



## OOXML - Top Ten Worst Responses (and one more thing...)

If you have the unenviable task of reviewing Ecma's Proposed Disposition of Comments on DIS 29500 - the OOXML specification – you might initially be impressed that Microsoft and Ecma have produced answers for all of the thousands of concerns raised about the specification. However, if you take the time to become more familiar with the content of those answers, you will see that not all of the concerns are resolved in a way that improves or moves the specification forward. To have some fun, here is a list (with apologies to Dave Letterman) of the Top Ten Worst Ecma Responses:

10. Worst Ignored Request - Response 620 - Identifies 'w:sz' as an element with 'major internal inconsistencies'. Elements have units of half points, points, pixels, percentages, etc, depending on the context, and when used as an attribute it can be eighths of a point, hundredths of a point, percentage of the screen, or even the values "full", "half", and "quarter". The response agrees that "the sz attribute/element has different meaning depending upon its parent element", but is determined to keep things the same. No change is offered.
9. Worst non-Answer - TIE!! Responses 759 and 862 - Both merely quote the comment. For Response 759 the proposed disposition states: "In the Numbering row of the table in Part 4, §2.16.5 page 1508, the second part of the description of 'Numbering' makes no sense." Just that, no changes.
8. Worst use of XML - Response 559 - The use of the element 'e' breaks XML usage conventions. Ecma's response lists 18 different uses for the element 'e' but does not solve the issue, rather it further emphasizes the poor quality of this specification and improper way in which XML has been used. No change is offered.
7. Worst Introduction of Security Holes - Response 102 - This Ecma response supplies a list of reserved values for hashing algorithms, the first four of which (MD2, MD4, MD5, and RIPEMD-128) are followed by this note: "It is recommended that applications should avoid using this algorithm to store new hash values, due to publically known breaks."
6. Worst 'Back Door' Tactic - Response 82 - The comments ask to 'specify' or 'remove' the ink element which is a placeholder for a non-interoperable application-defined functionality. Ecma's proposal adds a type element to the XML schema and recommends using InkML, which is a W3C Working Draft, but leaves open their ability to use non-interoperable application-defined formats. To make matters worse the response is littered with references to VML which is proposed to be deprecated in other responses, even stating 'An ink object is a VML object...'. Is VML reentering the specification?
5. Worst 'Baby and Bathwater' (part of a popular US phrase that means throwing away the good with the bad) - Another TIE! Responses 425 and 624 - The comments ask for explicit support for embedded objects from KDE and Gnome, in addition to Microsoft's OLE objects. Ecma's response states that OOXML allows for "a general method for embedding objects from an external component technology" and removes all references to OLE. In light of the heavy use

of OLE embedding in current MS Office documents, interoperability greatly depends on this functionality. Changing the language of the specification to not directly call embedded objects 'OLE objects' (i.e. removing the normative reference) has the potential to substantially degrade interoperability of OOXML documents. This makes embedded objects implementation-specific in the eyes of the specification.

4. Worst Standardizing of a Microsoft Bug - Response 782 - The comment questions an example for FILESIZE where 4660736 bytes is rounded to 4661 kb. This is incorrect; it uses a division by 1000 instead of 1024. The result should be 4552 kb. Instead of fixing the example, the Ecma proposal changes the unit from 'kilobyte' to 'thousand bytes' (and 'megabyte' to 'million bytes'). This is odd because file sizes are normally calculated in kilobytes and megabytes. This proposal suggests a bug in Microsoft's current application, where the programmer divided by 1000 instead of 1024 (and 1000000 instead of 1048576 which is  $1024*1024$ ) The Ecma proposal favors an existing bug over obvious behavior.
3. Worst 'Most ways to Skin a Cat' Answer - Response 43 - Ecma's response suggests that instead of having one standardized way of representing dates (ISO 8601), that there should be five! In addition to ISO 8601, they propose two legacy representations with base dates 1900 and 1904 with no support for dates before these years, plus two brand new representations based on the 1900 and 1904 systems which handle years before the base date but are not thoroughly defined.
2. Worst Dueling Answers -Response 222 and Response 691 – In this example, of considerable importance, the two proposals come to the opposite conclusion on a question over precedence between the text and the schemas in the electronic annexes. Which of these two sources is definitive is extremely important to the specification. Response 222 proposes to remove a line of text giving precedence to the schemas, stating "Agreed; this statement should be eliminated". Response 691 rejects that very suggestion, stating "Disagreed." and "... the ZIP file versions should be considered the definitive versions."
1. Worst Case for spending days of my life reading through these answers - Response 885 - "Agreed; deprecated features should not be used in newly created Office Open XML documents." ALL OOXML documents will be NEWLY CREATED. While there are many documents in the legacy binary format, only a few are in the OOXML format which is not even finalized yet. Except for an extremely small number of documents (a rounding error compared to the oft quoted 'corpus of existing documents'), all OOXML documents will be “newly created” either by creating them from scratch or by converting existing binary documents. If the differentiator between OOXML and ODF is the legacy compatibility features (see Response 926 for the revised scope), and these features are deprecated and not to be used, then why are we doing this in the first place?

