BS ISO 16612-2:2010

BSI Standards Publication

Graphic technology — Variable data exchange

Part 2: Using PDF/X-4 and PDF/X-5 (PDF/ VT-1 and PDF/VT-2)

National foreword

This British Standard is the UK implementation of ISO 16612-2:2010.

The UK participation in its preparation was entrusted to Technical Committee PAI/43, Graphic technology.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

ISBN 978 0 580 66530 1

ICS 35.240.30; 37.100.99

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2010.

Amendments issued since publication

Date	Text affected

BS ISO 16612-2:2010

INTERNATIONAL STANDARD

ISO 16612-2

First edition
2010-08-15

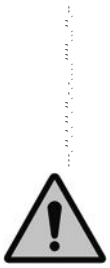
Graphic technology — Variable data exchange —

Part 2: Using PDF/X-4 and PDF/X-5 (PDF/VT-1 and PDF/VT-2)

Technologie graphique — Échange de données d'impression variables —

Partie 2: Utilisation de PDF/X-4 et PDF/X-5 (PDF/VT-1 et PDF/VT-2)

Reference number
ISO 16612-2:2010(E)



ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 16612-2 was prepared by Technical Committee ISO/TC 130, *Graphic technology.*

ISO 16612 consists of the following parts, under the general title *Graphic technology — Variable data exchange*:

— *Part 1: Using PPML 2.1 and PDF 1.4 (PPML/VDX-2005)*

— *Part 2: Using PDF/X-4 and PDF/X-5 (PDF/VT-1 and PDF/VT-2)*

Introduction

This part of ISO 16612 defines the PDF/VT document format which specifies methods for the use of the Portable Document Format (PDF) for the definition and exchange of all content elements and supporting metadata necessary for printing tasks involving variable or transactional document content. It allows the specification of document structure, document layout, content data, and interaction of graphical elements in a graphics model that supports transparency.

PDF/VT is designed to enable variable document printing (VDP) in a variety of environments from desktop printers to digital production presses. This includes hybrid workflows involving both conventional and digital printing.

This part of ISO 16612 defines three conformance levels as follows.

— PDF/VT-1 for a complete single-file exchange. PDF/VT-1 requires all resources necessary for proper interpretation of the PDF data to be included within the conforming PDF file.

— PDF/VT-2 for multi-file exchange. PDF/VT-2 enables a conforming file to refer to an external ICC profile file and additional content defined in external PDF/X conforming files for use as page content.

— PDF/VT-2s for streamed delivery. PDF/VT-2s allows for processing (streaming) of multiple compound entities representing graphical content before the entire PDF/VT instance has been generated. A PDF/VT-2s stream is a MIME package that contains a sequence of one or more PDF/VT files and supporting resources.

The present trend towards the separation of variable document content creation from the details of print production workflow and printing device dependencies is evolving rapidly. The job definition format (JDF) job ticket specification being developed by the International Cooperation for the Integration of Processes in Prepress, Press, and Postpress Organization (CIP4) provides one means of specifying a print product and corresponding production process in a way that is independent of any particular graphical content format.

PDF/VT is intended to be workflow-architecture-neutral. PDF/VT has no provision for encoding workflow or device-specific control information. The aspects of device control, resource, and production management are outside the scope of this part of ISO 16612. In a production environment, PDF/VT relies on the use of JDF, or similar job ticket formats, to define a print product and the corresponding production requirements. The primary focus of PDF/VT is on the exchange of content between businesses or within an integrated environment that produces variable document printing.

Graphics design applications continue to evolve with greater capability and increased sophistication of the graphical content and design effects based on a graphics model that supports transparency. This graphics model is required to support features such as drop shadows and colour blending effects that are associated with the interaction of transparent content objects. These capabilities are used in the creation of one-to-one customer communication print applications including direct marketing documents, transactional documents, and trans-promotional documents.

This part of ISO 16612, referred to as PDF/VT, includes support for the PDF 1.6 imaging model which includes support for transparency. It builds on the PDF/X-4 and PDF/X-5 standards (defined in ISO 15930-7 and ISO 15930-8), which, in turn, reference PDF 1.6. Like ISO 16612-1, this part of ISO 16612 guarantees portability of conforming VDP content and metadata across conforming digital printing systems. It is focused on defining the content data and metadata necessary to support efficient workflow manipulation and processing based on the use of JDF or similar job ticket formats. More specifically, the job ticket is expected to define the production requirements and draw upon PDF/VT for its content and metadata resources.

This approach supports the fundamental requirements of portability, device and workflow independence and guaranteed communication of the intended colour reproduction. The native constructs within PDF allow products to provide functionality equivalent to that of PPML workflows as used in ISO 16612-1. This part of ISO 16612 uses a single homogeneous format while adding support for a graphics model that supports transparency, including support for interacting transparent objects.

This part of ISO 16612 is based on PDF/X-4 and PDF/X-5 allowing users to continue to use, and solution vendors to build on, existing PDF-based workflow tools and job-ticket-based VDP workflow architecture similar to PPML/VDX, with the added capability of the full graphics model of PDF 1.6.

PDF/VT supports the use of graphical object definitions as a method of specifying graphical content data only once in a PDF/VT file independent of the number of times it is referenced in the file. This approach serves to reduce the file size of a PDF/VT instance and allows implementers of conforming readers to employ various processing optimization strategies. Within the context of PDF/X-4, these graphical objects are specified as image, form and transparency group **XObject**s. Use of PDF/X-5g and PDF/X-5pg allows for the use of reference **XObject**s where the **XObject's** content stream is that of a page of a referenced PDF file.

While strongly recommended, there is nothing in this part of ISO 16612 that enforces the creation of PDF/VT files that make efficient use of **XObject**s.

XObjects referenced multiple times from various content streams can be tagged with hint information that aids the conforming reader in its determination of **XObject** reuse. These hints include an explicit lifetime scope to indicate the context within which the **XObject** is known to be referenced multiple times, such as within the current file, across multiple files of a PDF/VT instance or across PDF/VT instances. An **XObject** can also be tagged with an identifier to assist a conforming reader in the identification and management of such recurring definitions, such as in the case where it is known to occur across multiple PDF/VT files.

An **XObject** can also be identified as an encapsulated **XObject** if its definition has a limited and well-defined interaction with the current graphic state when invoked. This explicit hint serves to assist a conforming reader in its optimization strategy.

This part of ISO 16612 enables an exchange of content where no additional technical information is required to be communicated between sender and receiver for the purpose of describing the appearance of page content.

Application notes for this part of ISO 16612 are available to provide assistance to developers and users of applications designed to conform to this part of ISO 16612 and can be found at http://www.npes.org/Standards/toolspdfvt.html.

A reader conforming to this part of ISO 16612 is not required to support documents conforming to ISO 16612-1, which is based on the use of PDF and PPML.

Graphic technology — Variable data exchange —

Part 2: Using PDF/X-4 and PDF/X-5 (PDF/VT-1 and PDF/VT-2)

1 Scope

This part of ISO 16612 defines the PDF/VT document format and methods to enable reliable document exchange for variable data and transactional (VT) printing. It uses the Portable Document Format (PDF) Version 1.6, as restricted by PDF/X-4 and PDF/X-5, for the representation of such documents. It allows the specification of document structure and layout, content data, and interaction of graphical objects in a graphics model that supports transparency and both device-dependent and device-independent colour spaces. All elements are either included or provision is made for unique identification of externally supplied graphical content or ICC profiles.

PDF/VT is designed to enable variable data and transactional printing in a variety of environments from desktop printers to digital production presses. This includes hybrid workflows involving both conventional and digital printing.

This part of ISO 16612 does not provide for the specification and encoding of production and device control information but is constructed to enable its use with the CIP4 Job Document Format (JDF) or similar job ticket formats.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 15930-7, *Graphic technology — Prepress digital data exchange using PDF — Part 7: Complete exchange of printing data (PDF/X-4) and partial exchange of printing data with external profile reference (PDF/X-4p) using PDF 1.6*

ISO 15930-8, *Graphic technology — Prepress digital data exchange using PDF — Part 8: Partial exchange of printing data using PDF 1.6 (PDF/X-5)*

Adobe PDF Reference, fifth edition, version 1.6., Adobe Systems Incorporated (ISBN 0-321-30474-8). Available from <http://www.npes.org/standards/toolspdfx.html>

Errata for the PDF Reference, fifth edition, version 1.6 dated 4 October 2006, Adobe Systems Incorporated. Available from <http://www.npes.org/standards/toolspdfx.html>

Extensible Markup Language (XML) 1.0 (Second Edition), 6 October 2000, World Wide Web Consortium, Available from <http://www.w3.org>

PDF Blend Modes: Addendum, 23 January 2006, Adobe Systems Incorporated. Available from <http://www.npes.org/standards/toolspdfx.html>

RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, November 1996. Available from <http://www.ietf.org/rfc/rfc2045.txt>

RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, November 1996. Available from <http://www.ietf.org/rfc/rfc2046.txt>

RFC 2047, *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text*, November 1996. Available from <http://www.ietf.org/rfc/rfc2047.txt>

RFC 2183, *Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field*, August 1997. Available from <http://www.ietf.org/rfc/rfc2183.txt>

RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, July 2005. Available from <http://www.ietf.org/rfc/rfc4122.txt>

XML Path Language (XPath), version 1.0, WC3 Recommendation, 16 November 1999. Available from <http://www.w3.org/tr/xpath>

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
compound entity
unit of work with all text, graphics and image elements prepared for final print reproduction and that might represent a single page for printing, a portion of a page or a combination of pages

3.1.2
document
collection of related document parts

NOTE A document part can also be a document.

3.1.3
document part
set of related pages or related sets of pages, or both

EXAMPLE Chapter pages of a book or all sets of pages intended for a recipient.

3.1.4
document part hierarchy
hierarchical data structure that specifies the organization of document parts

3.1.5
document part metadata
DPM
metadata associated with a document part

3.1.6
editor
application that is both a reader and a writer capable of modifying a file

3.1.7
encapsulated XObject
XObject having a well-defined limited interaction with the current graphics state at the point of invocation

3.1.8
environment context
set of reusable content

3.1.9
graphical object
graphical content definition comprising text, vector graphics, image data or some combination thereof

3.1.10
ICC
International Color Consortium
industry association formed to develop standardized mechanisms for colour management

3.1.11
ICC profile
set of colorimetric transforms prepared in accordance with ICC.1 or ISO 15076-1

3.1.12
interactive reader
reader that requires or allows human interaction with the content and other objects contained in the document during the software's processing phase

NOTE A file viewing tool is an example of an interactive reader; a raster image processor is an example of a reader that is not interactive.

