. In fact, Microsoft acknowledges this much in a recent statement by Gray Knowlton, Group Product Manager for Microsoft Office, who is quoted in *PC Word*<sup>4</sup>:

Also, if individual governments mandate the use of ODF instead of Open XML, Microsoft would adapt, Knowlton said. The company would then implement the missing functionality that ODF doesn't support.

If Ecma does not wish to add functionality to OOXML, then this path – adding features of OOXML to ODF or expressing the requirements of Microsoft Office on top of ODF – is the only path to harmonization. And, there is agreement that this approach is technically feasible. Knowlton goes on to say, in the same article, “However, those extensions would be custom-designed and outside of the standard, which is counter to the idea of an open document standard.”

Further agreed is that it is preferable to extend ODF within the standards process, not via custom, non-open extensions. So let's do it!<sup>5</sup> Moreover, let's start harmonization by avoiding further divergence.

### ***C. What is Missing? A Clear Name. Why deny obvious confusion?***

The confusion between the trademarked name of open source software, OpenOffice, and this

---

<sup>4</sup> <http://www.pcworld.com/article/id,141424-c.opensource/article.html>

<sup>5</sup> The OASIS ODF Technical Committee Chairs have said that they are ready and willing to host such a harmonization effort, or to participate in a JTC1/SC34 harmonization effort, if that is preferred.

proposed standard, OOXML, or Office Open XML, is not helpful for consumers. **As evidenced by Microsoft webpages, their own enterprise architects and the media, this confusion between “open office” and “office open” is persistent and widespread.** Therefore, we disagree with the assertion (see response to CA-003, for example) that no name change is needed.

For example, Microsoft hosts a report on “Document formats and the value of choice in healthcare” on their web site<sup>6</sup>. This report argues in favor of “Open Office XML”, making this error five different times. Further, in the last few days, we have seen the following examples of confusion (*italics added*):

- January 11 – “Version 6.1 of Office Mobile finally brings support for Microsoft’s *Open Office XML* document formats” (Cossacks.org.uk reports)
- January 14 – “It also has recommended schools stick with the Open Document Format for storing files, rather than Microsoft’s *Open Office XML*” (WindowsWorldNews.com report)
- January 15 -- “Microsoft’s *Open Office XML* and its competitor OpenDocument Format have always been rivals in the field of document formats.” (Pravda, in its lead sentence)
- January 15 -- “Microsoft introduced a new file format in Microsoft Office 2007 called the *Open Office XML* Format...” (eHacks blog announcing availability of a DocX convertor)
- January 15 – “The Commission investigators will see if Microsoft has withheld information needed for interoperability from rival companies. They will also study the *Open Office XML* file format to see if Microsoft is releasing technologies that in effect, reduce compatibility.” (HighPosition.net article on recent EC investigation)
- January 16 -- “Document Standards : MS *Open Office XML* versus ODF” (Philippe Destoop, Microsoft’s Enterprise Architect for Belgium and Luxembourg, titles his blog entry)
- January 17 -- “Ecma *Open Office XML* Comments Completed Today” (Owen Allen, a Microsoft “technical specialist” reports in his blog the headline)

#### ***D. What is Missing? A technically sound standard. Why introduce new errors?***

We are still reviewing the 2,293 pages of proposed changes to this 6,045 page specification, but from what has been analyzed to date, approximately 25 percent of Ecma's proposed fixes are flawed in that they are inadequate, erroneous or produce additional problems. We have highlighted a few examples below.

#### **Dating dilemma complicated – five different date representations and default settings still flawed.**

Objections to the way in which OOXML handles spreadsheet dates were raised by 27 National Bodies. OOXML does not allow dates before 1900 and it requires that the year 1900 be (incorrectly) treated as a leap year. Rather than fix the erroneous leap year calculation, Ecma's proposal (Response 43) complicates the problem in three ways.

One, the Proposal effectively provides five date representations. This undesirable and unnecessary solution entails four different dating schemes below plus the representation of a date as a string in ISO 8601 format.

---

<sup>6</sup> <http://www.microsoft.com/uk/nhs/content/articles/document-formats-and-the-value-of-choice-in-healthcare.aspx>

- Dates from 0000-9999, where January 1st , 1900 has numeric value of 0
- Dates from 0000-9999, but where January 1st, 1904 is given value of 0
- Dates from 1900-9999, where where January 1st , 1900 has numeric value of 0
- Dates from 1904-9999, where where January 1st , 1904 has numeric value of 0

Two, the proposal does not explain how to treat dates before the Gregorian era. The bottom of page 280 reads, “In the 1900 date-base system, the lower limit is January 1, 0000, which has a serial value -693959.” This clause fails to declare how dates are treated before the start of the Gregorian era (1582). For example, should the proleptic Gregorian calendar be used, which extends Gregorian leap year calculations backwards in time? Or should the Julian calendar be used?

