

# Unresolved Technical Concerns In DIS 29500 (OOXML)

Oracle Corporate Architecture Group  
March, 2008

This document details a number of current, unresolved, technical concerns with DIS 29500 (OOXML) that persist in the specification even after the Ballot Resolution Meeting (BRM) in Geneva during the week of February 25<sup>th</sup>, 2008. While this is not an exhaustive list of outstanding concerns, the following highlight the unsuitability of DIS 29500 for Fast Track consideration. As a whole these unresolved concerns support Oracle's position that National Bodies should not approve DIS 29500. Instead, National Bodies should request that Ecma and Microsoft take advantage of the normal JTC-1 standards development process for OOXML, given the level of market interest in open and interoperable XML-based document formats.

## 1) No Mapping of Binary Formats to OOXML

Microsoft claims this issue was resolved by the BRM, but the Ecma proposals responding to comments on binary mapping that passed the paper ballot refused to provide this mapping. No other application supporting OOXML will be able to faithfully or fully recreate the look of Microsoft's legacy binary documents. Although the binary Office document specifications have been posted by Microsoft, no standardized mappings were offered during the BRM, as requested by the US, United Kingdom, Brazil, and Malaysia, among others. Binary mappings explain how to translate a binary document into OOXML, or provide standardized guidance on how to "represent faithfully" legacy documents. Without standardized mappings, the same binary document source document will produce different OOXML documents in Microsoft Office, Apple iWork, OpenOffice.org, etc., breaking interoperability and preventing the realization of OOXML's stated goal of preserving legacy documents.

## 2) Major Changes to XML Schemas

There were several major changes to the XML schemas. These changes were often only partially referenced in Ecma's proposed dispositions. The current version of the draft standard has three copies of the XML schemas describing the various markup languages that comprise OOXML: Normative XML fragments in the run of the text (to be moved into an appendix), full schemas listed at the end of the Markup Part, and annexes containing copies of the schemas. All of the schemas were duplicated in order to create a strict and transitional schema for each document type (word processing, spreadsheet, and presentation).

These new schemas have never been seen, and have never been reviewed. There is a high probability that the resulting draft and schemas will contain errors, and that the schema used to validate documents will not be the one provided in the standard. Because the

default behavior for conformance is to ignore elements and attributes that are not understood, differences in the copies of the schemas will likely lead to interoperability problems with conforming applications ignoring the problems.

### **3) Significant Changes to the Scope of the Draft Specification**

There are a number of fairly large scope changes that are the result of proposed changes from Ecma. Worse yet, the actual changes will not be known until some time after the BRM (when the editor releases the final draft), and will not be available for in time for national bodies before they must make a final decision on the specification. These changes in scope result from:

- The document wide review of the usage of the word 'Shall' permitted by the resolution to allow the editors to rewrite the specification using ISO constructs (shall, should, etc.) without any review
- The restructuring of the document into a multi-part specification
- The number of existing specifications that have been added as normative references (e.g. ISO 8601 for dates)
- Lists of content types that have been removed and replaced with application-specific designations
- Large number of comments changing specific content either to normative or informative

In particular, the decision to permit Ecma to make wholesale changes to normative sections of the specification that impact what vendors will have to (or don't have to) implement is unprecedented. Much of the work of standards development pertains to the correct and desired application of ISO constructs (Shalls and Shoulds), and abdicating this responsibility to project editors is simply irresponsible.

Given the lead time required in making decisions for many National Bodies, most (if not all) National Bodies will not be able to review the final draft of the specification before passing judgment on it – a situation that should be unthinkable given the extensive level of redrafting that is required. How can National Bodies decide on their final vote, whether the final draft of DIS 29500 is in appropriate shape to be an ISO standard, when the final draft will look nothing like the draft they have reviewed, knowing that there will be substantial changes to the scope of the document, and they will not have timely access to the final draft?

### **4) Macros**

Four National Bodies, in their ballot comments from last September, pointed out that Section 2.16.5.41 of DIS 29500's Part 4 defines a "MACROBUTTON" field that allows the definition of a button in the document that will trigger a macro. But little is said about how the macro is stored, bound, what API's are available, or what the security model is for this feature. One National Body requested that Ecma "Describe this feature to a level where cross-platform, cross-application interoperability is possible." What Ecma provided in their draft Disposition of Comments report (approved in batch by the BRM

without discussion or opportunity for objection)<sup>1</sup>, was something quite different and unsatisfactory. Ecma simply added: "The mechanism by which the command specified by text in field-argument-1 is located and/or executed by an application is implementation-defined". Unfortunately, with this addition, not only is it impossible to have cross-platform interoperability of this feature, it is unlikely that vendors will be able to implement a reasonable security policy to detect, scan or block macros included in documents.

