

PDFlib GmbH Notes on PDF/VT (ISO 16612-2:2010)

The notes below were collected during PDFlib GmbH's implementation of PDF/VT support for the PDFlib 8 VT edition. The numbers in the first column refer to section numbers in ISO 16612-2:2010.

PDFlib GmbH (www.pdfli.com)

March 22, 2012

| Clause No. | Paragraph | Type | Comment | Proposed action |
|------------|-----------------------------|-----------|---|---|
| 6.5 | second but last, and Note 4 | technical | <p>The requirement to write all DPart dictionaries in a single compressed object stream is misleading and contradicts PDF Reference, 3.4.6 Object Streams, which mandates »the number of objects in an individual object stream should be limited.« As the number of DPart dictionaries in a transactional file may be very large this clause would lead to arbitrary large object streams, which would result in a degradation of performance.</p> <p>The clause »...should be written in a single compressed object stream« and Note 4 »multiple compressed object streams might be preferred« somewhat contradict each other.</p> <p>If the intention was to avoid mixtures of DPart dictionaries and other object types in the object stream this should be clearly stated.</p> | <p>Replace the second but last paragraph with the following, and delete the second and third sentences of Note 4:</p> <p>»All DPart dictionaries should be written into a conforming PDF/VT file in compressed object streams as defined in PDF Reference. These object streams should contain only DPart dictionaries, but not any objects unrelated to DPart dictionaries.«</p> |
| 6.6 | 7 | editorial | <p>PDF Reference 3.2.6 »Dictionary Objects« mandates »No two entries in the same dictionary should have the same key.« Therefore paragraph 7 is redundant and can be deleted.</p> | <p>Delete the following paragraph:</p> <p>»No two keys present in the DPM dictionary shall have the same name after expanding hash escape sequences«.</p> |
| 6.6 | all | technical | <p>References from DPM dictionaries to other parts of the PDF object hierarchy are allowed. For example, DPM could contain a reference to the first page object of the corresponding record. This forces the XML representation to include those referenced PDF objects as well. Even worse, the objects in the page are linked in both directions (the graph containing the page tree contains cycles), which leads to infinite recursion in the XML mapping algorithm. See also comment on D.2.2 below.</p> | <p>Add a requirement that DPM should be self-contained, i.e. the tree comprising DPM for a DPart node should not contain any references to objects outside of the same DPM dictionary, or outside the set of all DPM dictionaries (to allow optimization by re-using DPM objects).</p> <p>Cycles in the DPM object graph should probably be prohibited to facilitate XML mapping.</p> |

| Clause No. | Paragraph | Type | Comment | Proposed action |
|------------|------------------------------|-------------|---|--|
| 6.7.3 | first paragraph after Note 4 | enhancement | The GTS_Scope key supports XObject re-use on the record, file, stream and global levels. However, it might be useful to re-use XObjects on arbitrary levels of the DPart hierarchy. For example, if the recipients are grouped according to country, XObject re-use among all records for a country may be desirable. | <p>Change the first paragraph after Note 4 to read as follows: »If present, the GTS_Scope key shall have a value of type “name” that is one of SingleUse, Record, File, Stream, Global, Unknown, or one of the names in the NodeNameList array in the DPartRoot dictionary.«</p> <p>Add the following clause before Note 6: »If the value for the GTS_Scope key is one of the names in the NodeNameList array in the DPartRoot dictionary, this XObject, or an XObject with an identical GTS_XID, should be referenced from one or more Do operators in the pages belonging to the subset of the document part hierarchy corresponding to the specified node name.«</p> |
| 6.7.5 | all | technical | The SMask key of an Image XObject supports only type stream, not dictionary. | <p>Change the first bulleted item in b) from »an SMask key with a dictionary value« to the following: »an SMask key with a stream value«.</p> |
| 6.7.5 | all | editorial | The whole clause is very confusing. It mixes definitions, requirements, and conclusions, and fails to distinguish the requirements for Form XObjects, Image XObjects, and common requirements for both types of XObjects. For example, Image XObjects do not support the Group key, but paragraph 2 mandates a Group key for XObjects in general. | <p>Restructure the clause, e.g. according to the following scheme: »A document is said to contain transparency if... An XObject is called encapsulated if it satisfies the following conditions: ▶ Form XObjects: if the document contains transparency the Group key shall be present... The graphics state parameters shall be explicitly set... ...content stream shall not contain marked content operators... ▶ Image XObjects: the ImageMask key ... the Intent key ... An XObject may have a GTS_Encapsulated key...</p> |
| 6.7.5 | first paragraph after Note 3 | technical | <p>The concept of encapsulated XObject is inconsistent regarding the treatment of XObjects referenced directly or indirectly from an encapsulated Form XObject:</p> <p>On the one hand, the Form XObjects requirements »graphics state parameter initialization« and »no layers« are explicitly mandated also for Form XObjects which are referenced from an encapsulated Form XObject.</p> <p>On the other hand, the rules for encapsulated Image XObjects are not mandated for Image XObjects referenced from an encapsulated Form XObject.</p> | <p>If consistent rendering of individual referenced Image XObjects is desired change the sentence: »If the Subtype key of an encapsulated XObject has the value of Image...« to the following: »If the Subtype key of an encapsulated XObject or an XObject referenced directly or indirectly from an encapsulated XObject has the value of Image...«</p> <p>Alternatively, if consistent rendering of the compound entity (encapsulated XObject plus all referenced XObjects) is considered sufficient, delete the phrase »including any content streams referenced from its definition« from the first paragraph after Note 2, and delete the phrase »nor shall it directly or indirectly refer to other XObjects whose content streams contain marked content operators that identify optional content« from the first paragraph after Note 4.</p> |