Three, the Proposal is inconsistent as it says, “The use of the 1900 backward compatibility or 1904 backward compatibility date-base system should be avoided” but still the schema declares these to be the default options. The default settings should represent the preferred, not the avoided, settings.

To make this real let's suppose you are an astronomer calculating the number of days or years between an meteoric event in Roman times and now. If you have a date from Roman times, how do you enter it into a spreadsheet, and what value does the spreadsheet record and use for its calculations? If you use the wrong calendar, you will very likely be sitting up all night watching the sky on the wrong night, and possibly the wrong year.

**Document security and integrity at risk due to inadequate definitions and circumventing established definitions. We cite two examples here looking at macros and language strings.**

First, four National Bodies raised issues about the “MACROBUTTON” feature which allows a word processor field to present a button which will execute a macro when activated. Ecma's Response 110 declares that the location of any macro code or command will not be defined by the standard but rather is “implementation-defined”. While for some functions, e.g., runtime API, this is okay, it is unacceptable for two reasons to not define the location of the macro code:

- Security - mail gateway and virus scanners need the mechanism to locate, scan and possibly eliminate some of these active code elements.
- Integrity - when dividing or merging documents, the mechanisms to locate macros is critical in order for the macro code to be correctly associated with the revised documents, i.e., one can not correctly create the new document unless one can locate and assemble all document elements, including macros.

A better proposal would be to use the robust mechanism that OOXML already has for specifying parts and their relations. For example, an ID could be stored specifying the relationship which contains the macro code.

Second, as indicated by NB comments such as Australia's AU-0023, Ecma has failed to remove features which introduce invalid characters into XML. XML itself has a clearly defined lexical and

value space, and the XML strings are interpreted relative to this scheme. Characters outside of this value space have no formal meaning and would create security issues on name coding. As one XML expert has commented:

...one area where I think OOXML goes seriously wrong: in a few places it provides a mechanism for circumventing XML's character repertoire restrictions. I think the idea that just because someone generated an automatic name and used the backspace character as part of it, this should be regarded as acceptable practice in the standard is completely bogus. Several National Bodies have commented on it: I hope ECMA will have the good sense to remove it or severely deprecate it at the least. For example it is clearly a security hole to allow backspace in names, where the visible name may be coded differently than its readers expect: a kind of spoofing<sup>7</sup>.

Therefore, Ecma's Proposal to use attributes of type `xsd:string` is not a "fix." What the Proposal suggests in its including `ST_Xstring` is to introduce an alternative encoding in the lexical space of a string, in order to bring in character values that are otherwise not allowed in XML. The standard, conventional, and better way to handle this is not to use an `xsd:string`, but to use `xsd:base64Binary`.

**Not helpful for compatibility and cost burdens for customers, legacy behavior is still promoted in the proposal.** We have two examples where mapping for "legacy purposes" is unnecessary and creates instead a bias for conversions supported solely by Microsoft's systems.

One, comments on the use of OS-dependent path references to pictures (see CA-0023) were "addressed" by a proposed text change to reference fields located by IRI's (the successor to URL, to allow East-Asian Characters). While a good start, the proposal continues to maintain, for "legacy purposes" the existing behavior where files are referenced by Windows dependent file path specification, e.g., "g:\shared\pix\sky.gif."

Preserving the legacy behavior (OS-dependent paths) is unnecessary since it maps trivially and flawlessly into the new encoding (using IRI's). This conversion is supported by Microsoft's own `Uri` .NET class<sup>8</sup> as well as the `UrlCreateFromPath()` Windows API function<sup>9</sup>. So there remains no good reason to include the legacy behavior at all. It is not needed for compatibility; it should be eliminated altogether.

Two, Australia requested in its comments (see AU-0015) the mapping between the legacy codes and the ISO 639 language and country codes. No such mapping was provided. Ecma's proposed resolution shifts some text around but leaves the same underlying flaw. The remaining dual-mode solution offered by Ecma provides no additional expressivity or user benefit, but does increase the cost to implementors.

