

**ISO/TC 130/WG2/TF3 — Variable Data Printing**  
**Minutes of 3<sup>rd</sup> Meeting**  
**Amstelveen, The Netherlands**  
**22-23 September 2008**

## 1 Call to Order - Welcome - Introductions, Roll Call of Experts, Announcements

The meeting was called to order at 9:00 a.m. by Tim Donahue, Chair. The following technical experts were in attendance and introduced themselves:

1. Tim Donahue, Chair	Eastman Kodak Company	USA
2. Craig Benson	Adobe	USA
3. Dov Isaacs	Adobe	USA
4. Craig Revie	Fujifilm	UK
5. David Williams	Ricoh	USA
6. Rainer Prosi	Heidelberg	Germany
7. Hitoshi Urabe	Fujifilm	Japan
8. Makoto Matsuki	NTT Quaris	Japan
9. Olaf Drümmer	Callas Software	Germany
10. Reuven Ackner	EFI	USA
11. Paul Jones	Teclyn bv	Netherlands
12. Harry Raaphorst	Pascal Solutions bv	Netherlands
13. Leonard Rosenthol	Adobe	USA
14. Martin Bailey	Global Graphics	UK
15. Andy Kraushaar	Fogra	Germany
16. Bruno Mortara	ABTF	Brazil
17. Supree Thongpetch	Colour Doctor Co, Ltd.	Thailand
18. Santi Songsermsawas	Colour Group of Thailand	Thailand
19. Luci Wahrmann	Kodak	USA (Israel)
20. Joanna Cheng	Hong Kong Design Inst.	China
21. Christoph Oeters	Sofha	Germany
22. Hennes Leuchter	Sofha	Germany
23. Roland Thees	IFRA	Germany
24. Masayuki Nakajima	Tokyo Inst. Of Technology	Japan

The attendance sheet is WG2/TF3 N 067.

## 2. Review and approve agenda (WG2/TF3 N 087)

The committee reviewed the draft agenda N 087, which had been distributed prior to the meeting.

The following amendments were made to the agenda through the addition of the following additional agenda items:

- As agreed to in the last TF3 teleconference: *Discussion of the general need for features enabling raster caching of re-used elements* (e.g. *GTS\_GroupName*, *GTS\_CommonToSegment*, etc).
  - Should they be included in this version of the standard?
  - Could they be part of a future version?
  - Rosenthol noted that he has to leave at midday on Tues and asked that the caching discussion take place before then.

- As agreed to in the last TF3 teleconference: *discuss relevance of PDF/VT features to ISO 32000-2*. In particular the current DPart[Root] structure now defined in the PDF/VT draft.
- Time permitting: *preview of multiple output intents proposal to ISO 32000-2*.
- (Jones) *Proposal to aid adoption of PDF/VT*:
  - Can everything in PDF/VT be added using pdfmarks in Acrobat Distiller? Suggested that it may provide an easy migration path from PS-based workflows.
    - Rosenthol pointed out that you cannot currently make and validate PDF/X-4 files that way.
    - Agreed to consider this agenda item near the end to review this in the light of other discussions.
- (Hennes) raised the Paris decision that VT does not need to be stream-able and asked that the decision be revisited. Benson pointed out that Paris had suggested that a ‘chunking’ workflow addresses the need. It was agreed to add an agenda item for *discussion of streaming support for PDF/VT*.
  - Include consideration for a possible conformance level that enables synchronous transfer and RIPing of PDF/VT (so called streaming).

The agenda was approved as amended.

### **3. Approve minutes of the April 2008 meeting in Paris (N073)**

Donahue noted that there have been many electronic meetings between the time of the April Paris meeting and now. All minutes documents of those meetings have been distributed to the committee and all action items addressed in and resulting from those meetings have been tracked in these minutes documents.

The committee reviewed the minutes of the April 2008 Paris meeting, which had been distributed by e-mail. There being no corrections, the minutes were approved as distributed.

Donahue noted that the minutes of the electronic meeting held just prior to this meeting had not yet been distributed to the committee via email and that they will be made available very soon.

### **4. Review of open action items (N 073 and N 103)**

**08-01** – Closed.

**08-02** – Closed; will not be submitted to TC171. New action will be required to submit DPart structures.

**08-03** – Closed – included in draft 8

**08-04**: Open - Isaacs will prepare draft informative text relating to the decision not to permit caching/reusing objects to be included in the draft document.

- Isaacs reported this action item should be completed by Tuesday (second day of this meeting)

**08-05** – Closed – submission to ISO TC171 through Germany from Drümmer.

**08-08-28:1** – Bailey/Rosenthal: Define technical requirements as to what should constitute a graphics state independent XObject sans the referencing CTM when the proposed XObject /GTS\_Encapsulated=true.

- Closed; merged into action item 08-09-03.

**08-09-03** - Open - Rosenthal: propose a complete definition of technical requirements that defines an Encapsulated XObject for the next meeting.

- Open. Topic is on the agenda and may be closed during discussion at this meeting.