[ISO 19005-1:2005]

3.1.13
job definition
information that specifies the production requirements and workflow of a unit of work involving purposing PDF/VT content to one or more messaging channels

3.1.14
job ticket
electronic specification of print product or process control for print production, or both

3.1.15
PDF
Portable Document Format
file format defined in the *PDF Reference*

3.1.16
PDF/VT
document format and methods to enable reliable document exchange for variable data and transactional printing

3.1.17
PDF/VT chunk
conforming PDF/VT-1 file or PDF/VT-2 file set that is a member of a PDF/VT-2s stream

3.1.18
PDF/VT file
a conforming PDF/VT-1 or PDF/VT-2 file

3.1.19
PDF/VT file collection
one or more conforming PDF/VT-1 files or PDF/VT-2 file sets, or both

NOTE PDF/VT file collections can include files belonging to several PDF/VT instances.

3.1.20
PDF/VT instance
PDF/VT-1 file, PDF/VT-2 file set or PDF/VT-2s stream

3.1.21
PDF/VT-2 file set
single conforming PDF/VT-2 file and all its referenced PDF files and external ICC profile files

3.1.22
PDF/X-4
conformance level defined in ISO 15930-7

3.1.23
PDF/X-4p
conformance level defined in ISO 15930-7

3.1.24
PDF/X-5
set of conformance levels defined in ISO 15930-8

3.1.25
PDF/X-5g
conformance level defined in ISO 15930-8

3.1.26
PDF/X-5pg
conformance level defined in ISO 15930-8

3.1.27
print element
element intended for final print reproduction

NOTE For more information, see ISO 15930-7.

3.1.28
print product
outcome of the processing of a document through a print manufacturing process.

EXAMPLE A perfect bound book or a postcard.

3.1.29
product part
part of a print product

EXAMPLE The cover part of a saddle-stitched booklet.

3.1.30
reader
software application that is able to read and appropriately process files

[ISO 15930-7]

3.1.31
recipient record
collection of information related to the pages intended for a single recipient

3.1.32
writer
software application that is able to write files

[ISO 15930-7]

3.2 Abbreviated terms

CIP4 International Cooperation for the Integration of Processes in Prepress, Press, and Postpress Organization

DPM document part metadata

ICC International Color Consortium

JDF Job Definition Format as defined in the CIP4 JDF Specification

PDF Portable Document Format

4 Notations

PDF operators, PDF keywords, the names of keys in PDF dictionaries, and other predefined names are written in a bold sans serif font; for example, the key **Trapped**.

Operands of PDF operators or values of dictionary keys are written in an italic sans serif font; for example, the *False* value for the **Trapped** key.

The term “PDF/VT file” indicates requirements for files that comply with at least one of the conformance levels defined in this part of ISO 16612.

References to the “PDF Reference” are to the *Adobe PDF Reference*, fifth edition, Version 1.6, as modified by *Errata for the PDF Reference*, fifth edition, and by *PDF Blend Modes: Addendum*.

References to XML are to the Extensible Markup Language defined in *Extensible Markup Language (XML) 1.0 (Second Edition)*.

5 Conforming files and equipment

5.1 General

This part of ISO 16612 specifies the use of the PDF file format for the exchange of digital data representing the compound entities of one or more components intended for printing.

This part of ISO 16612, is based on ISO 15930, and defines three conformance levels based on ISO PDF/X standards as follows.

— *PDF/VT-1* requires all compound entities, metadata and resources used in representing graphical content to be defined within the PDF/VT-1 file as specified in PDF/X-4.

— *PDF/VT-2* allows one or more compound entities representing graphical content or ICC profiles required to render the file to be externally referenced according to the provisions specified in PDF/X-4p, PDF/X-5g and PDF/X-5pg.

— *PDF/VT-2s* allows for processing (streaming) of multiple compound entities representing graphical content before the entire PDF/VT instance has been generated.

PDF/VT conformance is identified in a PDF/VT file by use of the **pdfvtid:GTS_PDFVTVersion** property as specified in 6.3. Neither the version number in the header of a PDF file nor the value of the **Version** key in the **Catalog** of a PDF file shall be used in determining whether a file is in accordance with this part of ISO 16612.

A PDF/VT file shall conform to all requirements set out in Clause 6.

5.2 PDF/VT-1 conformance

A PDF/VT-1 conforming writer is a software application that is able to write PDF/VT-1 files in accordance with the conformance requirements specified in Clause 6.

A PDF/VT-1 conforming reader is a software application or system that is able to read and appropriately process all PDF/VT-1 files in accordance with the conformance requirements specified in Clause 6.

5.3 PDF/VT-2 conformance

A PDF/VT-2 conforming writer is a software application that is able to write PDF/VT-2 files in accordance with the conformance requirements specified in Clause 6.

A PDF/VT-2 conforming reader is a software application or system that is able to read and appropriately process all PDF/VT-2 file sets in accordance with the conformance requirements for PDF/X-4, PDF/X-4p, PDF/X-5g and PDF/X-5pg as specified in Clause 6.

5.4 PDF/VT-2s conformance

A PDF/VT-2s conforming stream is a MIME package of type *application/pdf-vt-stream* containing one or more PDF/VT chunks in which those features necessary for streaming exchange shall adhere to the conformance requirements defined in Annex A.

A PDF/VT-2s conforming writer is a software application that is able to write PDF/VT-2s streams that are in accordance with the conformance requirements defined in A.2.

A PDF/VT-2s conforming reader is a software application or system that is able to read and appropriately process PDF/VT-2s streams in accordance with the conformance requirements defined in A.3.

5.5 All PDF/VT file conformance levels

All conforming interactive readers should present a user interface based on the tree structure of the document part hierarchy defined in 6.5.

All conforming readers shall read and appropriately process the **DPartRoot** dictionary and its **DPart** sub-dictionaries defined in 6.5.

A conforming editor shall be both a conforming reader and a conforming writer.

All conforming PDF/VT editors, when updating a PDF/VT file, shall write a new value for both the **pdfvtid:GTS_PDFVTModDate** and the **xmp:ModifyDate** properties.

To the extent that ISO 15930-7 and ISO 15930-8 and this part of ISO 16612 permit more than one rendering of a PDF/VT file, a conforming reader may use a job ticket or metadata information to further restrict the rendering of the file.

Annex B gives a summary of the permitted PDF/X conformance levels.

5.6 Validation of PDF/VT files

A PDF/VT-1 file can be validated individually against the requirements of this part of ISO 16612.

A PDF/VT-2 file can be validated in isolation against the requirements of this part of ISO 16612. In addition, a PDF/VT-2 file set can be validated. Successful validation of a PDF/VT-2 file in isolation should not be taken to imply that the use of that file within a file set would necessarily lead to a conforming file set.

Only a complete PDF/VT-2 file set can be completely validated against the requirements of this part of ISO 16612. Only a complete PDF/VT-2s stream including all of its referenced files can be completely validated against the requirements of this part of ISO 16612.

Annex B gives a summary of the permitted PDF/X conformance levels.

6 Technical requirements

6.1 General

The PDF features that shall be required, prohibited or restricted are specified in 6.2 to 6.7. These features shall be used as prescribed in the *PDF Reference* and as further restricted by this part of ISO 16612.

6.2 PDF/VT file requirements

6.2.1 PDF/VT-1 file requirements

A PDF/VT-1 file shall conform to PDF/X-4 (but not PDF/X-4p) as defined in ISO 15930-7.

NOTE PDF/X-4p is not included because it permits a reference to an external ICC profile and the PDF/VT-1 conformance level is intended for single-file exchange only.

6.2.2 PDF/VT-2 file requirements

A PDF/VT-2 file shall conform to one of the following:

— PDF/X-5g as defined in ISO 15930-8;

— PDF/X-5pg as defined in ISO 15930-8;

— PDF/X-4p as defined in ISO 15930-7.

NOTE 1 Although PDF/X-5n is a PDF/X-5 conformance level, it is not included in PDF/VT-2 conformance due to the somewhat specialized nature of digital print workflows that make use of n-colorant data for exchange in variable document printing applications.

A PDF/VT-2 file shall not reference pages of a PDF/X-5g, PDF/X-5n or PDF/X-5pg conforming file or another PDF/VT-2 conforming file.

NOTE 2 This requirement means that only pages of PDF/X-1a, PDF/X-3, PDF/X-4, PDF/X-4p, PDF/VT-1 files and conforming ICC profile files can be referenced.

NOTE 3 The implication of only allowing a conforming PDF/VT-2 file to have an external reference to the noted file types is that the only secondary reference allowed from a referenced PDF file is to an ICC profile file.

If a referenced page is in a PDF/VT-1 conforming file, the file shall be interpreted as a PDF/X-4 file and its document part hierarchy and the *GTS_Scope* key within the referenced file shall be ignored by a conforming reader.

A conforming PDF/VT-2 file set shall be a conforming PDF/X-5 file set or PDF/X-4p file set.

6.3 PDF/VT-1 and PDF/VT-2 file identification

A PDF/VT-1 or PDF/VT-2 file shall be identified using the **pdfvtid:GTS_PDFVTVersion** property defined in Table 1. This property shall be present in the metadata stream associated with the **Metadata** key in the document's **Catalog** as required in ISO 15930-7.

The prefix **pdfvtid** with the namespace identifier *http://www.npes.org/pdfvt/ns/id/* shall always be used as shown in Table 1.

Table 1 — PDF/VT identification property

Property	Value type	Category	Description
pdfvtid:GTS_PDFVTVersion	*Text*	Internal	PDF/VT conformance level identifier
pdfvtid:GTS_PDFVTModDate	*Date*	Internal	The date and time at which the PDF/VT file was last written

The values of the **pdfvtid:GTS_PDFVTVersion** property for PDF/VT files prepared in accordance with this part of ISO 16612 shall be as shown in Table 2.

Table 2 — Required pdfvtid:GTS_PDFVTVersion values

Conformance level	pdfvtid:GTS_PDFVTVersion
PDF/VT-1	*PDF/VT-1*
PDF/VT-2	*PDF/VT-2*

NOTE 1 The above conformance levels are for files. The conformance level for a PDF/VT-2s stream is indicated in the MIME header (see Annex A).

All conforming PDF/VT files shall contain both the **pdfvtid:GTS_PDFVTModDate** and the **xmp:ModifyDate** properties in their XMP metadata and they shall both have the same value.

NOTE 2 The above requirements assist a conforming reader in recognizing whether or not a PDF/VT file has been modified by a non-conforming PDF/VT editor, because ISO 15930-7 (PDF/X-4) and ISO 15930-8 (PDF/X-5) require the **xmp:ModifyDate** to match the value of the **ModDate** key in the document information dictionary.

NOTE 3 Provisions for PDF/X-4 and PDF/X-5 with regard to the **pdfxid:GTS_PDFXVersion** entry still apply, independent of, and in addition to, the provisions in this clause with regard to the **GTS_PDFVTVersion** entry.

