The Printer Working Group

Standard for PDF Fax Format (PDFax)

Proposed Standard 510n.y-P0.2

23 October 2002

The Printer Working Group Standard for PDF Fax Format (PDFax) Proposed Standard 510n.y-P0.2

Abstract: This standard specifies a subset of PDF (Portable Document Format) 1.4 known as the PDF Fax Format (PDFax) by formally defining a series of PDFax “profiles” distinguished primarily by the method of image compression employed and color space used.

In summary PDFax is an image document format intended for use by, but not limited to, the IPPFAX protocol, which is used to provide a synchronous, reliable exchange of image Documents between Senders and Receivers. PDFax makes reference to the PDF 1.4 Reference [pdf], which describes the PDF representation of image data specified by the ITU-T Recommendations for black-and-white facsimile (see [T.4], [T.6]), the ISO/IEC Specifications for Digital Compression and Coding of Continuous-Tone Still Images (see [jpeg]), and Lossy/Lossless Coding of Bi-Level Images (see [jbig2]), and the general purpose Flate compression methods (see [RFC1950] and [RFC1951]).

This document is available electronically at:

ftp://pwg.org/pub/pwg/QUALDOCS/pwg-ifx-pdfax-P02-021023.pdf

A version showing the changes from the previous version is available at:

ftp://pwg.org/pub/pwg/QUALDOCS/pwg-ifx-pdfax-P02-021023-rev.pdf

The current version of this document is available at:

ftp://pwg.org/pub/pwg/QUALDOCS/pwg-ifx-pdfax-latest.pdf, .doc

Title: The Printer Working Group Standard for PDF Fax Format

About the IEEE-ISTO

The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and flexible operational forum and support services. The IEEE-ISTO provides a forum not only to develop standards, but also to facilitate activities that support the implementation and acceptance of standards in the marketplace. The organization is affiliated with the IEEE (http://www.ieee.org/) and the IEEE Standards Association (http://standards.ieee.org/).

For additional information regarding the IEEE-ISTO and its industry programs visit http://www.ieee-isto.org.

About the IEEE-ISTO PWG

The Printer Working Group (or PWG) is a Program of the IEEE Industry Standards and Technology Organization (ISTO) with member organizations including printer manufacturers, print server developers, operating system providers, network operating systems providers, network connectivity vendors, and print management application developers. The group is chartered to make printers and the applications and operating systems supporting them work together better. All references to the PWG in this document implicitly mean "The Printer Working Group, a Program of the IEEE ISTO." In order to meet this objective, the PWG will document the results of their work as open standards that define print related protocols, interfaces, procedures and conventions. Printer manufacturers and vendors of printer related software will benefit from the interoperability provided by voluntary conformance to these standards.

In general, a PWG standard is a specification that is stable, well understood, and is technically competent, has multiple, independent and interoperable implementations with substantial operational experience, and enjoys significant public support.

For additional information regarding the Printer Working Group visit: http://www.pwg.org

Contact information:

IFX Web Page: http://www.pwg.org/qualdocs
IFX Mailing List: ifx@pwg.org

To subscribe to the ipp mailing list, send the following email:

1) send it to majordomo@pwg.org
2) leave the subject line blank
3) put the following two lines in the message body:

```
subscribe ifx
end
```

Implementers of this specification are encouraged to join the IFX Mailing List in order to participate in any discussions of clarifications or review of registration proposals for additional names. Requests for additional media names, for inclusion in this specification, should be sent to the IFX Mailing list for consideration.