The need for legacy compatibility is acknowledged, but we believe that the use of ISO639 (or RFC

<sup>7</sup> [http://www.oreillynet.com/xml/blog/2008/01/the\\_design\\_goals\\_of\\_xml\\_1.html](http://www.oreillynet.com/xml/blog/2008/01/the_design_goals_of_xml_1.html)

<sup>8</sup> [http://msdn2.microsoft.com/en-us/library/system.uri\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/system.uri(VS.71).aspx)

<sup>9</sup> [http://msdn2.microsoft.com/en-us/library/bb773773\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/bb773773(VS.85).aspx)

4646) is directly mappable from the legacy codes in an unambiguous, lossless way. It should be the responsibility of those applications that deal with legacy binary documents to perform this conversion.

### *E. What is Missing? Interoperability. Why ignore re-use of proven standards?*

OOXML as originally submitted has numerous references which inhibit interoperability as they are tied to Microsoft platforms or application. In numerous comments, NBs requested that embedded objects, audio files, images, macros, etc. be de-linked to proprietary Microsoft technologies. The proposal to address this concern fails to remedy the interoperability needs. The proposal removes the references without supplying any documentation – no normative list of supported technologies.

One example is the proposal (see Responses 12 and 45) to change embedded objects to no longer reference OLE Link and OLE2 (which are Microsoft specific). That is good, but the proposal does not define them, but rather makes them implementation specific. While the proposers “agree that it is important for the specification to support multiple types of object linking,” they suggest changing oleLink(OLE Link) to oleLink(Generic Object Connection). And, instead of referencing the specific OLE2 connection they say to use any generic 'embedded object'. **Thus OLE is normatively dereferenced, and no list is supplied to replace it nor is any commonly used, non-proprietary standard suggested.**

Second, image types are no longer listed and are now implementation specific (see Responses 83 and 746). "Agreed; the intention was that all image formats are allowed, but the documentation referenced WMF and EMF, which was a mistake. .... There was a set of enumeration values provided for guidance, but any specific MIME type could also be used (for example, for TIFF; PNG; JPEG)." **Thus again the Proposal goes from a list for guidance (which was Microsoft specific) to no list at all.**

Lastly, macros are now too implementation specific (see Response 101). The proposal says "The mechanism by which the command specified by text in field-argument-1 is located and/or executed by an application is implementation-defined."

### *F. What is Missing? Support for legacy documents. Why deprecate in Version 1?*

Deprecation in the initial version of a standard is highly unusual, although not unprecedented. **While the concept has been seen before when multi-vendor support for a standard was already in-place, it is unclear what this achieves for OOXML, which lacks a substantial base of commercial implementations.** For example, the purpose for deprecation in C++ was to document, in an Annex, the various language variants which were in commercial use by one or more vendors, but which were not desired going forward. An implementor of ISO C++ was free to implement the core, non-deprecated core language, as well as to implement deprecated features, as needed for compatibility with existing source code which used those features. In the case of DIS 29500, deprecation is not

needed for multiple reasons:

- Unlike for C++, there is not a large body of existing OOXML implementations that are worth documenting in this way.
- Given that the very reason for OOXML is stated to be to support legacy documents, deprecating the features said to be necessary to support these very documents renders the whole point of OOXML totally unclear.
- Deprecated features in fact do nothing to address the millions of legacy documents because these documents are not in the OOXML format. This is even more true given the lack of mapping between these binary formats and OOXML instructing how these features are to be used when converting legacy documents to OOXML.

In addition to being purposeless, the proposed deprecation causes additional problems:

- Changing the contents of 59 clauses scattered across more than 6,000 pages (more work)
- Doubt on IP protections of these features once they are deprecated and moved to an Annex given that Microsoft's royalty-free licensing commitment only applies to what is "necessary" (more risk)
- Sub-sets of the original problem (same lack of legacy support)

For example, in response to sixteen NBs raising objections to the use of VML in OOXML, Ecma Response 92 introduces "sub-sets" of the problematic references. The proposal is to deprecate VML, move it to an Annex . And, in addition to DrawingML and VML, CT\_ObjectPr type, and CT\_ControlPR type, aCT\_ObjectAnchor type, etc., (which are all subsets of VML) are introduced.

Until the revised text is provided, the sufficiency of this proposal is difficult to evaluate. However, from the details provided, it appears that Ecma is merely taking a subset of VML, giving it another name (DrawingML), and using it in places where VML was previously called for. What is deprecated merely re-enters through the back door.

### ***G. What is Missing? Consistency of Fixes. Why does a disposition of a comment made to one National Body contradict one shared with another National Body?***

For one NB, the proposal suggests one course of action and for another NB the proposal suggests a contradictory course of action, e.g., contradicting where the primary schema reference resides and defining how macros can be executed.