6.4 Architecture

In variable data and transactional document printing, Page Definition Language (PDL) data used to define the pages of a print job is often an unstructured sequence of page definitions similar to the stream of pages defined in a PostScript file. Such a print stream defines a specific ordering of recipient pages, each containing a sequence of one or more pages where the pages are typically imaged by a digital printer in the exact order they are defined in the PDL file. Such stream-oriented PDL files usually require sequential processing and contain inline device controls used for such things as print substrate selection, single- or double-sided printing, and commands for controlling inline converting and finishing devices. These print data streams are usually device specific and optimized to a particular workflow or to specific capabilities of a target printing device setup and require specialized applications to generate them. Such PDL data streams are typically not portable across different production workflows and digital printing devices once they are created.

As a PDF-based structured page description format, PDF/VT encodes the pages of documents in a manner that allows a conforming reader efficient random access to pages. The random access efficiency of PDF/VT provides an ideal page content resource format for job-ticket-based workflows where the order of page processing by a conforming reader can be different from the order presented in the PDF/VT data.

The use of a separate job ticket file for specifying page ordering and reader processing requirements allows an exchanged PDF/VT instance to be late-stage-targeted or re-targeted to a production workflow, digital printing device or other messaging channel. Re-targeting or reprinting is possible without the need to recreate or modify the PDF/VT instance.

The document part hierarchy is a data structure that specifies the sequence and relationship of documents and/or document parts present in the PDF-VT file. The pages for each recipient are related by one or more **DPart** nodes where each node may specify a page range or contain other **DPart** nodes. The set of pages for

each recipient are defined in a PDF/VT file using a hierarchy of **DPart** dictionaries referred to as document part dictionaries. The root node of this hierarchy of dictionaries is identified by the **DPartRoot** dictionary referenced from the **Catalog** dictionary.

The various **DPart** dictionary nodes of the hierarchy are fully connected by explicit references to the immediate descendant **DPart** nodes and to the parent node (refer to Annex C for an example of a DPart structure).

NOTE 1 Although PDF/VT is based on the *PDF Reference*, which defines structured access to **Page** objects via the pages tree, a conforming reader can also access **Page** object entries indirectly from the **DPart** leaf nodes of the document part hierarchy for a structured presentation of pages.

External job ticket formats such as JDF have specific constructs designed for use with structured PDL formats that contain metadata in their document part structure. In support of that, PDF/VT provides Document Part Metadata (DPM) that can occur as a **DPM** dictionary entry in any **DPart** dictionary of the document part hierarchy. DPM is a way by which conforming writers, such as PDF/VT authoring applications, communicate information about a recipient's documents and document parts to a downstream production workflow or fulfilment service, or both.

There are many potential uses of DPM. For example, it can be used in a workflow to apply production rules, to conserve information about intended recipients, or for archival purposes, dynamic bar code printing or slug line generation.

Conceptually, a PDF/VT file with DPM in combination with the document part hierarchy is analogous to a structured database of final form variable content pages. This structuring and use of DPM allows a job ticket to refer to the PDF/VT pages in a way that is conceptually similar to a structured database select or query.

NOTE 2 The JDF 1.4 specification defines constructs that permit a JDF job ticket to specify dynamic production rules based on the PDF/VT document part hierarchy structure and DPM present within it.

6.5 Document part hierarchy

The **Catalog** dictionary of a conforming PDF/VT file shall have a **DPartRoot** key whose value is an indirect object reference to a **DPartRoot** dictionary.

In Table 3 and Table 4, the identifiers *Required*, *Optional* and *Conditional* are used to indicate the usage of the key.

At least one **DPart** dictionary shall be present in the document part hierarchy.

A **Page** object shall only be referenced in the range of pages specified by the **Start** and **End** keys of a single **DPart** dictionary entry.

Each **Page** object defined in the PDF/VT file shall be included in the page range defined by one and only one **DPart** dictionary.

Each **Page** object shall have a **DPart** key that has a value that is an indirect reference to the leaf node **DPart** dictionary whose range of pages includes this **Page** object.

NOTE 1 The **DPart** key in a **Page** object allows a conforming reader to directly retrieve the section of the document part hierarchy that applies to this **Page** object. For example, this allows for efficient retrieval of DPM based on page indices for certain implementation approaches for cut-and-stack imposition. It also enables ready access of DPM data in conforming interactive reader applications.

Table 3 — DPartRoot dictionary

Key	Type	Usage of key[a]	Value
Type	name	*Optional*	If present, it shall have the value *DPartRoot* to identify the dictionary as a **DPartRoot** dictionary.
DPartRootNode	dictionary	*Required*	It shall be present and shall be an indirect reference to the **DPart** dictionary that is the root node of the document part hierarchy.
RecordLevel	integer	*Optional*	If present, this attribute shall identify the zero-based level of the document part hierarchy where each **DPart** node of that level corresponds to a recipient record. The value 0 corresponds to the **DPart** node identified by the value of the **DPartRoot** dictionary's **DPartRootNode** key. NOTE 1 If this key is not present, then recipient records are not explicitly identified in this file. NOTE 2 The example in Annex C shows **DPart** nodes identified by **RecordLevel** with child **DPart** subnodes.
NodeNameList	array of names	*Required*	It shall be present. Each name entry in this array shall correspond to a **DPart** node level of the document part hierarchy beginning with the **DPart** dictionary identified by the value of the **DPartRoot** dictionary's **DPartRootNode** key. The number of entries present in this array shall be equal to the number of **DPart** node levels in the document part hierarchy. Each entry in the **NodeNameList** array shall conform to the rules for an XML NMTOKEN after expansion of # escapes. NOTE The **NodeNameList** array is used in the interpretation of the document part hierarchy as XML, as specified in Annex D.

[a] The usage of the key is defined as follows:

Required indicates that the key shall be present at all times.

Optional indicates that the key may be present at the discretion of the conforming writer.

The order of **Page** objects as defined by the page tree shall be the same order in which **Page** objects are referenced from leaf node **DPart** dictionaries in a depth-first traversal of the document part hierarchy.

NOTE 2 By ordering the PDF pages as referenced from the page tree to be in the same order as they are referenced from **DPart** leaf nodes of the document part hierarchy, it is possible for existing PDF viewers that do not conform to PDF/VT and ignore the document part hierarchy to present the pages in the same order as those that follow the document part hierarchy.

A child **DPart** dictionary shall not be referenced by more than one parent **DPart** dictionary.

NOTE 3 The above requirement makes the **DPart** dictionaries part of a tree structure.

All **DPart** dictionaries should be written into a conforming PDF/VT file in a single compressed object stream as defined in *PDF Reference*, 3.4.6.

NOTE 4 This enables improvement in conforming reader performance (e.g. single read operation) and file size optimization through the use of compression of similar items. However, performance of random access to the **DPart** nodes can be impacted in cases where PDF/VT files have a large number of records. In such a case, multiple compressed object streams might be preferred.

Table 4 — DPart dictionary

Key	Type	Usage of key[a]	Value
Type	name	*Optional*	If present, it shall have the value *DPart* to identify the dictionary as a **DPart** dictionary.
Parent	dictionary	*Required*	It shall be present. If this **DPart** dictionary is referenced from the **DPartRootNode** key of the **DPartRoot** dictionary, then the value of the **Parent** key shall be an indirect reference to the **DPartRoot** dictionary. In all other cases, the value of the **Parent** key shall be an indirect reference to the **DPart** dictionary that is its immediate ancestor.
DParts[b]	array of arrays	*Conditional*	It specifies immediate descendent **DPart** dictionaries and shall be present if this **DPart** dictionary has no **Start** key and shall not be present otherwise. If present, the value of this key shall be an array. Each element in the array shall be an array of indirect references to immediate descendant **DPart** dictionaries. If present, at least one immediate descendent **DPart** dictionary shall be specified. All but the last array entry shall have exactly 8192 elements and the last array entry shall have at least 1 and at most 8192 entries. NOTE This definition is intended to allow easy access of many elements in an efficient way while not limiting the total number of immediate **DPart** descendants that can be specified by a **DPart** dictionary.
Start[b]	dictionary	*Conditional*	It shall be present if this **DPart** dictionary has no **DParts** key and shall not be present otherwise. If present, the **Start** key shall be an indirect reference to the **Page** object that defines the first PDF page of the range of pages belonging to this **DPart** dictionary.
End[b]	dictionary	*Conditional*	It shall be present if this **DPart** dictionary has a **Start** key and the page range has more than one page. Otherwise it shall not be present. If present, the **End** key shall be an indirect reference to the Page object that defines the last PDF page of the range of pages belonging to this **DPart** dictionary.
DPM	dictionary	*Optional*	If present, it shall specify a dictionary of metadata represented as key/value pairs. NOTE See 6.6 for a description of the use of **DPM** dictionaries for specifying DPM.

[a] The usage of the key is defined as follows:

Required indicates that the key shall be present at all times.

Optional indicates that the key may be present at the discretion of the conforming writer.

Conditional indicates that the presence of the key is dependent on a condition such as the presence or absence of other keys.

[b] Only leaf DPart nodes use the Start and End keys and intermediate DPart nodes use the DParts key and no Start and End keys.

Non-leaf DPart dictionary nodes of the hierarchy cannot refer to a range of PDF pages, only to descendent DPart dictionaries.

6.6 Document part metadata (DPM)

DPM is application-specific information communicated between the creator of PDF/VT data and a receiving system, and can be used to classify the PDF/VT file and its document parts. The receiving system could, for example, use this information in the purposing of page content to print or to some other messaging channel for presentation to a recipient. The DPM can also be referenced in a job definition, such as a JDF job ticket, to vary processing control as necessary during print production.

Annex D defines a mapping between the document part structure and the encoding of DPM that provides a common language for external processes to query for the DPM present in a PDF/VT file.

DPM, if present, shall be specified using the **DPM** key in a **DPart** dictionary of the document part hierarchy.

Although **DPart** nodes are related in the structure of the document part hierarchy, the DPM specified by a **DPM** dictionary entry shall only apply to the **DPart** node in which it is defined and is not inherited by child **DPart** nodes.

The name of all keys present in a **DPM** dictionary or any other dictionaries contained within, or referenced from, a **DPM** dictionary shall conform to the rules of an XML NMTOKEN after expansion of # escapes.

NOTE 1 This ensures that the names of all those keys can be used directly in the XML representation of a DPM dictionary.

No two keys present in the DPM dictionary shall have the same name after expanding hash escape sequences.

DPM key names shall adhere to the requirements specified in Annex E.

If a **GTS_Managed** key is present in a **DPM** dictionary, it shall have a value that is a dictionary that contains the DPM properties that might need to be changed when the PDF/VT file is modified by a conforming PDF/VT editor. A PDF/VT editor shall ensure that the values of the keys in the **GTS_Managed** dictionary are consistent with the modifications made to the PDF/VT file or shall move them from the **GTS_Managed** dictionary to the **GTS_Suspect** dictionary.

NOTE 2 Keys present in the **GTS_Managed** dictionary have values that are dependent on other parts of the PDF/VT file; therefore, modifications to the PDF/VT file might impact these values.