But wait, there's a Bonus Worst ...:

Worst 'We'll Get to it Later' – (A TIE!)

- Response 803 - The comments ask for a clear definition of the parameters in function definitions. Ecma's response promises these, but only supplies 3 examples of how changes will be made. Because the definition of the functions is central to the interoperability of spreadsheets, these are highly consequential changes.
- Response 1 - Improper use of the word 'shall'. There are over 6000 instances of them

and Ecma's response is: "Agreed; some uses of “shall,” “may,” and so on, are inconsistent. Rather than enumerate all places that need changes in this response, general comments on such usage will be addressed in an editorial pass over all parts."

Interestingly, neither of these proposed dispositions (if accepted at the BRM) can be reviewed until the final draft of DIS 29500 is available, which is likely to be after many of the National Bodies will meet to finalize their votes.

For those with access to Ecma's proposed disposition document, here are the NB comment numbers referenced to the Ecma responses:

Response 1 (AU-0007, DE-0042, DK-0076, ECMA-0018, GB-0014, JP-0006, JP-0007, JP-0008)

Response 43 (AU-0016, BR-0046, CA-0044, CH-0006, CH-0007, CH-0017, CL-0013, CL-0147, CO-0035, CO-0154, CO-0155, CO-0156, CZ-0009, DE-0030, DE-0031, DE-0032, DE-0072, DK-0033, DK-0136, DK-0137, DK-0153, FI-0013, FR-0182, FR-0183, FR-0351, GB-0300, GB-0301, GB-0304, GB-0363, GB-0364, GH-0002, GR-0003, GR-0004, GR-0005, GR-0006, IE-0002, IN-0007, IN-0057, IN-0058, IN-0061, IN-0080, IR-0001, IR-0002, KE-0054, KE-0055, MX-0005, PE-0002, PE-0003, PH-0005, PT-0085, SG-0002, US-0130, US-0131, UY-0003, VE-0011, VE-0060, ZA-0014)

Response 82 (BR-0057, CL-0209, FR-0373, GB-0485, PT-0111, US-0157)

Response 102 (CA-0037, CL-0027, CL-0028, CL-0055, CL-0197, CL-0202, CO-0096, CO-0143, CO-0146, DK-0030, DK-0114, DK-0139, FR-0338, FR-0341, FR-0345, GB-0219, GB-0291, GB-0292, B-0298, GH-0008, GR-0021, GR-0022, GR-0076, IN-0010, IN-0026, IN-0063, IN-0064, IN-0077, IR-0012, IR-0047, JP-0068, KR-0024, MY-0017, PT-0090, PT-0091, PT-0093, US-0027, US-0051, US-0138, US-0144, US-0252, VE-0001, VE-0017, VE-0054, ZA-0009)

Response 222 (GB-0061)

Response 425 (DK-0031)

Response 559 (GB-0537)

Response 620 (IN-0011)

Response 624 (DK-0032 and ECMA-0045)

Response 691 (DK-0053)

Response 759 (FR-0147)

Response 782 (GB-0256)

Response 803 (FR-0453)

Response 862 (FR-0356)

Response 885 (GB-0013)