



The Technical Case Against DIS 29500/OOXML

This paper presents the principal arguments on why DIS 29500 “Office Open XML” (OOXML) is not an acceptable standard, as defined by recognized criteria. Though there are several hundred additional problems with OOXML (available upon request), this paper addresses just a few. Given the severity of these problems, we urge National Bodies to vote “no” (disapprove) on OOXML.

1. Standards should be repeatable, optimal/best practices and interoperable. Recognized standards bodies have used those words in defining standards (underline added for emphasis):

- ISO says a standard is a: “document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context.”¹
- The British Standards Institute (BSI) says: “...a standard is an agreed, repeatable way of doing something... They are intended to be aspirational - a summary of good and best practice rather than general practice.”²
- ISO/IEC JTC1 Directives say: “A purpose of IT standardization is to ensure that products available in the marketplace have characteristics of interoperability, portability and cultural and linguistic adaptability.”³

This paper demonstrates how OOXML does not satisfy the criteria of being precise, repeatable, common, aspirational, best practice and culturally and linguistically adaptable. Keep in mind that these are merely a handful of examples. The sheer volume of serious problems with OOXML, known and unknown, speaks against its maturity and suitability as an ISO standard.

2. Precise, Repeatable, Common -- *Does the OOXML specification provide sufficient, detailed, written description to allow for the common practice of the technology?*

The examples below illustrate OOXML does not.

First, the "Compatibility Settings" WordProcessingML⁴ section within OOXML does not provide for repeatable practices. While it provides Microsoft the ability to store information related to various behaviors in their legacy file formats, the specification merely lists the names of these settings without

1 ISO/IEC Guide 2:2004, definition 3.2

2 <http://www.bsi-global.com/en/Standards-and-Publications/About-standards/What-is-a-standard/>

3 JTC1 Directives, 5th Edition, Version 3.0, Section 1.2

4 Part 4, Section 2.15.3.9. All OOXML section references are from Ecma 376 “Office Open XML” specification, available at <http://www.ecma-international.org/publications/standards/Ecma-376.htm>.

proper definitions. This clearly is not precise and certainly does not provide for repeatable or common practice of these features. An OOXML-consuming application, presented with a document using these attributes, will be unable to interpret them properly and render the page in a high-fidelity manner. Further, since these attributes are merely listed but not defined, the ability to practice the benefit of being “fully compatible with the large existing investments in Microsoft Office documents” (the goal of OOXML according to its authors)⁵ is consequently reserved for Microsoft alone.

Second, the WordProcessingML part of OOXML lists a large number of styles representing various different writing systems, language and business conventions.⁶ These are given names such as “chicago”, “ideographDigital”, “ideographLegalTraditional”, and “koreanDigital2.” These are mere labels, and again, not precisely defined. The would-be implementors of the OOXML specification are told that something called “Korean Legal Numbering” exists, but they are not told what it means or how to practice it in their application.

Third, the SpreadsheetML part of OOXML describes a “securityDescriptor” attribute, which according to the specification⁷ “defines user accounts who may edit this range without providing a password to access the range. Removing this attribute shall remove all permissions granted or denied to users for this range.” A would-be programmer implementing this feature would need to know, with certainty, the level of precision how these user accounts are set up and allowed. OOXML does not provide those details (although it does seem to imply that more than one are allowed). Since there is no universal concept of digital identity, it is crucial to know where these user accounts live. There will be no interoperability (which in the end is what repeatable, common use is all about) with a feature ill-defined in this way.

In summary, many areas of OOXML are undefined such that no other implementation will be successful in reproducing equivalent features. Although it provides a good framework for Microsoft in which to represent its own documents, it does not provide equal access for others to obtain similar benefits. The lack of precision leaves OOXML, albeit bloated at 6000 plus pages, neither repeatable nor common.

3. Aspirational and a consolidation of best practices -- How well does the OOXML specification incorporate the work of the other recognized and widely implemented standards, such as those from the World Wide Web Consortium (W3C)⁸ which maintains the core XML technologies (e.g., XHTML, XForms, SVG, MathML, SOAP, etc.)?