NOTE 3 **GTS_Managed** and **GTS_Suspect** assist a conforming editor to preserve consistency of DPM which it does not recognize.

If a **GTS_Suspect** key is present in a **DPM** dictionary, it shall have a value that is a dictionary.

NOTE 4 Keys in the **GTS_Suspect** dictionary are associated with DPM values that might no longer be consistent.

6.7 Recurring graphical object definitions

6.7.1 General

XObjects should be used for encoding graphical objects referenced more than once from page content streams present in a PDF/VT file or commonly used in page content streams present in multiple PDF/VT files.

NOTE 1 An optimized conforming writer typically factors out the definitions of all multiple-referenced graphical objects and writes them only once in the PDF/VT instance as an appropriate type of **XObject**. This method of factoring out redundant graphical content definitions within a PDF/VT file is a means by which a conforming writer can reduce the size of a conforming file and concisely convey graphical object semantics to an optimized conforming reader.

NOTE 2 A common performance optimization strategy for a conforming reader implementation, such as a PDF/VT conforming RIP, is to minimize the need to re-process an **XObject** known to be referenced multiple times with the same intended appearance.

A graphical object may be encoded as an **XObject** regardless of whether it is referenced a single time or multiple times from one or more **Page** object content streams in the PDF/VT instance.

Transparency should not be used in annotation appearance streams.

6.7.2 Unique identification of an XObject

An **XObject** can occur in more than one related PDF/VT file such as those that belong to a PDF/VT instance or file collection. In the case where the conforming writer is aware that equivalent **XObjects** (as described in the **XObject** comparison algorithm in Annex F) occur in multiple related PDF/VT files, a **GTS_XID** key should be present in each of those equivalent **XObjects** and if present shall have the same unique value.

If present, the **GTS_XID** key of an **XObject** shall have a value of type "string" and should have a value that is a 128-bit number in the form of a uuid-schemed URI as defined in RFC 4122. This value should be generated in such a way that there is a high probability that it is unique. There are various common schemes for generating a unique identifier as defined in Annex F.

NOTE 1 A number in the form of a uuid-schemed URI starts with the prefix "uuid:".

Two **XObject**s with the same value for their **GTS_XID** keys specified within a PDF/VT file collection shall be considered equivalent **XObject**s.

NOTE 2 This provision allows for optimized processing of recurring content across multiple PDF/VT files.

6.7.3 XObject usage scope hints

The number of references to an **XObject** within a PDF/VT file is defined as the number of occurrences of the **Do** operator that refer to that **XObject** present in all of the content streams in the PDF/VT file.

NOTE 1 An **XObject** can be referenced multiple times with different active graphics states. This can influence a conforming reader's performance optimization strategy.

NOTE 2 It is possible to have an **XObject** referenced from another **XObject** that itself is referenced multiple times. Therefore, the number of occurrences of the **Do** Operator is not a measure of the times an **XObject** will be rendered.

Any equivalent **XObject**s that occur within or across a PDF/VT file collection should have a **GTS_XID** and a **GTS_Scope** key.

NOTE 3 This requirement provides a hint to a conforming reader that an equivalent **XObject** might be present in other PDF/VT chunks.

The **GTS_Scope** key may be present in an **XObject**.

The **GTS_Scope** key identifies the scope within which the graphical content defined by an **XObject** may be referenced from within a PDF/VT file, file collection or instance. A conforming writer should specify a **GTS_Scope** key in an **XObject** if the lifetime scope of the **XObject** is known.

NOTE 4 Knowledge of the lifetime scope of references to **XObject**s provides hint information useful for optimized processing by a conforming reader.

If present, the **GTS_Scope** key shall have a value of type "name" that is one of *SingleUse*, *Record*, *File*, *Stream, Global* or *Unknown*. If the **DPartRoot** dictionary does not contain the **RecordLevel** key, the **GTS_Scope** key shall not have the value of **Record.** A **GTS_Scope** key whose value is **Stream** shall not be used in a PDF/VT file not contained in a PDF/VT-2s stream.

The **GTS_Scope** key may be present in an **XObject** and have a value of *Unknown* if the scope of the **XObject** is not known to the conforming writer.

If **GTS_Scope** is not present in an **XObject**, its default value shall be *Unknown.*

If the value for the **GTS_Scope** key is *SingleUse*, then the **XObject** shall be referenced no more than once from a **Do** operator present in the PDF/VT file.

NOTE 5 Even though an **XObject** could be referenced only once, this single reference can occur in a content stream that is referenced multiple times, causing that **XObject** to be rendered multiple times.

If the value for the **GTS_Scope** key is *Record*, the **XObject** should be referenced more than once from **Do** operators present in the PDF/VT file from the pages belonging to a single recipient record, and shall not be referenced from pages outside the pages belonging to this record.

If the value of the **GTS_Scope** key is *File*, the **XObject** should be referenced more than once from **Do** operators present in the PDF/VT file.

If the value for the **GTS_Scope** key is *Stream*, this **XObject**, or an **XObject** with an identical **GTS_XID**, should be referenced from one or more **Do** operators present in the PDF/VT-2s stream containing this PDF/VT file.

If the value for the **GTS_Scope** key is *Global*, this **XObject**, or an **XObject** with an identical **GTS_XID**, should be referenced from one or more **Do** operators present in more than one PDF/VT instance.

NOTE 6 The performance of a conforming reader can be severely reduced if the **GTS_Scope** values are not precise or are incorrect.

6.7.4 Identification of XObject environment context

XObjects specified in a conforming PDF/VT file may be identified as belonging to an environment context that may be managed by a conforming reader.

The value of the **GTS_Env** key shall be of type text string and shall specify the name of the environment context to which the **XObject** belongs.

A **GTS_Env** key shall be present in an **XObject** if the **GTS_Scope** key is present and has a value of *Stream* or *Global*, and may be present for all other values of the **GTS_Scope** key.

NOTE The presence of a **GTS_Env** key in an **XObject** allows a conforming reader, such as a RIP, to provide a user interface for managing groups of related cached entries. Such a management interface can allow for loading and unloading of processed and reused **XObject**s cached by a conforming reader. Named environments of related **XObject**s can also enable selective archiving and previewing.

6.7.5 Encapsulated XObject**s**

An encapsulated **XObject** is an **XObject** having a well-defined limited interaction with the current *graphics state* (see *PDF Reference*, 4.3) at the point of invocation, as specified in this subclause.

All encapsulated **XObjects** shall have a **Group** key present if:

a) the graphic state for any element in any content stream on any page of the conforming PDF/VT file contains:

   — an **SMask** key with a dictionary value; or

   — a **ca** or **CA** key with a real number value less than 1; or

   — a **BM** key with a value other than *Normal* or *Compatible*; or

b) any Image **XObject** that is a print element in the conforming PDF/VT file contains:

   — an **SMask** key with a dictionary value; or

   — an **SMaskInData** key with an integer value greater than zero.

Each of these **Group** keys shall have a value (of type “dictionary”) that shall contain an **I** key and a **CS** key. The **I** key shall have a value of *true*.

NOTE 1 The above requirement ensures that the **XObject** is defined as an isolated transparency group in a well-defined blending colour space, thereby ensuring a consistent rendering, independent of the calling context.

NOTE 2 Marking encapsulated XObjects as transparency groups when a conforming PDF/VT file is otherwise fully opaque might cause some PDF/VT readers to perform less optimally.

An encapsulated **XObject**, including any content streams referenced from its definition, shall explicitly set all graphic state parameters that influence the appearance of path painting, text showing, **XObject** and inline image operators (see *PDF Reference*, Table 4.1) used in that encapsulated **XObject**, except the current transformation matrix (**CTM**), clipping path, soft mask (**SMask**), current fill opacity (**ca**), current stroke opacity (**CA**) and transparency blending mode (**BM**) graphic state parameters.

NOTE 3 The excluded transparency-related graphics state parameters are set explicitly to default values when the **XObject** is invoked if it is marked as an isolated transparency group. If it is not a transparency group, then these parameters are already defined in their default state.

If the **Subtype** key of an encapsulated **XObject** has the value of *Image*, and the **ImageMask** key is not present or has the value *false*, then the **Intent** key shall be present.

NOTE 4 Use of the **Intent** key of an **Image XObject** will affect the appearance of the **XObject** rendition and therefore needs to be explicitly defined.

The content stream of an encapsulated **XObject** shall not contain marked content operators used for identifying optional content nor shall it directly or indirectly refer to other **XObject**s whose content streams contain marked content operators that identify optional content.

NOTE 5 This does not allow an encapsulated **XObject** to vary its appearance based on optional content; however, the **XObject** itself can be part of the optional content by specifying an **OC** key.

An **XObject** may have a **GTS_Encapsulated** key whose value is a boolean, which indicates whether the **XObject** is an encapsulated **XObject**. If the **GTS_Encapsulated** key is present with a value of *true*, then the **XObject** shall conform to all requirements of 6.7.5 and should also have a **GTS_Scope** key as specified in 6.7.3.

Annex A
(normative)

Use of multipart MIME for streamed generation of PDF/VT data

A.1 General considerations

Streamed generation of PDF/VT data allows a PDF/VT-2s conforming reader to start processing a PDF/VT instance even when the conforming writer has not yet finished producing it.

A PDF/VT instance can be sent as a stream of one or more PDF/VT chunks encoded as a MIME package.

NOTE MIME packaging guarantees the order of processing of the sequence of chunks.

In a streaming workflow, the PDF/VT-2s conforming reader can start processing the PDF/VT instance after receiving the first PDF/VT chunk, which can be much smaller than the whole PDF/VT instance.

The MIME packaging mechanism is intended for the transmission of a PDF/VT instance as a stream via a communication channel. It is not intended for storing a PDF/VT instance as a file on a file system.

A.2 PDF/VT-2s conformance requirements

A PDF/VT-2s stream is a sequence of PDF/VT chunks packed together using MIME packaging, as defined in RFC 2045, RFC 2046, RFC 2047, and RFC 2183.

A PDF/VT-2s stream (the complete stream) shall have the Content-Type *multipart/mixed.*

A PDF/VT-2s stream shall contain a header field named *X-PDFVT-Stream-version* with the value *1.* This header field shall appear in the stream ahead of the content type field.

The MIME parts of a PDF/VT-2s stream that are PDF files shall have the content type *application/pdf*.

The MIME parts of a PDF/VT-2s stream that are ICC profile files shall have the content type *application/vnd.iccprofile* and shall have content disposition *attachment*. Additionally, the content disposition parameter *filename* shall be present.

The MIME parts of a PDF/VT-2s stream that encode PDF/VT files containing content to be directly interpreted shall have content disposition *inline*.

The MIME parts of a PDF/VT-2s stream that encode PDF files containing re-usable content only (files referred to by PDF/VT-2 conforming files) shall have content disposition *attachment*. Additionally, the content disposition parameter *filename* shall be present.