**08-09-08:1** – Open - Donahue, Wharman & Ackner: Provide examples of use cases for GTS\_GroupName, GTS\_CommonToSegment & XMP InstanceID to carry sufficient information to enable inter-job caching.

- Donahue reported that an off-line teleconference was recently held to discuss this action item.
- This item may be closed by discussion of WD8

**08-09-08:2** – Open - Donahue: proposal to add new Boolean into DPart structure to state that all page boxes in a branch of the tree are equivalent (from agenda item of teleconference held 2008-09-08).

## **5. Review of WD8 of ISO 16612-2 (N 099)**

The review was started at section 6.6 of Draft #8 (N 099). See tracked changes and comments in new working draft (WD 9) for details of all changes to WD 8 made in this meeting.

Rosenthal asked that the use of “PDF/VT reader” be reviewed for consistency, e.g. both “PDF/VT conforming reader” and “PDF/VT reader” are currently used. The document editor (Donahue) acknowledged this and will update the document to use the term “conforming PDF/VT reader” consistently.

Donahue noted that the definition and requirements for interactive/rendering/general readers need to be added to the respective sections of the document where discussed.

Donahue will make such editorial changes as he prepares the next draft.

### ***Discussion of PDF/VT file segments, collections & file sets:***

Drümmer requested that the term “instance identifier” be clearly defined (see last paragraph of 6.6.1). It was agreed that a definition for this term will be added to the document.

Before getting into the details of XObject reuse in the current draft, it was decided that the added agenda item to first determine the necessity of reused XObject hinting be discussed.

### ***Discussion of the need for technical requirements that enable different levels of caching in an optimized consumer***

Donahue recommended that any XObject hinting we decide to include in PDF/VT useful to an optimized consumer (e.g. for caching) should not dictate a particular caching strategy or implementation approach.

Bailey agreed, but suggested that different strategies do require different metadata; therefore, we need to identify the *important strategies* so that we can be sure that we have enabled those.

Isaacs made it clear that he doesn't see any need for hints beyond the "single use" Boolean already discussed. He suggested that history shows that system-level cache controls work better than application-specific ones. Isaacs cited the attempts to improve virtual memory performance by OS vendors who implemented various predictive caching strategies, etc..., and how such strategies failed in light of the generally unpredictable pattern of memory usage by applications.

Drümmer suggested that PDF/VT consumers should not be required to act on XObject reuse hints that may be present in the PDF/VT data. It was felt we should not worry too much about creators using hints incorrectly because their use is testable. Without such hints there is very little added value in PDF/VT over and above PDF. If some developers feel that their systems can take advantage of more detail in hints, should we not support such hints to support them?

Jones asked if an implementation would be hurt if hinting is included? It was felt this could lead to requirements to make different PDF/VT for different readers.

It was noted that this is what happened with PPML because they did not define how hints could/should be included from day one. It was felt it would be useful to have normative statements requiring that hinting keys must be correct if they are present. However, it should not be required to either write or read these keys.

It was noted that all such hinting keys are currently optional (for a conforming reader and writer) and that we need to discuss whether or not the hints, if used, are normative or informative.

One concern raised was the result of having different creators/consumers applying different meanings to the same hint.

Donahue suggested that all such semantics be normatively defined in the standard and be unambiguous and testable for exactly that reason.

Some felt that PDF/VT will only succeed outside of transparent jobs if it can be consumed as fast as current systems supporting other VDP formats (and faster than current baseline PDF RIPs). If we need to require that every XObject start with a set series of commands then that is acceptable.

Drümmer noted that we have representatives here from companies who have been dealing with this for some time; we should listen to what they need.

Bailey stated that he would like to identify which cache strategies should be enabled; for example, he knows of caching COS objects, display list and final raster.

Christophe indicated that he would like to look at the caching strategy from the perspective of the "lifetime" of usage of re-used objects.

Jones suggested three levels of *lifetime* be considered:

- single use,
- used across many jobs (or “within one job”),
- used “within one file”.

He agreed that these are only to be treated as hints, and are not instructions to the consumer.

Donahue noted that these three levels are in keeping with the currently proposed set of XObject hints such as GTS\_SingleUse, GTS\_CommonToSegment, etc., where the particular caching strategy is left to the reader implementation to decide based on the presence of these hints.

It was noted that all of this is dependent on out-of-band workflow information and decisions that are not fully represented within the PDF/VT file itself. This led to some discussion about whether such hint data should come from the PDF/VT file or be contained in associated information, such as a job ticket. It was agreed that this XObject hint information is only content level metadata, not a processing instruction such as would be specified in a job ticket. Given that such information is known by the writer application and is not control information, it was agreed that it is reasonable for the PDF/VT file to carry it. It was noted that a job ticket may, of course, override or enhance this information, depending on application or workflow requirements, etc.

After this discussion to determine the need for XObject level hinting to assist in a reader optimization of XObject caching, it was agreed that we would enable XObject hints in PDF/VT in this version of the standard. For each hint suggested it was agreed that it must enable the suggested three levels of cache strategy. Each suggested hint must therefore be reviewed on its merits.