## 5) Dates

27 National Bodies raised objections to the way in which OOXML handles spreadsheet dates. OOXML does not allow dates before 1900 and it requires that the year 1900 be (incorrectly) treated as a leap year. These date concerns were discussed and approved at the BRM, although with little real improvement. Rather than fix the erroneous leap year calculation, Ecma's resolution, while adding the possibility to store dates in ISO 8601 format, also kept in place the old error (leaves room for serial numbered dates to be used rather than the standard form) and in effect now leaves would-be developers with **five** different ways of representing dates. Ecma's approach to fixing a simple problem of representing dates within a document format once again creates barriers to interoperability.

## 6) Spreadsheet Formula Bugs

Although the CEILING function was recognized to have a legacy bug and fixed during the BRM, there may exist more mathematical inaccuracies in OOXML's spreadsheet function. e.g. The FLOOR function has been identified to have a similar but for negative numbers. Further work is necessary to review the accuracy of the spreadsheet functions before this specification is approved.

## 7) Known Contradictions Exist in Editing Instructions

The paper ballot at the BRM dealt with all unaddressed issues. This mass approval included, among other issues, approval of responses that directly contradicted each other. While the following example is simply the worst example we have come upon, it is not the only example of contradictions in the responses. Incredibly, Microsoft and Ecma specifically knew of this problem before the BRM through a widely published and discussed "Top Ten" list of concerns, however no action was taken at the BRM to resolve this issue due to lack of time.

Included in the proposed responses Ecma published in preparation for the BRM were the conflicting Responses 222 and 691. These two proposed responses contradicted each other in terms of the editing instructions and the concept of whether XML schema contained in the text or in the electronic annex have primacy. In Response 222 Ecma suggests the removal of Part 2, Annex D, page 81, lines 5–6. In Response 691 a National Body also requests the lines be removed, however, the proposed disposition by Ecma

---

<sup>1</sup> Ecma Response 101, approved at the Geneva BRM in a 9-4 vote as part of a large batch of 1027 changes, without discussion or opportunity for dissent.

disagrees with this change.<sup>2</sup> This particular example is further confused, as the entire schema is being copied into the document as part of the restructuring of the draft standard into a multiple part standard. In other words, there are now at least 3 copies of the schemas, with some question as to which is the definitive copy.

This is a perfect example of how the lack of discussion time at the BRM affected the quality of the output. The conflict between these two proposals was published before the BRM, and was known to Microsoft and Ecma. This issue could easily have been resolved, would not have achieved consensus in its current form, and was passed using block votes that prevented discussion on critical technical details of the specification.

### **8) Vague Conformance Clause for Conforming Applications**

This approach used in the current draft of DIS 29500 is fatally flawed in terms of interoperability, as the conformance clauses create a situation where almost any application could be considered conforming. At the same time “conforming” applications have little likelihood of being able to share documents with each other with any amount of fidelity. Here are the current proposed text for the conformance clause for applications:

- A conforming consumer shall not reject any conforming documents of at least one document conformance class.
- A conforming producer shall be able to produce conforming documents of at least one document conformance class.
- A conforming application shall treat the information in Office Open XML documents in a manner consistent with the semantic definitions given in this Specification. An application's intended behavior need not require that application to process all of the information in an Office Open XML document. However, the information that it does process shall be processed in a manner that is consistent with the semantic definitions given in this Specification.

For implementers of DIS 29500, the conformance clause means:

- A conforming consumer needs to open a conforming document without throwing an error.
- A conforming producer needs to at least create a single conforming document.
- Any features of OOXML implemented by this application needs to be implemented faithfully to the standard.

One example of how any application could pass this conformance clause would be the GNU cp command for copying files:

---

<sup>2</sup> **Response 222** -- Proposed Ecma Disposition: Agreed; this statement should be eliminated. The following changes [deletion] will be made to Part 2, Annex D, page 81, lines 5–6 and Part 2, Annex E, page 82, lines 6–7: ~~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.~~

**Response 691** -- Proposed change by the National Body - Delete Part 2, Annex D, p 81 line 5-6.

"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".

Proposed Ecma Disposition: **Disagreed** ....

- Start with a conforming WordProcessingML document - test.docx
- Run the command 'cp test.docx test\_copy.docx' - no errors are thrown, and the new copy of the conforming document is created.
- This has now shown:
  - 'cp' is a conforming consumer of OOXML, as it did not reject the conforming document and all features of the standard that are implemented in the application (none) are implemented faithfully to the specification.
  - 'cp' is a conforming producer of OOXML, as it created a conforming document and all features of the standard that are implemented in the application (none) are implemented faithfully to the specification.

Some other applications that could be considered conforming applications:

- pkzip or infozip - zip programs can open and create conforming documents without error
- GNU command 'cat' – the file will be displayed as raw text without throwing an error
- the trash can (windows, Mac, or Linux!)