When referring to PDF files or ICC profiles in the PDF/VT-2s stream, the PDF files shall only reference those MIME parts that are encoded earlier in the stream and shall refer to them by the values of their *filename* parameters defined in the MIME stream.

MIME parts of a PDF/VT-2s stream with the content disposition of *inline* shall not be referenced by PDF/VT-2 files present in the stream.

All PDF/VT files in a PDF/VT-2s stream shall be prepared for the same characterized printing condition.

NOTE 1 This requirement makes the set of files in a PDF/VT-2s stream consistent with the requirements of a PDF/VT-2 file set.

All PDF/VT files in a PDF/VT-2s stream shall have a **DPartRoot** dictionary whose **NodeNameList** keys have the same value. Additionally, all **RecordLevel** keys in each of the **DPartRoot** dictionaries shall have the same value.

NOTE 2 Job ticketing requirements might be more complex if the set of pages intended for a single recipient span more than one PDF/VT chunk.

For more efficient transmission of the PDF/VT-2s stream data, the MIME parts should have a content transfer encoding field set to binary.

A.3 PDF/VT-2s reader conformance requirements

A PDF/VT-2s conforming reader shall interpret a PDF/VT-2s stream as having a virtual document part hierarchy that references the pages of all the PDF files present in the stream marked with a content disposition of *inline* in the order in which they appear in the PDF/VT-2s stream. The virtual document part hierarchy is the aggregation of the document part hierarchy of each of those PDF/VT files in the order in which they occur.

NOTE 1 Conceptually, each PDF/VT file's document part hierarchy has a super root node that is the parent of the top-level **DPart** node. In a virtual document part hierarchy, the super root node has multiple child nodes, one for each of the aggregated PDF/VT files. This difference needs to be taken into consideration by a PDF/VT-2s conforming reader when navigating the virtual document part hierarchy of a PDF/VT-2s stream.

The default page processing order of a PDF/VT-2s conforming stream by a PDF/VT-2s conforming reader shall be the order defined by the virtual document part hierarchy.

NOTE 2 A PDF/VT-2s conforming reader is expected to start processing the PDF/VT chunks of the MIME stream as soon as each PDF/VT chunk is completely received.

A.4 XML representation

Should the virtual document part hierarchy, as defined in A.3, need to be represented as an XML hierarchy, a conforming reader shall do so as specified in Annex D and modified by the present Clause.

The **PDFVT** root XML element shall have as its children the XML representation of the root **DPart** node of each of the PDF/VT files in the PDF/VT-2s stream that has a content disposition inline and shall appear in the order in which they appear in the stream.

Annex B
(informative)

PDF/X conformance summary

Table B.1 summarizes the permitted PDF/X conformance levels to which a PDF/VT-1 or PDF/VT-2 file also conforms.

Table B.1 — Cross-reference of conformance levels of PDF/X and PDF/VT files

PDF/X	PDF/VT-1	PDF/VT-2
PDF/X-4	√	
PDF/X-4p		√
PDF/X-5g		√
PDF/X-5n		
PDF/X-5pg		√
PDF/X-1a		
PDF/X-3		

Table B.2 summarizes the conformance of files permitted to be referenced from a PDF/VT-2 file or PDF/VT-2s stream.

Table B.2 — Cross-reference of PDF/VT and PDF/X files

PDF/X	Referenced by a PDF/VT-2 file	Contained in a PDF/VT-2s stream
PDF/VT-1	√	√
PDF/VT-2		√
PDF/X-4	√	√
PDF/X-4p	√	√
PDF/X-5g		
PDF/X-5n		
PDF/X-5pg		
PDF/X-1a	√	√
PDF/X-3	√	√
ICC profile	√	√

Annex C
(informative)

Example DPart structure

The following PDF objects represent that portion of a conforming PDF/VT file that defines its document part hierarchy and includes document part metadata (DPM). This hierarchy includes the definition of three recipient records, each of which is an immediate child of the root document part (**DPart)** dictionary. For each recipient record, three **DPart** nodes are included (record level, cover and body). The record level **DPart** node refers to two other **DPart** nodes which refer to the pages for the cover and body pages. The **DPart** node for the cover refers to two pages and the **DPart** node for the body refers to four pages. All pages have the same size and orientation, and are intended for full colour printing.

The **DPM** dictionary present in the root **DPart** node of the DPart hierarchy contains metadata regarding the PDF/VT instance as a whole (**CIP4_Metadata** and **CIP4_Summary**).

Each record level **DPart** node includes metadata defining a recipient record identifier (**CIP4_UniqueID**), recipient's name (**CIP4_Person)**, recipient's address (**CIP4_Address**), and customer status information (**ACME_CustStatus**).

The **DPart** nodes for the cover and the body are identified using the **CIP4_ProductType** keys.

Production information indicating that the finished document for each recipient record is a saddle-stitched booklet is provided externally to the PDF/VT file (e.g. in a JDF job ticket).

In the example below, the **CIP4** prefix is used for DPM entries defined by CIP4. The fictitious **ACME** prefix is used for DPM entries not defined in this part of ISO 16612 or in CIP4 specifications. See Annex E for more information.

```
% The file's Catalog dictionary:
2 0 obj
<<
    /DPartRoot 41 0 R
    /Pages 4 0 R
    /Type /Catalog
>>
endobj

% The DPartRoot node is the root of the hierarchy referenced
% from the file's Catalog dictionary:
41 0 obj
<<
    /DPartRootNode 42 0 R
    /NodeNameList [/Root /Record /DocPart ]
    /RecordLevel 1
    /Type /DPartRoot
>>
endobj

% The first DPart node that is the root of the hierarchy referenced
% from the DPartRoot dictionary:
42 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Metadata <<
```

```
                /CIP4_Conformance (Meta-L1)
                /CIP4_Creator (WG2TF3 scripting prototype - v0.2)
                /CIP4_JobID (JobIdentifier0)
                /CIP4_ModificationDate (2010-02-10T19:34:00+01:00)
                /CIP4_Sender <<
                    /CIP4_Address <<
                        /CIP4_City (Red Hook)
                        /CIP4_CivicNumber (2400)
                        /CIP4_Country (United States)
                        /CIP4_PostalCode (16612-0002)
                        /CIP4_Region (NY)
                        /CIP4_StreetName (Easy Street)
                    >>
                    /CIP4_Person <<
                        /CIP4_Department (Procurement)
                        /CIP4_FirstName (Baldrick)
                        /CIP4_LastName (Turnip)
                        /CIP4_Organization (Acme Communications)
                    >>
                >>
            >>
            /CIP4_Summary <<
                /CIP4_PageCount 18
                /CIP4_RecipientCount 3
                /CIP4_Uniform <<
                    /CIP4_Color true
                    /CIP4_Orientation true
                    /CIP4_Size true
                >>
                /CIP4_UniformRecipientStructure true
            >>
        >>
    >>
    /DParts [43 0 R ]
    /Parent 41 0 R
    /Type /DPart
>>
endobj
43 0 obj
[44 0 R 48 0 R 51 0 R ]
endobj

% Intermediate DPart node representing the first recipient record:
44 0 obj
<<
    /DPM <<
        /ACME_CustStatus (prospective)
        /CIP4_Root <<
            /CIP4_Summary <<
                /CIP4_PageCount 6
            >>
            /CIP4_Production <<
                /CIP4_CopyCount 1
                /CIP4_Part <<
                    /CIP4_ProductType /Brochure
                >>
            >>
            /CIP4_Recipient <<
                /CIP4_Contact <<
                    /CIP4_Address <<
```

```
                        /CIP4_City (Phoenix)
                        /CIP4_CivicNumber (10)
                        /CIP4_Country (USA)
                        /CIP4_PostalCode (81203)
                        /CIP4_Region (AZ)
                        /CIP4_StreetName (Ocean Drive)
                    >>
                    /CIP4_Person <<
                        /CIP4_FirstName (Jane)
                        /CIP4_LastName (Smith)
                    >>
                >>
                /CIP4_UniqueID (ID_0)
            >>
        >>
    >>
    /DParts [[46 0 R 47 0 R ] ]
    /Parent 42 0 R
    /Type /DPart
>>
endobj

% Leaf DPart node that is a sub-node of the first record representing
% the two pages of the booklet cover:
46 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Production <<
                /CIP4_Part <<
                    /CIP4_ProductType /Cover
                >>
            >>
        >>
    >>
    /End 6 0 R
    /Parent 44 0 R
    /Start 5 0 R
    /Type /DPart
>>
endobj

% Leaf DPart node that is a sub-node of the first record representing
% the four pages of the booklet body
47 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Production <<
                /CIP4_Part <<
                    /CIP4_ProductType /Body
                >>
            >>
        >>
    >>
    /End 10 0 R
    /Parent 44 0 R
    /Start 7 0 R
    /Type /DPart
>>
```

```
endobj

% Intermediate DPart node representing the second recipient record:
48 0 obj
<<
    /DPM <<
        /ACME_CustStatus (Prospective)
        /CIP4_Root <<
            /CIP4_Summary <<
                /CIP4_PageCount 6
            >>
            /CIP4_Production <<
                /CIP4_CopyCount 1
                /CIP4_Part <<
                    /CIP4_ProductType /Brochure
                >>
            >>
            /CIP4_Recipient <<
                /CIP4_Contact <<
                    /CIP4_Address <<
                        /CIP4_City (Phoenix)
                        /CIP4_CivicNumber (96)
                        /CIP4_Country (USA)
                        /CIP4_PostalCode (81215)
                        /CIP4_Region (AZ)
                        /CIP4_StreetName (South Ave)
                    >>
                    /CIP4_Person <<
                        /CIP4_FirstName (Mary)
                        /CIP4_LastName (Smith)
                    >>
                >>
                /CIP4_UniqueID (ID_1)
            >>
        >>
    >>
    /DParts [[49 0 R 50 0 R ] ]
    /Parent 42 0 R
    /Type /DPart
>>
endobj

% Leaf DPart node that is a sub-node of the second record representing
% the two pages of the booklet cover:
49 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Production <<
                /CIP4_Part <<
                    /CIP4_ProductType /Cover
                >>
            >>
        >>
    >>
    /End 12 0 R
    /Parent 48 0 R
    /Start 11 0 R
    /Type /DPart
>>
```

```
endobj

% Leaf DPart node that is a sub-node of the second record representing
% the four pages of the booklet body:
50 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Production <<
                /CIP4_Part <<
                    /CIP4_ProductType /Body
                >>
            >>
        >>
    >>
    /End 16 0 R
    /Parent 48 0 R
    /Start 13 0 R
    /Type /DPart
>>
endobj

% Intermediate DPart node representing the third recipient record:
51 0 obj
<<
    /DPM <<
        /ACME_CustStatus (Prospective)
        /CIP4_Root <<
            /CIP4_Summary <<
                /CIP4_PageCount 6
            >>
            /CIP4_Production <<
                /CIP4_CopyCount 1
                /CIP4_Part <<
                    /CIP4_ProductType /Brochure
                >>
            >>
            /CIP4_Recipient <<
                /CIP4_Contact <<
                    /CIP4_Address <<
                        /CIP4_City (Phoenix)
                        /CIP4_CivicNumber (54)
                        /CIP4_Country (USA)
                        /CIP4_PostalCode (81218)
                        /CIP4_Region (AZ)
                        /CIP4_StreetName (North Street)
                    >>
                    /CIP4_Person <<
                        /CIP4_FirstName (Frederick)
                        /CIP4_LastName (Adams)
                    >>
                >>
                /CIP4_UniqueID (ID_2)
            >>
        >>
    >>
    /DParts [[52 0 R 53 0 R ] ]
    /Parent 42 0 R
    /Type /DPart
>>
```

```
endobj

% Leaf DPart node that is a sub-node of the third record representing
% the two pages of the booklet cover:
52 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Production <<
                /CIP4_Part <<
                    /CIP4_ProductType /Cover
                >>
            >>
        >>
    >>
    /End 18 0 R
    /Parent 51 0 R
    /Start 17 0 R
    /Type /DPart
>>
endobj

% Leaf DPart node that is a sub-node of the third record representing
% the four pages of the booklet body:
53 0 obj
<<
    /DPM <<
        /CIP4_Root <<
            /CIP4_Production <<
                /CIP4_Part <<
                    /CIP4_ProductType /Body
                >>
            >>
        >>
    >>
    /End 22 0 R
    /Parent 51 0 R
    /Start 19 0 R
    /Type /DPart
>>
Endobj

% Page objects:
5 0 obj
<<
    /Contents [23 0 R ]
    /DPart 46 0 R
    /Parent 4 0 R
    /Resources <<
        /Font <<
            /Fm1 3 0 R
            /FmS1 45 0 R
        >>
    >>
    /Type /Page
>>
endobj
%
% Additional PDF objects occur here  ...
%
```

```
%
22 0 obj
<<
    /Contents [40 0 R ]
    /DPart 53 0 R
    /Parent 4 0 R
    /Resources <<
        /Font <<
            /Fm1 3 0 R
            /FmS1 45 0 R
        >>
    >>
    /Type /Page
>>
endobj
%
% Additional PDF objects occur here  ...
%
%
```