Prosi suggested that we should look at hints in the context of the whole *job*, and assume that the consumer will know enough about the fact that it may be processing a subset to do something sensible with hints that were created for the whole.

Williams suggested that a reprint should only require adjusting the job ticket, and not changing the PDF/VT file(s).

### ***Review different object lifetimes to be supported***

The notion of a "job" was discussed, as well as that of a collection of related PDF/VT files that make up one or more related jobs, given that the lifetime of an XObject may span multiple job contexts, etc. A *lifetime hint* could be an XObject marker that indicates simply ‘this is the kind of content that is likely to be reused’ and would specify the scope of the lifetime of its reuse.

The following terminology was defined to minimize confusion (Note: these terms should be added to the document):

- *PDF/VT file set* - a PDF/VT-2 file and all dependent PDF files.
- *PDF/VT file collection* - is a collection of one or more PDF/VT-1 files and/or PDF/VT file sets.
- *PDF/VT Job* - is a unit of work that includes use of one PDF/VT file collection.

There was discussion regarding whether there is value in assessing multiple classes of lifetimes larger than a single file collection spanning multiple jobs. It was concluded that there is no value in doing so.

The following table was developed:

Life time	Hints proposed
Single use	?
Within a single PDF/VT-1 file or a PDF/VT file set	?
Within a single PDF/VT job	?
Across multiple PDF/VT jobs	?

The XObject usage hint key *GTS\_LifeTime* was proposed which would take a value of a closed enumerated set of values that indicates a lifetime scope of reuse of the XObject.

The committee discussed possible values for a new *GTS\_LifeTime* key, *SingleUse*, which inherits its meaning from the currently proposed *GTS\_SingleUse* Boolean key. There was some discussion as to whether XObject dictionaries referenced via the content stream of a form XObject rather than directly from a Page's content stream should be tagged with reuse hints or not. As a result of discussion, it was agreed to remove the *GTS\_SingleUse* key currently in the document.

Jones drew comparisons with PPML including the full set of CTMs that will be used with each mark so that a cache could be pre-filled, but recommended that we do not do that for PDF/VT.

Drümmer suggested that when we get to encapsulated XObjects there should be a separate copy of the XObject source if the form is used at more than one scaling.

Jones felt that *SingleUse* should be used if the XObject is only referenced once from one page (this is not limited to within a single file set).

Other values proposed in the discussion (all are of type *Name*):

- *MultipleUseInFile* - XObject is known to be referenced from page(s) more than one time within the containing PDF/VT-1 or PDF /VT-2 file.
- *MultipleUsePerJob* - XObject is known to be referenced from page(s) more than one time within the one or more PDF/VT-1 and/or PDF /VT-2 files used in the job (e.g. used for segmented jobs).
- *MultipleUseAcrossJobs* - XObject is known to be referenced from page(s) more than one time within the one or more PDF/VT-1 and/or PDF /VT-2 files used in multiple jobs (e.g. used for related jobs or for frequent repeat jobs).

Noted that this is similar to the semantics of PPML Global scoped occurrences which are expected to be stored persistently in the consumer.

- *Unknown* (default)

The following shows the *GTS\_LifeTime* proposal in the context of the completed table:

<b>XObject Life time</b>	<b>GTS_LifeTime hint</b>
Single use	<i>SingleUse</i>
Within a single PDF/VT-1 file or a PDF/VT file set	<i>MultipleUseInFile</i>
Within a single PDF/VT job	<i>MultipleUsePerJob</i>
Across multiple PDF/VT jobs	<i>MultipleUseAcrossJobs</i>

In the case of reference XObjects used in a PDF/VT-2 file set, the *GTS\_LifeTime* key should only be present in the reference XObjects defined in the PDF/VT-2 conforming file, *not* in the referenced PDF/X file.

It was agreed that the ‘highest’ lifetime level value of the *GTS\_LifeTime* key should be used if more than one applies.

**Proposal to be discussed:** Both Bailey and Drümmer would like shorter values for the *GTS\_LifeTime* key (possibly even as short as an integer value) given that this key may appear frequently in a file (e.g. ½ million times in a single file). It was suggested the committee consider shortening the key name as well (e.g. *GTS\_LT*).

The committee reviewed the XObject hint keys currently defined in the WD8 document affected by *GTS\_LifeTime*, as follows:

***GTS\_CommonToSegment:***

Delete this key currently defined in the document since it has been subsumed into the *GTS\_LifeTime* (*MultipleUseAcrossJobs* or *MultipleUsePerJob*) key.

***GTS\_GroupName***

Optional XObject key in the current draft is used to assign reused XObjects to a named class to enable labelling of persistent cache contexts for later out-of-band cache management (e.g. flushing, querying, viewing, etc.). It was noted that a job ticket or operator console option on the consumer implementation can use this mechanism. It was agreed that this attribute should be supported in PDF/VT and was renamed to ***GTS\_LifeTimeContextName***. It was agreed that the Type of this key should be a test string.