| Clause No. | Paragraph | Type | Comment | Proposed action |
|------------|------------------------------|-----------|--|---|
| A.2 | various | editorial | The spelling of keywords for MIME headers is inconsistent, e.g. »Content-Type« vs. »content type«. For clarity the original keywords from the MIME RFCs should be used. | Replace the following: »content type « with »Content-Type« (paragraphs 3, 4, 5) »content disposition« with »Content-Disposition« (paragraphs 5, 6, 7, 9) »content transfer encoding« with »Content-Transfer-Encoding« (last paragraph) |
| A.2 | first paragraph after Note 1 | technical | This clause requires identical NodeNameList values for all PDF/VT files in the stream, including referenced documents. This doesn't make sense since referenced files may conform to PDF/VT as well, but the DPart structure of referenced documents is completely ignored when processing the PDF/VT-2s stream. | Replace the following: »All PDF/VT files in a PDF/VT-2s stream shall have a DPartRoot...« with »All PDF/VT files in a PDF/VT-2s stream except referenced files shall have a DPartRoot...« |
| A.4 | 2 | editorial | See A.2 | Replace the following: »content disposition« with »Content-Disposition« |
| D.2.2 | all | technical | Recursive data structures (cycles in the object relationship graph) in DPM are allowed. For example, representing parent/child relationships with forward/backward references is naturally expressed with cyclic references within the object tree. However, the XML representation does not include any means for dealing with recursion. Representing recursive data structures in XML is not possible according to the specified algorithm. | Since prohibiting cycles in the object relationship graph may impose unjustified restrictions on DPM, the XML representation should be amended to properly deal with cyclic references, e.g. by introducing a mechanism based on XML's id and idref constructs. |
| D.2.2 | last list item before Note 2 | technical | PDF/VT allows PDF stream objects to be used in DPM. However, PDF 2.0 according to ISO 32000-2, draft 2012-03, does not allow stream objects in DPM which creates a small incompatibility between PDF/VT and PDF 2.0. | Avoid the use of stream objects in DPM. |
| E | Note 2 | editorial | Duplicate sentence | Delete the following sentence: »The GTS_ second class name prefix is registered by CGATS and is used in PDF-based ISO standards.« |
| F.2 | all | technical | The algorithm does not take into the account the relationship of layers and XObjects. Consider the OC entry in two XObject dictionaries in two different documents. Both may reference optional content group dictionaries with identical literal entries (i.e. same layer name), but the visibility rules of both layers may be different for both documents. As a result, two XObjects with different visibility could wrongly be treated as equivalent. If, on the other hand, two XObjects which are equivalent except for optional content group membership are intended to be treated as equivalent the reverse error may happen: two optional content group dictionaries may specify e.g. different layer names, but the document's visibility rules may activate both layers and the XObjects are incorrectly not treated as equivalent. | Redesign the equivalence testing algorithm to prevent wrong positive results in the presence of layers, e.g. by always treating XObjects with layers as not equivalent: »If one of the XObject dictionaries contains the OC key they are not equivalent.« |