Annex D
(normative)

XML representation of the document part hierarchy

D.1 General considerations

A PDF/VT file's structure can be represented as an XML node hierarchy to facilitate processing by applications (e.g. JDF-based) that use XML XPath queries [e.g. use of **count**(**PDFPage**) and other XPath operations] to refer to a structured PDL's metadata definitions. Further guidance is given in the XML Path Language (XPath) 1.0 specification.

Representation of conforming PDF/VT files as an XML structure includes the mapping of **DPart**, **DPM** dictionaries, and **Page** objects to their representation as XML nodes. This mapping is defined in D.2.

D.2 Technical requirements

D.2.1 DPart encoding

The root element of the XML representation of the document part hierarchy shall be **PDFVT**.

Each **DPart** dictionary at the same level in the document part hierarchy shall have the XML element name as defined by that level's entry in the **NodeNameList** key of the **DPartRoot** dictionary.

The XML representation of a **DPart** node includes the XML representations of each of its immediate descendent **DPart** nodes in the order in which they are defined.

The XML representation of the **DPart** node at level 0 shall be a child of the **PDFVT** XML root element.

The XML representation of a leaf **DPart** node shall include an XML element named **PDFPage** for each **Page** object contained in the page range of that **DPart** node.

The XML representation of a **DPart** node shall include as its first child an XML representation of the **DPM** dictionary of that **DPart** node, if present.

D.2.2 DPM encoding

Each **DPM** dictionary of a **DPart** dictionary shall be represented by the XML element named **DPM**.

PDF name hash escape sequences shall be expanded before XML representation rules are applied.

Each key present in a **DPM** dictionary, except those whose value is *null*, shall be represented as an XML element that is a sub-element of the XML **DPM** element and the name of that element shall be the name of the key where all COLON (3Ah) characters shall be substituted with LOW LINE (5Fh) characters. The value of that key shall be represented as the content of that XML sub-element in accordance with the following rules.

— A string value shall be represented as the sequence of valid XML characters.

— A name value shall be represented as the sequence of valid XML characters that form the string representation of the name value (i.e. without the leading slash).

— An array value shall be represented as a sequence of XML child elements, and each of the XML child elements shall have the XML element name **Item**. The content of each such XML child element shall be derived from the corresponding PDF array element entry according to these rules for values of keys in a **DPM** dictionary.

— A dictionary value shall be represented as a sequence of XML child elements. Each XML child element shall have an element name that is the name of the key associated with the corresponding dictionary entry where all COLON (3Ah) characters shall be substituted with LOW LINE (5Fh) characters. The content of each XML child element shall be derived from the key value according to these rules for values of keys in a **DPM** dictionary. A dictionary key whose value is *null* shall not be represented in XML.

NOTE 1 The reason for excluding references to *null* is for consistency with *PDF Reference,* 3.2.6.

— Boolean values shall be represented as *true* or *false*.

— Numeric values shall be represented as a sequence of valid XML characters representing the value using PDF syntax (see *PDF Reference*, 3.2.2).

— A stream value shall be represented by the value of its associated stream dictionary and not by the actual data in the stream.

NOTE 2 The associated stream dictionary provides information about the stream while the stream itself is typically binary data.

If the **DPM** dictionary of a **DPart** node is queried for a particular key, the presence of that same key in the **DPM** dictionary of an ancestor node shall not be considered in the scope of that query.

D.3 Example XML representation of a document part hierarchy

The following is an example of the XML representation of the PDF encoded document part hierarchy example of Annex C in accordance with the technical requirements specified in D.2.

```
<PDFVT>
    <Root>
        <DPM>
            <CIP4_Root>
                <CIP4_Metadata>
                    <CIP4_Conformance>Meta-L1</CIP4_Conformance>
                    <CIP4_Creator>WG2TF3 scripting prototype - v0.2</CIP4_Creator>
                    <CIP4_JobID>JobIdentifier0</CIP4_JobID>
                    <CIP4_ModificationDate>2010-02-10T19:34:00+01:00\
                    </CIP4_ModificationDate>
                    <CIP4_Sender>
                        <CIP4_Address>
                            <CIP4_City>Red Hook</CIP4_City>
                            <CIP4_CivicNumber>2400</CIP4_CivicNumber>
                            <CIP4_Country>United States</CIP4_Country>
                            <CIP4_PostalCode>16612-0002</CIP4_PostalCode>
                            <CIP4_Region>NY</CIP4_Region>
                            <CIP4_Street>Easy Street</CIP4_Street>
                        </CIP4_Address>
                        <CIP4_Person>
                            <CIP4_Department>Procurement</CIP4_Department>
                            <CIP4_FirstName>Baldrick</CIP4_FirstName>
                            <CIP4_LastName>Turnip</CIP4_LastName>
                            <CIP4_Organization>Acme Communications</CIP4_Organization>
                        </CIP4_Person>
                    </CIP4_Sender>
                </CIP4_Metadata>
                <CIP4_Summary>
                    <CIP4_PageCount>18</CIP4_PageCount>
                    <CIP4_RecipientCount>3</CIP4_RecipientCount>
```

```
                <CIP4_Uniform>
                    <CIP4_Color>true</CIP4_Color>
                    <CIP4_Orientation>true</CIP4_Orientation>
                    <CIP4_Size>true</CIP4_Size>
                </CIP4_Uniform>
                <CIP4_UniformRecipientStructure>true\
                </CIP4_UniformRecipientStructure>
            </CIP4_Summary>
        </CIP4_Root>
    </DPM>
    <Record>
        <DPM>
            <ACME_CustStatus>Prospective</ACME_CustStatus>
            <CIP4_Root>
                <CIP4_Summary>
                    <CIP4_PageCount>6</CIP4_PageCount>
                </CIP4_Summary>
                <CIP4_Production>
                    <CIP4_CopyCount>1</CIP4_CopyCount>
                    <CIP4_Part>
                        <CIP4_ProductType>Brochure</CIP4_ProductType>
                    </CIP4_Part>
                </CIP4_Production>
                <CIP4_Recipient>
                    <CIP4_Contact>
                        <CIP4_Address>
                          <CIP4_City>Phoenix</CIP4_City>
                          <CIP4_CivicNumber>10</CIP4_CivicNumber>
                          <CIP4_Country>USA</CIP4_Country>
                          <CIP4_PostalCode>81203</CIP4_PostalCode>
                          <CIP4_Region>AZ</CIP4_Region>
                          <CIP4_Street>Ocean Drive</CIP4_Street>
                        </CIP4_Address>
                        <CIP4_Person>
                          <CIP4_FirstName>Jane</CIP4_FirstName>
                          <CIP4_LastName>Smith</CIP4_LastName>
                        </CIP4_Person>
                    </CIP4_Contact>
                    <CIP4_UniqueID>ID_0</CIP4_UniqueID>
                </CIP4_Recipient>
            </CIP4_Root>
        </DPM>
        <DocPart>
            <DPM>
                <CIP4_Root>
                    <CIP4_Production>
                        <CIP4_Part>
                          <CIP4_ProductType>Cover</CIP4_ProductType>
                        </CIP4_Part>
                    </CIP4_Production>
                </CIP4_Root>
            </DPM>
            <PDFPage/>
            <PDFPage/>
        </DocPart>
        <DocPart>
            <DPM>
                <CIP4_Root>
                    <CIP4_Production>
                        <CIP4_Part>
                          <CIP4_ProductType>Body</CIP4_ProductType>
                        </CIP4_Part>
                    </CIP4_Production>
                </CIP4_Root>
            </DPM>
            <PDFPage/>
            <PDFPage/>
            <PDFPage/>
```

```
        <PDFPage/>
    </DocPart>
</Record>
<Record>
    <DPM>
        <ACME_CustStatus>Prospective</ACME_CustStatus>
        <CIP4_Root>
            <CIP4_Summary>
                <CIP4_PageCount>6</CIP4_PageCount>
            </CIP4_Summary>
            <CIP4_Production>
                <CIP4_CopyCount>1</CIP4_CopyCount>
                <CIP4_Part>
                    <CIP4_ProductType>Brochure</CIP4_ProductType>
                </CIP4_Part>
            </CIP4_Production>
            <CIP4_Recipient>
                <CIP4_Contact>
                    <CIP4_Address>
                      <CIP4_City>Phoenix</CIP4_City>
                      <CIP4_CivicNumber>96</CIP4_CivicNumber>
                      <CIP4_Country>USA</CIP4_Country>
                      <CIP4_PostalCode>81215</CIP4_PostalCode>
                      <CIP4_Region>AZ</CIP4_Region>
                      <CIP4_Street>South Ave</CIP4_Street>
                    </CIP4_Address>
                    <CIP4_Person>
                      <CIP4_FirstName>Mary</CIP4_FirstName>
                      <CIP4_LastName>Smith</CIP4_LastName>
                    </CIP4_Person>
                </CIP4_Contact>
                <CIP4_UniqueID>ID_1</CIP4_UniqueID>
            </CIP4_Recipient>
        </CIP4_Root>
    </DPM>
    <DocPart>
        <DPM>
            <CIP4_Root>
                <CIP4_Production>
                    <CIP4_Part>
                      <CIP4_ProductType>Cover</CIP4_ProductType>
                    </CIP4_Part>
                </CIP4_Production>
            </CIP4_Root>
        </DPM>
        <PDFPage/>
        <PDFPage/>
    </DocPart>
    <DocPart>
        <DPM>
            <CIP4_Root>
                <CIP4_Production>
                    <CIP4_Part>
                      <CIP4_ProductType>Body</CIP4_ProductType>
                    </CIP4_Par   t>
                </CIP4_Production>
            </CIP4_Root>
        </DPM>
        <PDFPage/>
        <PDFPage/>
        <PDFPage/>
        <PDFPage/>
    </DOCPart>
</Record>
<Record>
    <DPM>
        <ACME_CustStatus>Prospective</ACME_CustStatus>
        <CIP4_Root>
```

```
                    <CIP4_Summary>
                        <CIP4_PageCount>6</CIP4_PageCount>
                    </CIP4_Summary>
                    <CIP4_Production>
                        <CIP4_CopyCount>1</CIP4_CopyCount>
                        <CIP4_Part>
                            <CIP4_ProductType>Brochure</CIP4_ProductType>
                        </CIP4_Part>
                    </CIP4_Production>
                    <CIP4_Recipient>
                        <CIP4_Contact>
                            <CIP4_Address>
                              <CIP4_City>Phoenix</CIP4_City>
                              <CIP4_CivicNumber>54</CIP4_CivicNumber>
                              <CIP4_Country>USA</CIP4_Country>
                              <CIP4_PostalCode>81218</CIP4_PostalCode>
                              <CIP4_Region>AZ</CIP4_Region>
                              <CIP4_Street>North Street</CIP4_Street>
                            </CIP4_Address>
                            <CIP4_Person>
                              <CIP4_FirstName>Frederick</CIP4_FirstName>
                              <CIP4_LastName>Adams</CIP4_LastName>
                            </CIP4_Person>
                        </CIP4_Contact>
                        <CIP4_UniqueID>ID_2</CIP4_UniqueID>
                    </CIP4_Recipient>
                </CIP4_Root>
            </DPM>
            <DocPart>
                <DPM>
                    <CIP4_Root>
                        <CIP4_Production>
                            <CIP4_Part>
                              <CIP4_ProductType>Cover</CIP4_ProductType>
                            </CIP4_Part>
                        </CIP4_Production>
                    </CIP4_Root>
                </DPM>
                <PDFPage/>
                <PDFPage/>
            </DocPart>
            <DocPart>
                <DPM>
                    <CIP4_Root>
                        <CIP4_Production>
                            <CIP4_Part>
                              <CIP4_ProductType>Body</CIP4_ProductType>
                            </CIP4_Part>
                        </CIP4_Production>
                    </CIP4_Root>
                </DPM>
                <PDFPage/>
                <PDFPage/>
                <PDFPage/>
                <PDFPage/>
            </DocPart>
        </Record>
    </Root>
</PDFVT>
```