Donahue noted that the proposed name of this key should be revisited later, since its intended purpose of attaching an XObject to a grouping context is somewhat independent of the discussed notion of LifeTime. For example, XObjects of different lifetimes may belong to the same group.

Drümmer agreed that ensuring a single interpretation of each key is vital. If different tools use the same key for different things then it is useless and the data should be carried in private extensions. No specific requirement that this attribute be present, even if *GTS\_LifeTime=MultipleUseAcrossJobs*. It is expect that it will typically be used when XObject's *GTS\_LifeTime* is *MultipleUsePerJob* or *MultipleUseAcrossJobs*.

### ***Discussion of XObject identification***

Donahue noted that in discussion in a previous TF3 teleconference it was agreed that requiring reused XObject dictionary definitions be uniquely identified within the scope of their use is beyond a single file. For example, in the context of the current proposal. All copies of the XObject that occur among multiple PDF/VT files of a job context having a *GTS\_LifeTime* value of *MultipleUsePerJob* should have the same unique identifier. This allows such XObjects to be more easily identified by a cache optimized PDF/VT reader as being recurrent at the source level. It was noted that the current proposal for XObject identification is the use of *XMP::InstanceID* in the XObject definition.

Rosenthal asked if we are trying to uniquely identify the source data, or if the identifier part of the selection criteria is for identifying a cache instance (e.g. part of the discriminator ID of the cached object). He noted that *XMP::InstanceID* is a pre-existing construct that addresses unique identification of the source data.

It was noted that the cache discriminator may involve use of this identifier as well as other information such as a hash value based on a page content stream's referencing graphic state context, etc.

Drümmer indicated that he would prefer to *not* use XMP encoding of metadata for identifying an XObject. He was concerned that opening an XMP reader during parsing to access just a single key will cause an unnecessary performance hit, especially as XMP gets used more and more where the consumer would need to wade through a lot of extra (possibly unrelated) XMP information to find the InstanceID. For example, if many XMP entries are defined in the XObject dictionary, all of the XMP XML data would need to be parsed even if the reader is only looking for the InstanceID (which may not even be there).

Rosenthal agreed that given this performance argument, it is fine not to use *XMP::InstanceID* for identifying these XObjects and to instead use a COS key. However the COS key should not be used in the identification of the original source content asset itself (such as TIFF, JPEG, PDF identified using XMP, etc.).

As a result of discussion it was agreed to add a COS object key, *GTS\_InstanceID*, to the PDF/VT XObject dictionary. The value should be a globally unique identifier (GUID) encoded as a string. If two XObjects (in the same or different files) have the same *GTS\_InstanceID* value, then the consumer can assume that the XObject dictionary, its stream and any other referenced resources are equivalent. It was noted that the technical requirements as to what defines XObject equivalence has to be defined in the standard and that this cannot simply be a byte-for-byte comparison of the XObject dictionary.

It was pointed out that the acronym COS is not defined in the PDF specification and therefore should not be used in the PDF/VT standard itself.

**Action Item 08-09:** Rosenthal will propose an algorithm for XObject comparison that validates their equivalence.

Drümmer suggested the technical requirement that if both the *GTS\_InstanceID* and *XMP::InstanceID* are present in an XObject definition, then both *should* have the same value.

There was considerable discussion of this suggestion, resulting in agreement that requiring identifiers to match identically is not necessary.

### *Discussion of Encapsulated XObjects - GTS\_Encapsulated*

Bailey set out the objectives to allow a PDF/VT creator to identify an XObject as worthy of being cached by an optimized PDF/VT reader, such as a digital printing system's digital front end (DFE), at the raster level.

Drümmer identified what he thought to be a case where the content of an XObject that is fully opaque fully covers its bounding box, with no overprint or transparency.

Bailey suggested we must enable some way of caching transparent data, otherwise PDF/VT will fail because that's the key reason for using PDF for VDP.

Jones pointed out that PDF does have some other advantages such as preview, pre-flight etc.

Drümmer felt that it must be at least as fast as other formats for non-transparency and faster than baseline PDF for transparency.

### **Discussion of what affects cachability of an XObject by an optimized consumer:**

- Graphics state (gstate) context:

Drümmer felt this can be addressed by requiring the XObject to have all gstate values explicitly set. Rosenthol noted the importance of making sure that a form XObject sets *all* gstate variables. Images don't do that; they must explicitly set rendering intent in the image dictionary.

As a result of discussion it was **agreed** that all gstate variables should be explicitly set in an Encapsulated XObject.

**To Do:** Need to work out the text of the document that defines this requirement, including that the gstate call must be first entry in the XObject's contents stream.

- Default colour spaces:

Drümmer felt this can be addressed by requiring that the XObject have its own explicitly set default colour space. Bailey disagreed, suggesting that it is only necessary to ensure that there is a Resources dictionary in the XObject definition ( according to ISO 32000). Rosenthol suggested we should check that against current implementations, which may do something different.

Drümmer felt this doesn't work for image XObjects, and that the easiest way to circumvent that is to prohibit use of default colour spaces. Jones noted that would require producers to re-write PDF data in many cases, and producer applications are unlikely to do that work, even though they could.