The first example is the contradiction between GB-0061 and DK-0053. The two proposed "fixes" (see Responses 222 and 691) come to the opposite conclusion on a question over precedence between the text and the schema in the electronic annexes. Which of these two sources is definitive is extremely important to the specification.

The line from Part 2 of the OOXML specification is: "If discrepancies exist between the electronic version of a schema and its corresponding representation as published in this part, Part 2, the

electronic version is the definitive version."

In Response 222, the proposal states "Agreed; this statement should be eliminated." But, in Response 691, Ecma states "Disagreed" and "Those fragments are intended to be the same as the corresponding information in the zip file. However, if they are not, the ZIP file versions should be considered the definitive versions." Which schema has precedence is rather important to a specification for an XML based document format. **These contradicting responses leave unanswered the important schema precedence question.**

Second the proposed resolution to Australia's comment (see AU-0009) is in conflict with the resolution to the United Kingdom's comment (see GB-0265). Ecma's proposed resolution to Australia says to insert the text, in Part 1, "It is a requirement of this standard that dynamic extension mechanisms, such as scripting languages and macro mechanisms, shall use, for the executable parts, the correct content types, and shall not use any of the content types already defined in this standard."

However, the British NB (see GB 0265) noted the MACROBUTTON field in WordProcessingML can run a macro. Ecma's proposed resolution of GB-0265 was to add the following language to Part 4, Section 2.16.5.41: "The mechanism by which the command specified by text in field-argument-1 is located and/or executed by an application is implementation-defined."

These are clear contradictions. Further analysis of the proposal and comparison of it country-by-country might reveal additional contradictions.

### **III. Summary – Why You Should Vote No**

From the start OOXML was inappropriate for Fast-Track processing, both for low level of initial quality as well as because of its length and complexity. It is a 6,045 page specification with over 1,000 proposed fixes described in a 2,293 pages report. Do we really want to rush this further? Further rushing of defect fixing and further review will only lead to greater problems. It should be clear now that DIS 29500 needs more time in committee process to mature as a specification before consideration as an International Standard can be given.

Furthermore, the proposed changes to DIS 29500 fail to address harmonization, naming confusions, interoperability, and support for legacy documents. Can we in good faith endorse a standard that is not technically sound with conflicting recommendations on technical remedies? It should be clear, as described above, that OOXML has been vastly changed with limited improvements and new errors introduced.

Therefore, we urge a "Disapproval" position on OOXML and recommend that a new work item be initiated within JTC1/SC34 to take the draft of OOXML and harmonize it with ODF at a pace which allows for a high level of review and quality.

## **Annex A**

### **The BRM and Afterwards**

#### ***The BRM – Determines Changes to OOXML and NOT its Approval or Disapproval***

The BRM, or Ballot Resolution Meeting, will occur February 25/29 in Geneva. All NB's who voted on the September 2 OOXML ballot are able to attend, and approximately 35 NB's are planning on sending delegates.

The intent of the BRM is to decide on changes to the text of OOXML that will make it more acceptable. Proposals for changes may come from Ecma's as well as from NBs. Resolutions may be debated, amended, substituted, approved, rejected, etc., according to a vote of the meeting.

At the adjournment of the meeting will be an agreed upon set of editing instructions for the Ecma Project Editor to apply to OOXML. Only changes approved by the BRM may be made to the standard. Note that the BRM does not indicate approval or disapproval of the OOXML standard itself. Its purpose is to play a technical role to suggest changes to the text of the standard.

#### ***After the BRM – 30 Days to Change Vote in any Direction***

After the BRM adjourns (February 29<sup>th</sup>) there will be a 30-day period in which those NBs who voted on the September 2<sup>nd</sup> ballot will be able to reconsider and change their position. They can change their vote in any direction, from approval to disapproval, approval to abstention, abstention to approval, abstention to disapproval, disapproval to approval, or disapproval to abstention. NBs should consider in this final vote the “complete specification, not only the changes to comments they have submitted in the 5 month ballot period. The criteria for the vote is the same as on September 2<sup>nd</sup> –

1. more than 66% of JTC 1 P-Members need to approve
2. less than 25% of all votes cast need to disapprove

Note: Abstentions do not count in the above calculations.

Remember, the scope of the entire specification has changed, a reconsideration of any vote is appropriate. This open, technically-rich and inclusive process for setting an international standard (as described in the opening and conclusion of this paper) and the technical deficient OOXML (as described in the middle of this paper) are just not complimentary to each other. We strongly urge a vote of “No” to DIS 29500.