Annex E
(normative)

Naming conventions for DPM keys

All keys used in or referenced from a DPM dictionary of a conforming PDF/VT file, that are not defined in *PDF Reference*, shall have a second class PDF name prefix as defined in *PDF Reference*. Each second class PDF name prefix shall be registered in accordance with the recommendation specified in *PDF Reference*, Appendix E.

A conforming writer that writes a key with a registered second-class name prefix should abide by all requirements, restrictions and recommendations set forth for that key by the registrant of that second-class name prefix.

New keys, either public or private, should follow existing naming conventions for attribute and element names, where such conventions exist in JDF or other job ticket specifications.

NOTE 1 Keys in a DPM dictionary other than **GTS_Managed** and **GTS_Suspect** have no intrinsic meaning as defined by this part of ISO 16612 and the mere presence of those keys does not invoke any action in a conforming reader. A conforming reader is not required to support any DPM keys. A conforming writer can therefore not rely on the support for any DPM keys by a conforming reader. However, a conforming reader might be able to extract the values assigned to keys supported by that reader and use those extracted values by a job ticket or system settings. Any action taken as a response to the presence of a DPM key and value is defined by the system setting or job ticket.

NOTE 2 The **GTS_ second class name prefix** is registered by CGATS and is used in PDF-based ISO standards. The **GTS_ second class name prefix** is registered by ANSI/CGATS and is used in ISO standards that build on the *PDF Reference*. Each individual standard includes the requirements, restrictions and recommendations on the use of keys starting with **GTS_** within that standard.

NOTE 3 CIP4 has defined a set of standardized metadata keys and values in the ICS *Common Metadata for Document Production Workflows* available from www.CIP4.org. It is anticipated that additional registries, technical reports and/or application notes will be developed that define DPM keys. Any such documents prepared under the ISO process will be available from ISO and its national member bodies.

Annex F
(normative)

Determining XObject equivalence

F.1 General considerations

This annex describes a comparison technique for PDF objects which allows uniqueness between a pair of **XObject**s to be determined**.**

An **XObject** is an object of type *stream*. A stream contains a *dictionary* object. A *dictionary* is a collection of key/value pairs, where a value can be of any PDF object type. Therefore, this annex is applicable to the comparison of any PDF object type.

F.2 Comparison techniques for determining XObject equivalence

F.2.1 General

In order to compare two PDF objects, the first step is to determine if the objects are of the same type (e.g. stream). If they are not, then they are not the same. If they are the same type, the further comparison process differs based on the type of object.

F.2.2 Boolean objects

PDF Reference, 3.2.1, clearly states that the keywords shall be *true* and *false*. Comparison is a simple boolean logic test.

F.2.3 Numeric objects

As described in *PDF Reference*, 3.2.2, PDF supports two types of numeric objects, i.e. integer and real. Integers are well defined and their comparison shall be done using standard integer mathematical principles. Real numbers shall be interpreted and compared using the internal representation of the conforming reader. In the case where an integer is being compared to a real, the integer shall be promoted to a real and then comparison shall take place.

F.2.4 String objects

Although *PDF Reference*, 3.2.3, defines two types of strings (literal and hexadecimal), when processing them for comparison, they should first be processed into a single canonical form and then compared in a simple binary comparison.

Literal strings shall be processed looking for any escape characters (*PDF Reference*, 3.2.3, Table 3.2) and replacing them with their meanings. Any strings that are split across multiple lines shall be processed in accordance with *PDF Reference*'s instructions to “disregard the REVERSE SOLIDUS and the end-of-line marker following it when reading the string (i.e. the resulting string value shall be identical to that which would be read if the string were not split)”. The result of these expansions shall be the value to be used for binary comparison.

Hexadecimal strings simply shall be converted from hexadecimal format to literal format, and that value shall be used for the binary comparison.

F.2.5 Name objects

A name object is a special type of string object, where the string is defined as being a sequence of any characters (8-bit values) except null (character code 0), with any values outside the range of EXCLAMATION MARK to TILDE being escaped using a NUMBER SIGN and hexadecimal digits (*PDF Reference*, 3.2.4). All such escaping shall be expanded and the result shall form a sequence of bytes that shall be used for binary comparison.

F.2.6 Array objects

To compare two array objects, each object (N) of the first array shall be compared with the same indexed object in the second array as shown in Figure F.1. Since any of the objects in the array can be a dictionary or an array, a form of "tree comparison" needs to be used, as processing can go from array to dictionary to array, etc.

If, at any time, two objects do not match, then the arrays shall be considered not equal.

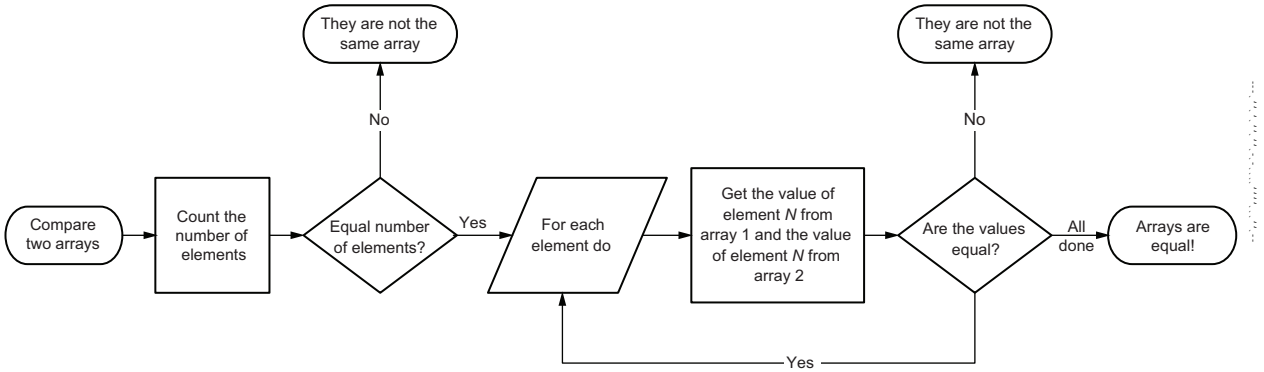


Figure F.1 — Typical array object comparison

F.2.7 Dictionary objects

The flexibility of dictionary objects in PDF makes the comparison of dictionaries a more complex operation.

First, the order of keys in a dictionary is not defined. They can be alphabetically sorted, sorted by entry into the dictionary, or in any other order that a producer chooses. This, of course, leads to the possibility that the same key/value pair exists in both dictionaries but in a different order, so a binary compare or crypto-hash will not be successful.

This lack of a standard order for keys in a dictionary requires that comparison be done to ensure that the number of keys is equal and then iterating over each key in the first dictionary and comparing its value to the value of the key of the same name in dictionary 2. If a key from dictionary 1 cannot be found in dictionary 2, then the dictionaries are not equal. However, if each key exists in both and they have the same associated values, then the dictionaries are equal.

Second, unlike Postscript where a dictionary value of null is considered a valid value, PDF stipulates that "A dictionary entry whose value is null shall be treated the same as if the entry does not exist" (*PDF Reference*, 3.2.6). To ease comparison, any such key/value pairs shall be removed from a dictionary prior to processing.

Finally, since any of the values can itself be a dictionary (or an array), there is the need to be able to do a form of "tree comparison" since processing can go from dictionary to dictionary to dictionary, etc.

Fortunately, even with these complexities, the process for dictionary comparison can be clearly defined and made standard. A typical dictionary object comparison is shown in Figure F.2.