**Proposal:** If a contents stream uses an encapsulated image XObject in a device colour space, then the current resources dictionary for the image XObject shall not contain a matching default colour space.

Revie suggested an alternative of only allowing form XObjects to be encapsulated. It is always possible to define a form XObject to carry an image. It was felt this might be slightly cleaner (the XObject doesn't carry data describing its calling context), but a more heavyweight solution.

Bailey suggested perhaps we should not try to block out default colour here, especially for image XObjects. It's not something that would stop a consumer making a cache; multiple default colour spaces would simply result in one keeping multiple caches from the same XObject.

It was **agreed** that Encapsulation only applies to a form XObject (including a reference XObject); it shall not be used for an image XObject. A consumer doesn't need the hint to know whether or not it should cache an image XObject.

- *Overprint:*  
Bailey felt this can be addressed by assuming the consumer will create one mask per colorant. Benson noted this gets a little tricky if process color equivalents are used for spot colors, but is really not a big problem.)
- *Optional content:*  
Rosenthal noted that a form, reference or group XObject can have an OC key, or can have embedded marked content operators in its content stream.
  - Any form XObject with the explicit key for the optional content operator shall not be marked as encapsulated.

Drümmer agreed in the case of the marked content operators in the contents stream, but felt the OC key should be allowed since it just defines whether or not it should be included as cached content.

Benson felt optional content status should just be one more selector in your cache identification hash.

Drümmer noted that to determine whether an XObject's content stream includes an OC operator, one needs to parse the contents stream. Benson stated that if you are going to cache anyway, then you would tag the cache you have just created with the OC status (or 'any' if it has no OC commands in it). If you hit the same XObject again then you can check to see if you have a cache for the current context and either re-render for the new status or select an existing one.

Prosi noted that optional content in VDP is very important, e.g. to enable viewing and printing of content used in hybrid (e.g. offset printed shell imprinted with digitally printed variable content) workflows.

It was **agreed** that if there are optional content marked content operators present in the contents stream of an encapsulated reference or form XObject, or that of form XObjects referenced by it, that XObject shall not be marked as encapsulated. The OC key *is* allowed in an encapsulated XObject.

- *Transparency*

Rosenthal proposed that any XObject marked as encapsulated that uses transparency shall be marked as a group XObject.

It was **agreed** that a transparency group that is isolated and defines a blend colour space may be marked as encapsulated.

Rosenthal noted if no CS is defined in the XObject then it cannot be encapsulated. Bailey suggested it could be used as another selector for which cache. Drümmer stated it is good practice to include a CS in an isolated transparency group anyway.

It was **agreed** that an isolated trans group with no CS shall not be tagged as encapsulated

Rosenthal suggested that if it is non-isolated then it should not be tagged as encapsulated. Bailey questioned whether that prohibits any common use cases for transparency being marked as encapsulated. Rosenthal noted that drop shadows using Multiply (or Overlay, or Normal) can all be represented in isolated transparency groups. KnockOut does not impact encapsulation. Jones noted that is what PODi had concluded, as well).

It was **agreed** that a non-isolated transparency group shall not be marked as encapsulated.

***Discussion of Japanese proposal to include a reused XObject list and corresponding reference count:***

Donahue initiated review of the proposed *XObj* and *XobjRefCout* keys of the DSM dictionary defined in Table B.1 of the document. It was noted that these keys are based on the TF3 response to the Japanese proposal for reused XObject information including reference counts as summarized in the minutes of the electronic meeting held 2008-08-11 (N 092). The Japanese later responded in favor of this proposal.

Jones pointed out that the Japanese request for such a complete list of reuse information defined in the file up-front is unprecedented. The only approach he is aware of that is closest to this proposal is the weighting numbers in PPML, which he understands most consumers ignore.

Ackner agreed and pointed out that in PPML most of the producers don't use the weight field. In addition, since the definition of weight is not exact (very subjective), different producers can give you the same weight but their real usage may be different. Because of this, the weight definition is not so useful in PPML and EFI doesn't use it. However, with XObject he would like to get an exact usage count as proposed, which is well defined and reliable and can easily be used to optimize the cache. This case really demonstrates that any hinting information that we need should be well defined and exact otherwise will not be useful.

Matsuki summarized that the Japanese want to maximize RIPing efficiency, and cannot rely only on hints (but didn't explain why).

Jones said he sees how a list of all reused XObjects that may be easily cachable would be useful.

Bailey pointed out that this may clash with streamability, which will be discussed later on.

Rosenthal stated that values may be irrelevant if the PDF/VT file is split and run on multiple machines, and might lead to a DFE caching things that are not relevant for the part of the run that it is processing.

Ackner suggested that such XObject reference counts are useful if present to guide a consumer cache clearance strategy.

Donahue directed discussion to the XObj array in DPM. Rosenthal suggested that such an approach is very inefficient because loading an array implies loading all of the objects in it. Prosi, Bailey and others disagreed and pointed out that it doesn't imply loading of all XObjects. Prosi asked of this would be put just at the leaves of the DPart tree, or also at higher levels. Drümmer questioned if this does not only list cacheable XObjects, and felt this is up to the creator. Bailey disagreed, stating that for the list to be useful to him, he needs to know what it represents.