OOXML uses very little of the consolidated best practices of the industry. Worse, would-be implementors of OOXML would have to use proprietary, legacy formats used by Microsoft, and only them, even when relevant W3C standards are available. OOXML is not aspirational.

First, Vector Markup Language (VML) was developed by Microsoft and proposed by them to the W3C,

5 Part 1, Introduction

6 Part 4, Section 2.18.66

7 Part 4, Section 3.3.1.69

8 <http://www.w3.org>

where it was evaluated by a technical committee and rejected. The industry-supported Scalable Vector Graphics (SVG) was developed into a standard by the W3C and widely adopted over the past decade. OOXML uses the proprietary VML (a 600 page embedded specification) despite Microsoft's acknowledgment that VML is the wrong “deprecated” standard to use for vector graphics.⁹ The amount of extra work this causes for everyone, but Microsoft, who wishes to implement OOXML is immense. They would need to support two different markups for the same function, without additional benefit to their users. Only Microsoft is spared the duplicate work.

The second example is the definition of spreadsheet dates. OOXML does not use the Gregorian Calendar, the base calendar of commerce, science and government worldwide, for “legacy reasons.”¹⁰ The result is that all would-be implementors of OOXML are required to have their applications give their users incorrect answers to questions like “What day of the week is February 1st, 1900?”, if they want to be conform to the OOXML standard. This causes particular pain in the common task of exchanging spreadsheet data with relational databases via ISO SQL, a standard which explicitly requires the use of the Gregorian calendar.¹¹

Third, rather than using well-known, accepted algorithms, like SHA-256, OOXML requires the use of an unnamed, legacy cryptographic algorithm¹² described only by C-language source code, lacking peer review or use by anyone outside Microsoft. Microsoft doesn't even recommend using these algorithms. Instead, they provide DRM-based protections in Office 2007. These DRM-protections are not documented in OOXML, so no other vendor is able to freely use those features. Instead would-be implementors have only the flawed legacy security support of OOXML, support which is not even FIPS-180 compliant. Again, Microsoft is keeping the best practices to themselves, and leaving the OOXML specification with crippled security.

OOXML is a literal porting of the features of a single vendor's binary document formats. The avoidance of re-using relevant existing international standards, as well as the inconsistent use of Microsoft's own preferred technologies demonstrates that OOXML does not represent the consolidated results of science, industry and experience. It is not aspirational.

4. Interoperable and Portable¹³ -- *Can the proposed OOXML specification be fully implemented by multiple applications on multiple operating systems?*

9 Part 4, Section 6.1 - “The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in this Standard for backwards compatibility reasons. The DrawingML format is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a deprecated format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML”

10 Part 4, Section 3.17.4.1 - “For legacy reasons, an implementation using the 1900 date base system shall treat 1900 as though it was a leap year... A consequence of this is that for dates between January 1 and February 28, WEEKDAY shall return a value for the day immediately prior to the correct day, so that the (non-existent) date February 29 has a day-of-the-week that immediately follows that of February 28, and immediately precedes that of March 1.

11 Database Language SQL—Part 2: Foundation (ISO/IEC 9075-2:1999), Section 4.7.3

12 For example, in Part 4, Section 2.15.1.28 instead of, for example, using SHA-256 which is FIPS-180 compliant as well as being ratified by ISO/IEC 10118-3:2004

13Portability and Interoperability are two of JTC1's “Common Strategic Characteristics” and as such are requirements of all JTC1-approved standards.

The answer is no; OOXML is heavily tied to the Microsoft Office applications and to Microsoft Windows, to the detriment of interoperability and portability.

First, OOXML defines a ST_CF type¹⁴, which records the allowed clipboard formats which may be used with a graphical object. The allowed values of this type, EMF, WMF, etc., are all proprietary Windows formats. No allowance seems to have been made for use by other operating systems. For example, in Linux images are typically copied on the clipboard in an open standard format like PNG. But if a vendor encodes “PNG” into a document record of this type, the document and the application therefore will no longer conform with the OOXML standard.