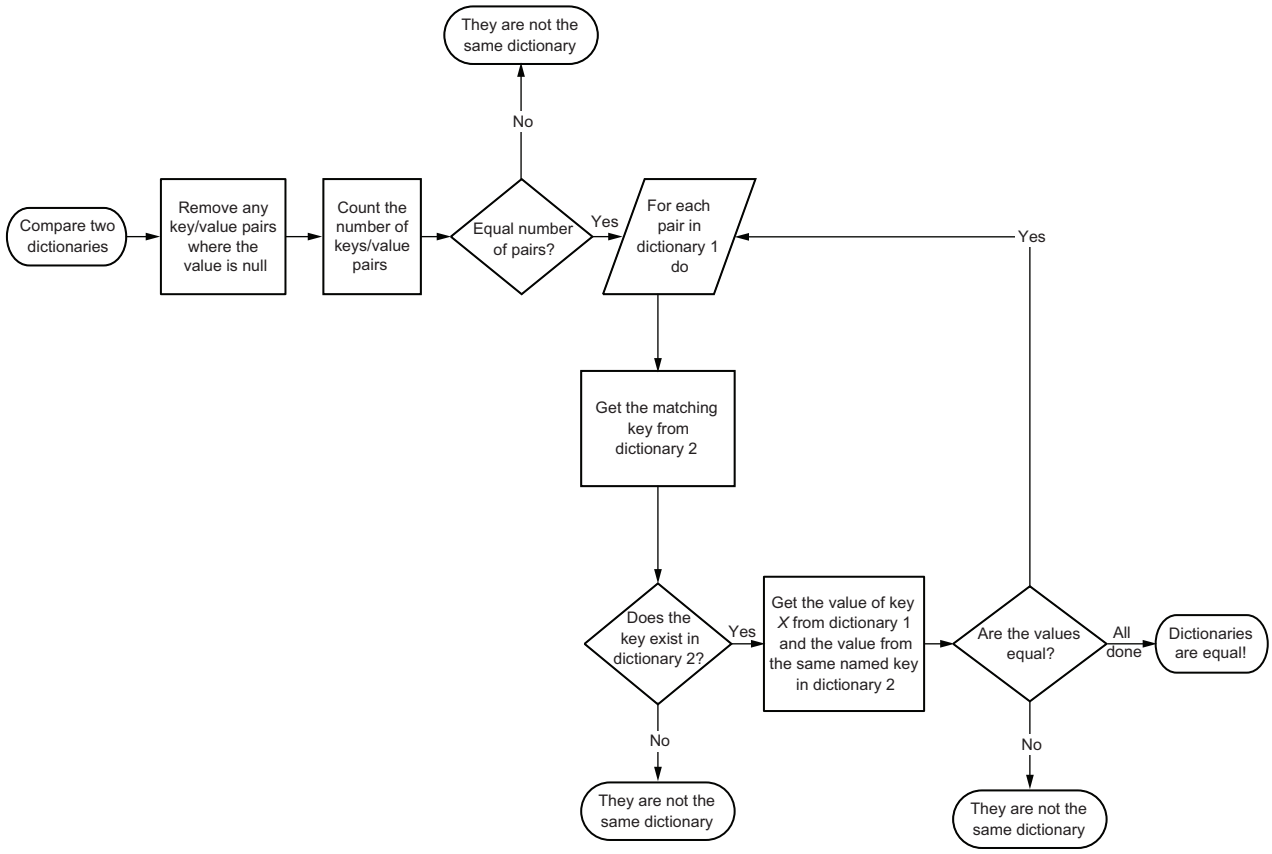


Figure F.2 — Typical dictionary object comparison

F.2.8 Stream Objects

In comparing streams, both the stream data and the associated dictionary need to be compared. Comparison of the dictionary was addressed in F.2.7.

The stream of data in a PDF file can have had one or more filters applied to it (*PDF Reference*, 3.3, Table 3.5) that compress or encode the stream's data. In order to compare the streams, all filters shall be removed and the original (unfiltered) data shall be used for all comparisons. The actual comparison of the data stream shall be done using a simple linear binary (byte-for-byte) comparison. A typical stream object comparison is shown in Figure F.3.

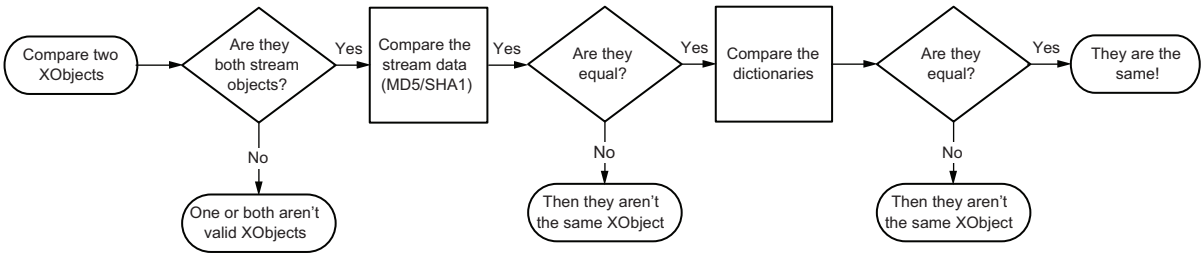


Figure F.3 — Typical stream object comparison

F.3 Known issues

F.3.1 Equal versus equivalent

One issue with this particular solution for comparison of **XObject** equality is that it is not suitable for the comparison of two functionality equivalent but binary different **XObject**s.

In the case where a PDF optimization process has optimized the data of the stream itself, for example converting all the end-of-line characters (e.g. CARRIAGE RETURN or LINE FEED) in the data of the stream to simpler SPACE characters in order to improve compression. The streams are equivalent, i.e. in that they will produce the exact same content on the page, but they will never compare correctly via binary comparison or hash generation.

In addition, a similar optimization process can remove any private data or metadata associated with an object. Doing so would leave an object with the same data stream but with a different dictionary, and thus a different object.

F.3.2 Recursion in dictionaries

As noted in the section on dictionary objects, it is possible that, when processing a dictionary, the value of one or more keys can, itself, be a dictionary that needs to be processed; that dictionary can then refer to another dictionary, and so on. If one of those dictionaries references an object “higher up” in the document's object structure, this has the potential to lead to an infinite recursive situation when doing the comparison.

For the comparison of **XObject**s, it is not expected that this will occur. However, if the recommendations in this document are used in the general case, then the reader will need to protect itself.

Bibliography

[1] ISO 15930-1, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 1: Complete exchange using CMYK data (PDF/X-1 and PDF/X-1a)*

[2] ISO 15930-3, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 3: Complete exchange suitable for colour-managed workflows (PDF/X-3)*

[3] ISO 15930-4, *Graphic technology — Prepress digital data exchange using PDF — Part 4: Complete exchange of CMYK and spot colour printing data using PDF 1.4 (PDF/X-1a)*

[4] ISO 15930-6, *Graphic technology — Prepress digital data exchange using PDF — Part 6: Complete exchange of printing data suitable for colour-managed workflows using PDF 1.4 (PDF/X-3)*

[5] ISO 19005-1, *Document management — Electronic document file format for long-term preservation — Part 1: Use of PDF 1.4 (PDF/A-1)*

[6] ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

[7] ISO/IEC 9541-1, *Information technology — Font information interchange — Part 1: Architecture*

[8] ISO/IEC 11578*, Information technology — Open Systems Interconnection — Remote Procedure Call (RPC)*

[9] *Application Notes for PDF/X Standards*, version 4, September 2006. Available from <http://www.npes.org/standards/toolspdfx.html>

[10] ICC.1:1998-09, *File Format for Color Profiles*, International Color Consortium. Available from <http://www.color.org/>

[11] ICC.1:2001-12, *File Format for Color Profiles*, *version 4*, International Color Consortium. Available from <http://www.color.org/>

[12] ICC.1:2004-10 *Image technology colour management — Architecture, profile format, and data structure,* International Color Consortium. Available from <http://www.color.org/>

[13] ICC.1:2003-09, *File Format for Color Profiles*, International Color Consortium. Available from <http://www.color.org/>

[14] RFC 1738, *Uniform Resource Locators (URL),* December 1984. Available from http://www.ietf.org/rfc/rfc1738.txt

[15] *JDF Specification,* CIP4 Organization, Available from <http://www.CIP4.org>

[16] ISO 15076-1, *Image technology colour management — Architecture, profile format and data structure — Part 1: Based on ICC.1:2004-10*

[17] *ICS — Common Metadata for Document Production Workflows,* CIP4, August 2010. Available from <http://www.CIP4.org>

ICS 35.240.30; 37.100.99

Price based on 36 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the independent national body responsible for preparing British Standards and other standards-related publications, information and services.
It presents the UK view on standards in Europe and at the international level.

It is incorporated by Royal Charter.

Revisions

British Standards are updated by amendment or revision. Users of British Standards should make sure that they possess the latest amendments or editions.

It is the constant aim of BSI to improve the quality of our products and services. We would be grateful if anyone finding an inaccuracy or ambiguity while using this British Standard would inform the Secretary of the technical committee responsible, the identity of which can be found on the inside front cover.

Tel: +44 (0)20 8996 9001 Fax: +44 (0)20 8996 7001

BSI offers Members an individual updating service called PLUS which ensures that subscribers automatically receive the latest editions of standards.

Tel: +44 (0)20 8996 7669 Fax: +44 (0)20 8996 7001
Email: plus@bsigroup.com

Buying standards

You may buy PDF and hard copy versions of standards directly using a credit card from the BSI Shop on the website **www.bsigroup.com/shop.** In addition all orders for BSI, international and foreign standards publications can be addressed to BSI Customer Services.

Tel: +44 (0)20 8996 9001 Fax: +44 (0)20 8996 7001
Email: orders@bsigroup.com

In response to orders for international standards, it is BSI policy to supply the BSI implementation of those that have been published as British Standards, unless otherwise requested.

Information on standards

BSI provides a wide range of information on national, European and international standards through its Knowledge Centre.

Tel: +44 (0)20 8996 7004 Fax: +44 (0)20 8996 7005
Email: knowledgecentre@bsigroup.com

Various BSI electronic information services are also available which give details on all its products and services.

Tel: +44 (0)20 8996 7111 Fax: +44 (0)20 8996 7048
Email: info@bsigroup.com

BSI Subscribing Members are kept up to date with standards developments and receive substantial discounts on the purchase price of standards. For details of these and other benefits contact Membership Administration.

Tel: +44 (0)20 8996 7002 Fax: +44 (0)20 8996 7001
Email: membership@bsigroup.com

Information regarding online access to British Standards via British Standards Online can be found at **www.bsigroup.com/BSOL**

Further information about BSI is available on the BSI website at **www.bsigroup.com/standards**

Copyright

Tel: +44 (0)20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Tel +44 (0)20 8996 9001
Fax +44 (0)20 8996 7001
www.bsigroup.com/standards

raising standards worldwide™