**Proposal:** We should define that all XObjects referenced from the XObj array be included or all encapsulated.

- Donahue has added text in the current draft (N 099) describing these.
- A change was made to the XObj key definition where its Type was updated to be an Array of PDF Object references. The original term COS was replaced with PDF.
- Matsuki will take this back to Japan for review and comment.

#### *Discussion of streaming support for PDF/VT*

Why streaming?

- Adoption of PDF/VT for transactional applications requires some level of correspondence with print data stream languages such as IPDS.

Basic requirements to be addressed:

- Processing/printing while data is received and/or being generated.
- Continuous generation while printing without a priori knowledge of when the run ends.

Three general use cases of streaming discussed:

1. Streamed generation. Assumes a file is being generated and an Xref table may be added to the end of the file.
2. As a Consumer being receiving data, it processes that data.
  - a. Noted that a feature known as linearization is defined in PDF that is designed for this purpose. It is designed for use "over the WWW". Requires Page 1 of the PDF file to be completely defined at the head of the file. Xref for the first page is at beginning AND another Xref table is at the end of file.

3. Simultaneous continuous generation and continuous consumption.
  - a. Data is never stored as a file and typically piped through an I/O channel.
  - b. Not a capability currently available in PDF.
  - c. *PDF IS* specification already exists (IS=Internet Streaming) but is not sufficient according to Rosenthol.

Rosenthol commented that 1 above is easy to accomplish with PDF as it stands. Could not be linearized if streamed, and 2 above can be addressed with linearization. It was suggested that a *byte serving* provider would be the easiest way to handle this in an implementation using current linearization tools, but the consumer can do its own byte server as a front end. To be efficient that would require that the linearization be created in specific ways to start with.

Benson stated he would not expect any significant re-ordering of pages during processing in a streaming system because it's assumed to be more or less a closed workflow system. Oeters and Jones agreed.

Donahue asked about the key value of streaming over a chunked (segmented data stream) workflow? Oeters noted that current UIs show each file as "individual jobs", so a chunking workflow with a sequence of multiple files would be confusing to the user.

Rosenthol stated that PDF currently has no way to address 3 above. Anything to enable that workflow would need to be done in PDF under ISO 32000.

Jones asked if it is possible to read a PDF stream in the absence of an Xref? Drümmer and Bailey replied that it is, but it is not 100% reliable. It was noted that Acrobat can, for example, repair a PDF file by generating a new Xref if the original is corrupted; however, it is not always successful.

Drümmer suggested the committee should consider how to add a reliable method for streaming and *then* figure out whether that method needs to be incorporated into ISO 32000 instead of making it an extension to PDF in VT.

Donahue challenged the requirement for streaming over and above chunking. Oeters and Jones identified the requirement that for chunking there needs to be a positive association of each chunk with the next. Donahue pointed out that basic chunk sequencing information was in an earlier PDF/VT draft and encoded as keys in `DPartRoot` for this purpose.

Donahue also asked whether or not first-page-out time was fundamentally an important aspect of a streaming (3) printing system implementation, even given the size/length of some of these streaming jobs, which in some cases may make first-page-out a low priority. He suggested that if first-page-out time is important then the first chunks of the job can be smaller in order to assist with first-page-out processing time.

Oeters asked if incremental update (discussed as appending pages to a PDF file) could be used to extend a stream. Bailey asked whether JDF could be used to manage the chunks. Jones felt it could not be used in a streaming workflow. The creator may know the number of recipients, but doesn't necessarily know the number of pages, and therefore doesn't have up-front knowledge of the number of chunks. In streaming workflows the pages conventionally do not require imposition on the back end.

The implication is that the pages are sheet surface definitions that are consumed and output by the printer in the order they are defined by the producer.

Donahue suggested this is simply a JDF RunList resource abstraction issue. For example, the JDF RunList/LayoutElement could include a mechanism abstracting the multiple file stream sequence as a single indeterminate length page stream thus not requiring exhaustive Run partitioning of the RunList resource to connect to each segment file. The RunList could also be defined as a pipe resource.

Drümmer suggested three options be considered:

1. One big PDF/VT file that can be streamed from creator to consumer.
2. A sequence of PDF/VT files delivered to the consumer as chunks.
  - a. Jones suggested that unencoded PDFs be included in a MIME-like package to associate chunks together and provide a chunk length.
3. Job-ticket driven chunking.

### *Discussion*

How reused object caches get carried over, or persist in the consumer's memory between chunks.  
How the PDF/X-4 files referenced by multiple PDF/VT-2 file sets of a chunk sequence are interleaved among the PDF/VT files.

It was noted that archiving of real print data (as opposed to regenerated data) is somewhat easier with chunks rather than with single monolithic files. It was felt a need exists for these archives in cases such as call center support where there may be a legal requirement that the actual print data be archived.

