

# Samples from Google analysis of Ecma's “Disposition of Comments” for National Bodies

Here is a small representative sample of National Body comments on DIS 29500, followed by Google's analysis of Ecma's response where we feel such responses do not adequately address the issues raised.

## National Body comment (CA-0041):

*“The securityDescriptor attribute, “defines user accounts who may edit this range without providing a password to access the range”. It is a string. But no information is given as to what user accounts are referred to here, or what the delimiter is. Are these comma-delimited local machine user accounts? Or semi-colon delimited LDAP DN's? There will be no interoperability if this is not defined.”*

## Google analysis of Ecma response:

The clarification of the Security Descriptor field does not fully fix the problem. The original specification described this field as “application defined.” The clarification adds a definition of this field, but still retains the capability for it to be “application defined.” The “application defined” capability should be removed, and the clarified definition of the field should be the only allowed use. Allowing application defined security descriptors (which is an obvious historical reference to the Microsoft Windows specific definition of the term) breaks the goal of interoperability for an important feature of document formats.

## National Body comment (NZ-0003):

*“Seek to harmonize with the existing ODF standard to reduce the:*  
*1. cost of interoperability*  
*2. cost of having two Standards*  
*3. cost of support/maintenance”*

## Google analysis of Ecma response:

The Ecma response to this issue is essentially to claim this is not a problem. We do not feel this is addressing the core issue with DIS 29500, which is that it is an unnecessary standard. The overlap between DIS 29500 and the existing ISO/IEC IS 26300 (ODF) standard is so great that the needs of the DIS 29500 designers could easily have been met by adding the additional features requested to a future revision of ODF. The response states *“Harmonization would require functional changes to two International Standards and would fall under the JTC 1 procedures for new work within SC 34 and could be done in the future. Such work should not be done in this Fast-Track process and should not impede the adoption of DIS 29500.”* We fully

agree such work should not be done in this Fast-Track process, and would prefer to make functional changes to one existing International Standard. On this basis alone, we feel DIS 29500 should be rejected.

Examining DIS 29500 it becomes clear this is not a serious attempt at an International Standard, but more of the enumeration into XML of the idiosyncrasies of one particular application format, Microsoft Office. Whilst the move to a documented format should be applauded, adopting it as an International Standard is a terrible mistake which will have serious repercussions for many years to come.

National Body comment (KR-0013):

*“autoSpaceLikeWord95 : unknown application behaviour”*

Google analysis of Ecma response:

The addition of deprecated features in an initial version of a standard is an obvious sign of errors in design. If the features are truly deprecated they should be deleted, which brings into question the fundamental goals of DIS 29500 in representing legacy documents. If DIS 29500 is meant to represent legacy documents the correct solution is to document that behavior, which has now been attempted, and to provide mapping from the legacy behavior into the new standard. What should not be necessary is the ability to store the flawed behavior of legacy documents in a new International Standard. The fact that DIS 29500 provides this capability (for example allowing the storage of Windows metafiles or Windows printer DEVMODE structures) betrays its design as a way to maintain application specific capabilities in what is supposed to be an inter-operable standard, implementable by anyone on any platform.

National Body comment (US-0037):

*“It is unsatisfactory to store printer settings in OS-dependent binary formats like DEVMODE structures. This is a considerable security concern (DEVMODE structures are loaded directly into device driver memory), as well as lacking cross-platform adaptability. There is also no interoperability with print settings as currently defined.”*

Google analysis of Ecma response:

This change is unacceptable. DEVMODE is a completely Windows-specific binary structure with no defined mapping into platform independent settings. The “printersettings” part should be removed, and definitions for platform independent printer features should be defined. An external specification that would suit is the Adobe PPD definition here:

[http://partners.adobe.com/public/developer/en/ps/5003.PPD\\_Spec\\_v4.3.pdf](http://partners.adobe.com/public/developer/en/ps/5003.PPD_Spec_v4.3.pdf)

National Body comment (BG-0001):

*“1. To be ensured compatibility of ISO/IEC DIS 29500 with other ISO/IEC standards i.e standards for numeric representation of the dates and time (ISO 8601), codes for the representation of names of languages (ISO 639), cryptographic hash functions (ISO/IEC 10118-3) etc.”*

Google analysis of Ecma response:

There is no explanation as to why keeping the flawed date system is required for compatibility with older documents. On conversion to DIS 29500 from older formats ISO 8601 dates can be used.

National Body comment (VE-0019):

*“This feature has been defined in a way which ignores the existence of current browsers other than Internet Explorer. What about Firefox? What about Safari? What about Opera? None of these can be set as target browsers. This section requires that “all settings which are not compatible with the target web browser shall be disabled.” But what if I want my application to produce standards-compliant output? So yes to PNG, no to VML, yes to MathML and SVG? I can't seem to specify this.”*

Google analysis of Ecma response:

The correct response to the problem of Web browser platform dependency is not to enumerate a list of platform specific Web browsers and the feature sets they support, which this change attempts to do. Such a list is a fundamentally flawed proposition in a standards document and doomed to be hopelessly out of date even before adoption. Either a mapping between OOXML features and standard HTML should be included, or references to Web display should be removed. Converting between XML document standards and Web display is an application specific feature upon which applications should compete.

The fact that this is seriously suggested as a solution in a standards document leads us to question the applicability of DIS 29500 to be considered as an International Standard.

National Body comment (CO-0237):

*“The naming of elements is very inconsistent. Even though the choice of one letter for common elements seems appropriate, there seems to be no common technique for naming. The capitalization and vowel removal is inconsistent, as there are elements with names like: (from, but not limited to, Part 4 Section 2.15.1): ActiveWritingStyle, attachedSchema, documentType, docVars, endnotePr, hdrShapeDefaults) . This is not a problem with the clarity of the specification, but it complicates the implementation of it unnecessarily, as developers will need to refer more often than necessary to the document to check for a certain element's particular spelling.”*

Google analysis of Ecma response:

The proposed disposition fails to address the problem. As documents are stored in a ZIP-encoded format, verbosity in element names or attributes does not create a storage size problem in practice. Ecma should expand the terse element names to have meaning for human readers. The fact that they have refused to address this reasonable request adds more evidence to the accusation that this is merely an attempted enumeration of the current Microsoft Office XML document format, not a serious attempt at a reasonable cross platform interoperable International Standard.

National Body comment (GB-0368):

*Dates as stored in SpreadsheetML are not processible by other applications of XML as they are not defined using the standard XML datetime datatype. It would be an easy matter to add an extra attribute, ISO8601datetime, to the representation that would allow a processible form of the date to be associated with the XML record of each SpreadsheetML specific date.*

Google analysis of Ecma response:

The resolution is flawed. There is no explanation as to why keeping the hexadecimal language code is required for compatibility with older documents. There should be no need for DIS 29500 supporting applications to ever store a legacy ST\_LangCode. Applications that read legacy formats should do the mapping into an RFC4646 code when reading such files.

Many issues still remain

There are many such issues remaining in DIS 29500, even after Ecma's responses. Google does not feel that DIS 29500 is of sufficient quality to qualify as an ISO standard and we urge National Bodies to vote “no” on the fast-tracking of this specification.