Second, the definition of a password hashing algorithm in SpreadsheetML is given by presenting 5-pages of C-language source code.¹⁵ However, the bit manipulations of this code are inherently machine dependent, and will give different results depending on the processor architecture. That means a document created on one environment will not be able to be opened on another platform.

Third, the “optimizeForBrowser” element of WordProcessingML16 has been defined in a way that advantages Internet Explorer and ignores the existence of current browsers, e.g., Firefox, Safari, Opera. OOXML requires that “all settings which are not compatible with the target web browser shall be disabled.” What if one wants their application to produce standards-compliant output?

In summary, OOXML has been designed to work exclusively with the technologies supported by Microsoft Office. In some cases extraordinary efforts are made to incorporate other proprietary specifications, like VML, into OOXML. Not only does OOXML ignore alternative, standard and open technologies, it prevents other vendors from adding this support.

5. Cultural and Linguistic Adaptability -- Does it consider different cultures and languages?

OOXML fails to consider and achieve cultural and linguistic adaptability.

First, OOXML has a function called ,NETWORKDAYS()¹⁶ that does not define “weekend” and does not provide a way for the user to define it. As implemented in Excel, the function assumes the weekend is always Saturday/Sunday. This OOXML function is defined in a way which renders an incorrect answer for potentially billions of people across the globe who do not follow the Saturday/Sunday weekend.

Second, WordProcessingML in OOXML uses “Border Styles”¹⁷ which shares a closed list of specific named border styles with mandated images. As the list is a closed list (happens to match exactly what Microsoft Word provides), a would-be implementor of OOXML could not add image types to better suit their needs. An example of two such graphics is shown in figure 1 below. OOXML offers only two possibilities for displaying a globe in a page border and neither of them show Asia. Similarly, there are

14 Part 4, Section 6.4.3.1

15 Part 4, Section 3.2.29, pg. 1917

16 Part 4, Section 3.17.7.224

17 Part 4, Section 2.18.4

graphics for birthday cakes, painted Easter eggs, Christmas gingerbread men, and other images that are appropriate for a Western cultural milieu, but have less applicability elsewhere.

earth1 (Earth Art Border)	<p>Specifies an art border consisting of a repeated image of Earth, as follows (showing two repetitions):</p> 
earth2 (Earth Art Border)	<p>Specifies an art border consisting of a repeated image of Earth, as follows (showing two repetitions):</p> 

Figure 1: Illustration 1: Page Borders

Third, as mentioned previously, WordProcessingML defines a number of numeration styles for numbered lists.¹⁸ Rather than using a declarative/generative approach, such as used by the W3C's XSL:FO and OpenDocument Format, OOXML's styles are defined as a closed list, again matching what Microsoft Word supports, but is not extensible by other vendors. Since the list of styles provided is incomplete, they lack, for example, support for Armenian, Tamil, Greek alphabetic, Ethiopic and Khmer numerations, as well as the larger number of historic systems used by scholars.

Using Western working week conventions, images and number styles, OOXML fails in cultural and linguistic adaptability. The closed-ended lists which, although matching perfectly what Microsoft Office offers today, are not extensible by vendors in an interoperable way. Further, because the lists are closed, conversion with an open list will fail thus rendering interoperability failures.

6. Next Steps for ISO -- What to do when a specification fails to meet the criteria for a standard?

Reject OOXML as an ISO standard.

We urge National Bodies to vote “no” (disapprove) in this JTC1 ballot. While any vendor is entitled to their own design decisions and their own priorities, an ISO standard must have the characteristics of re-use, portability, interoperability and cultural and linguistic adaptability, so that all vendors may have that same right to their own design decisions and priorities. The arbitrary restrictions of OOXML, which work extremely well with Microsoft's products, but not others, render the proposed DIS 29500 “Office Open XML” specification inappropriate for approval as an International Standard.

For more information, please visit www.odfalliance.org.

¹⁸ Part 4, Section 2.18.66