It was suggested that the sequence of PDF/VT chunks and PDF/VT-2 referenced PDF/X-4 files and other referenced resources could be packaged in a stream oriented format.

The concept of a messaging/wrapping structure of a logical stream comprising a sequence of chunks was discussed.

Oeters suggested there should be an indication that “there will be another chunk”.

Bailey asked how a short separate job being interleaved between chunks is handled; as well as tagging a PDF/X-4 re-used content file that’s part of the same collection.

Given the breadth of requirement for these streaming concepts it was decided that an action item is necessary so a small group can work out a proposal.

**Action Item 08-10: Drümmer, Oeters, Rosenthol** will look at metadata to enable the connection between chunks and offer a proposal.

**Action Item 08-11: Jones, Rosenthol, Oeters** will develop a multiple chunk packaging specification proposal that can accommodate efficient chunk granularity streamed consumption by a reader (e.g. ASN.1 as suggested by Jones).

***Discussion of additional metadata from PDF/VT creators***

Jones noted for a majority of jobs, most VDP PDL creation tools already provide information about the interaction of content objects on a page, such as whether any PDF XObjects intersect with any other XObject; either behind or on top of it. Availability of such hint information in the PDL means that the consumer can, for example, easily determine if masks need to be created for cached rendered objects, etc.

Donahue suggested adding an application note stating that bounding boxes be as small as possible for all objects (e.g. minimum rectangular extent of marked content). Jones pointed out that this kind of key is used in (EFI's) *FreeForm*.

It was **agreed** to add a new XObject key with the name ***GTS\_ZOrder*** having the type Name. Allowed values shall include: *Unknown* (default) | *Top* | *Bottom* | *TopAndBottom*. For use in image, form, & reference XObjects.

[Editorial note: What about Group Xobjects?]

- *Top* specified when the XObject's bounding box is not overlaid by the bounding box of another XObject placed on the same page.
- *Bottom* specified when the XObject's bounding box does not overlay that of any other XObject placed on the same page.
- *TopAndBottom* specified when the XObject's bounding box does not interact with (overlay or underlay) the bounding box of any other XObject placed on the same page. For example, recurring content such as fixed terms and conditions text specified as an XObject that is the only XObject placed on a page.

A hint for black-only content was discussed and it was decided that an implementation to determine on the fly while PDF data is interpreted and rendered makes the most sense.

***Return to review of working draft 8 (N 099):***

**Section 6.5** – Document Part Metadata (see changes to draft document):

Jones asked why DPM is limited as a simple dictionary allowing only key value pairs and proposed that DPM be extended to support more complex structures.

**Action Item 08-12:** Jones will develop a proposal for more complex data structures within DPM, including dictionaries as data structuring tools.

Prosi suggested prohibiting arrays. Bailey felt if the data is application-specific and the consumer is not concerned about an XPath representation there is no need to prohibit arrays. Bailey proposed that all strings be read as text strings, but not apply any other restrictions.

It was **agreed** that indirect objects should be allowed in a DPM dictionary so that metadata definitions can be shared among multiple DPM dictionary instances.

Benson suggested that some data types should be prohibited (e.g. XObjects, streams etc).

It was **agreed** that such a restriction is not necessary.

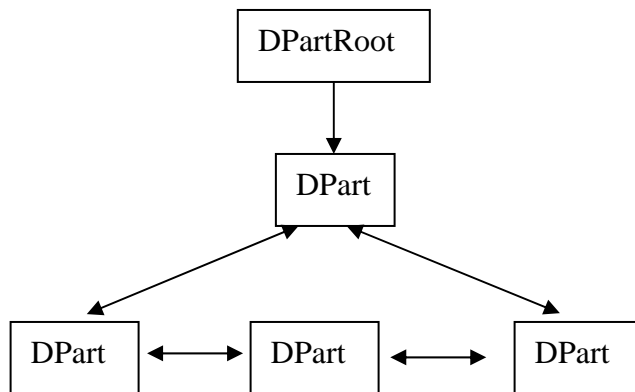
### ***Updates to document section 6.6 re-titled as Recurring Content***

Changes were made to this section based on earlier discussions and agreements on technical requirements for XObject hint information and XObject encapsulation. Refer to WD 9 for details of these changes.

Note: The editor to restructure the subsections of section 6.6.

### ***Review of DPart[Root] Struct requirements***

Prosi noted that the attributes of DPartRoot contain those of a DPart node and proposed removing the common attributes from DPartRoot and have DPartRoot reference a DPart node that is the root node of the following tree:



It was **agreed** to update Table 4 to reflect features of DPartRoot that were impacted by this change. Refer to the updated WD 9 for details of the changes.

### ***Discussion of proposal to clarify the DPM Copies key***

Prosi proposed the following definition of the value of the *Copies* key of Table B.1:

*(Optional) Value shall be interpreted as the number of copies to produce of the DPart node within which it is defined. If specified, the value shall be non-negative.*

*Note Copies is useful in situations where multiple copies of recipient records or parts thereof are requested, for instance in versioning jobs.*

There was a concern by many as to the use of this attribute, which was introduced to the document as DPM metadata in the Spring 2008 TF3 meeting in Paris.

Isaacs expressed a concern as to the expectations of the normative requirements of a reader in response to the presence of this attribute. It was noted that as DPM metadata may be ignored by a reader.

The question was then raised that perhaps it should not be optionally ignored by a reader as essentially it represents a short-hand way of specifying page repetitions referenced in lower level nodes, or within the DPart node itself that is intended to be processed by a consumer.

It was suggested that in the situation where repetitions of page groups are required for a job, the writer could output a sequence of multiple equivalent DPart nodes – one for each copy to be produced. It was noted that although this is a solution, it has the disadvantage of being much less concise than allowing use of the Copies attribute. It was also suggested that if this is a key that a reader cannot ignore, then it should be a key of the DPart node itself rather than being encoded as a DPM key.

**Proposal:** Since the intended purpose of the *Copies* key is as a short-hand way of describing repetitions of a DPart node, it was proposed that the key be renamed to *Repetitions*.

Isaacs noted that this particular key may have significant implication to such applications as viewers and suggested the committee take this proposal under advisement for now and that he will discuss this internally within Adobe.

**Action Item 08-13: All TF3 members:** take the DPM *Copies* or *Repetitions* key under advisement and think through the ramifications of its use in PDF/VT versioning jobs and in the context of a supporting workflow and use by various readers. Be prepared to discuss its use and requirements for a conforming reader.

It was pointed out that if this particular feature is used that it should be considered for ISO 32000.

#### ***Proposal to aid adoption of PDF/VT***

Jones withdrew this proposal in the light of other discussions.

#### ***Discuss relevance of PDF/VT features to ISO 32000-2***

Due to time constraints this was not discussed.

#### ***Preview of multiple output intents proposal to ISO 32000-2***

Due to time constraints this was not discussed.

## **6. PDF/VT Application notes**

It was proposed that information describing market positioning of PDF/VT should be covered in the application notes. The recommendation was made to add an informative Annex of the standard for positioning purposes.

It was **agreed** that, given the significant interest in PDF/VT worldwide, the standard provide an appropriate level of positioning and possibly best-practices information in an informative Annex.

It was noted that application notes can be provided without cost and are useful as a source of information for those interested in PDF/VT.

It was decided that this Annex should address best practices in the optimization and structuring of a PDF/VT file respective of certain use cases.

A PDF/VT informative Annex on Efficiency considerations was also discussed as a primary focus. The standard may point to other published documents such as an application note document.

It was noted that attention should be paid to beneficiaries of PDF/VT in terms of advantages of a common format and data exchange. It was thought this should be a major part of the position statement.

**For the next application note teleconference:** determine an outline for this informative annex for the standard for best practices and positioning.

It was **agreed** that a repository for PDF/VT example files should be created under the vPDF Yahoo group forum. Benson and Williams volunteered to manage this repository.

**Action item 08-14:** Benson will provide a sample PDF/VT conforming file based on the latest working draft document.

## 7. Requirements for next meeting

The next TC130 general meeting is scheduled for May 18, 2009 in Dallas/Fort Worth, Texas. As was proposed in the last teleconference, an interim face-to-face meeting of this task force should occur before this meeting in Texas.

The goal of WG2/TF3 as discussed is to get to CD ballot return in time for that meeting (CD balloting is 3 months). That could lead to all decisions made to move to DIS after May 2009.

The concern was raised that some members cannot justify attending additional TF3 face-to-face meetings (such as the meeting proposed for January 2009) before the TC130 meetings in Dallas. Therefore, Donahue suggested allowing those interested in attending, who cannot travel, be allowed to attend remotely via voice dial-in and web-based document sharing (e.g. Adobe connect) for the interim face-to-face meeting in January. It was agreed that remote access shall be a requirement for this meeting.

Given the current commitments of many participants to attend other standards meetings in Europe in the late January time frame, it was decided to plan the Europe meeting for January 19-21 (leading into GWG Gent) as many participants will already be there.

Drümmer indicated that Callas could likely host this meeting in Berlin, Germany. Alternatively, Adobe could host the same dates at Adobe in London or San Jose January 20-28 (after Lyra Palm Springs).

Prior to the January meeting, additional telecon meetings will be held as in the past, with the first two calls scheduled for Monday the 6<sup>th</sup> and 13<sup>th</sup> of October. Meeting notices and agendas will be emailed ahead of each meeting.

For the Spring TC130 meeting in Dallas Texas, WG2/TF3 will need 3 days, preferably Monday-Wednesday.

## **8. Adjournment**

The committee thanked Wilco de Groot and IGT, Koninklijke, Grafimedia and VIGC for their support of the meeting and lunch, IGT for the social event, and Eleni Tzatzalos and Leonie Kuijpes for their support and assistance with meeting-related needs.

Special thanks to extended Martin Bailey of Global Graphics and Craig Revie of FFEI for taking detailed notes of meeting discussions which have been incorporated into these minutes, along with the notes of the chairman of TF3 captured during the meeting.

There being no further business the meeting was adjourned.