Contents

1 Introduction 7

2 Terminology 7

2.1 Conformance Terminology 7

2.2 Other Terminology 8

3 PDFax Support 8

3.1.1 Image Profiles 8

3.1.2 Security Profiles 9

3.1.3 Color Profiles 10

3.2 PDF Object Requirements 10

3.3 PDF Field Specification 12

3.3.1 'PDFax' object 12

3.3.2 'FlateDecode' Filter 14

3.3.3 'CCITTFaxDecode' Filter 14

3.3.4 'JBIG2Decode' Filter 15

3.3.5 'DCTDecode' Filter 15

3.3.6 File Trailer 15

3.3.7 Encryption Dictionary 15

3.3.8 Document Catalog 16

3.3.9 Page Tree Nodes 16

3.3.10 Page Objects 17

3.3.11 Content Stream Operators 17

3.3.12 Resource Dictionaries 18

3.3.13 Color Spaces 18

3.3.14 Image XObjects 19

3.3.15 Masked Images 20

3.3.16 Interactive Form Dictionary 20

3.3.17 Annotation Field Dictionary 20

3.3.18 Signature Dictionary 21

3.3.19 Document Information Dictionary 21

3.4 Cached Objects 22

3.4.1 Cache Hold 22

3.4.2 Cache Release 22

3.5 Implementation Details 22

4 Conformance Requirements 22

4.1 Creator conformance requirements 23

4.2 Renderer conformance requirements 23

4.3 File Layout 23

5 Issues 24

6 Sample PDFax PDFs 24

7 Normative References 25

8 Informative References 26

9 Revision History (to be removed when standard is approved) 26

10 Contributors 26

11 Acknowledgments 26
12 Author's Address 26

Table of Tables

Table 3-1: Image Profiles 9
Table 3-2: Security Profiles 9
Table 3-3: Color Profiles 10
Table 3-4: PDF Object Requirements 11
Table 3-5: PDFax Object 12
Table 3-6: PDFax Object 'IMAGES' Element 13
Table 3-7: PDFax Object 'SECURITY' Element 13
Table 3-8: PDFax Object 'COLOR' Element 13
Table 3-9: FlateDecode Filter 14
Table 3-10: CCITTFaxDecode Filter 14
Table 3-11: JBIG2Decode Filter 15
Table 3-12: DCTDecode Filter 15
Table 3-13: File Trailer 15
Table 3-14: Encryption Dictionary 15
Table 3-15: Document Catalog 16
Table 3-16: Page Tree Nodes 16
Table 3-17: Page Objects 17
Table 3-18: Content Stream Operators 17
Table 3-19: Resource Dictionaries 18
Table 3-20: Color Spaces 18
Table 3-21: Image Resolutions 19
Table 3-22: Image XObjects 19
Table 3-23: Masked Images 20
Table 3-24: Interactive Form Dictionary 20
Table 3-25: Annotation Field Dictionary 20
Table 3-26: Signature Dictionary 21
Table 3-27: Document Information Dictionary 21
Table 4-1: File Layout 24

1 Introduction

In summary, PDFax (pronounced “PDF FAX”) is a raster image data format intended for use by, but not limited to, the IPPFAX protocol. IPPFAX is used to provide a synchronous, reliable exchange of image Documents between Senders and Receivers. PDFax makes reference to the PDF 1.4 specification [pdf], which describes the PDF (Portable Document Format) representation of image data specified by the ITU-T Recommendations for black-and-white facsimile (see [T.4], [T.6]), the ISO/IEC Specifications for Digital Compression and Coding of Continuous-Tone Still Images (see [jpeg]), and Lossy/Lossless Coding of Bi-Level Images (see [jbig2]), and the general purpose Flate compression methods (see [RFC1950] and [RFC1951]).

PDFax is a image-only, streamable, subset specification of PDF 1.4 [pdf] and, as such, follows all of the specification requirements except as noted in the “Deviations from PDF” section of this document.

As a streamable version of PDF, it is not required that a Renderer of a PDFax document be able to randomly access the PDF. The format has been adopted in such a way as to allow a Renderer the ability to read the PDFax document from the beginning to end without the necessity to cache more data than is necessary to print the current page.

2 Terminology

This section defines terminology used throughout this document.

2.1 Conformance Terminology

Capitalized terms, such as **MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, NEED NOT, OPTIONAL,** and **PROHIBITED**, have special meaning relating to conformance as defined in RFC 2119 [RFC2119] and [RFC2911] section 12.1. If an implementation supports the extension defined in this document, then these terms apply; otherwise, they do not. These terms define conformance to *this document (and [RFC2911]) only*; they do not affect conformance to other documents, unless explicitly stated otherwise. To be more specific:

REQUIRED (REQ) - an adjective used to indicate that a conforming PDFax Creator or Renderer’s implementation MUST support the indicated operation, object, attribute, or attribute value. See [RFC2911] “Appendix A - Terminology for a definition of “support”.

RECOMMENDED (REC) - an adjective used to indicate that a conforming PDFax Creator or Renderer’s implementation SHOULD support the indicated operation, object, attribute, or attribute value.

OPTIONAL (OPT) - an adjective used to indicate that a conforming PDFax Creator or Renderer’s implementation MAY support the indicated operation, object, attribute, or attribute value.

PROHIBITED (PROH) - an adjective used to indicate that a conforming PDFax Creator or Renderer’s implementation MUST NOT support the indicated operation, object, attribute, or attribute value.

REQUIRED DEPENDENCY (REQ-DEP) – an adjective used to indicate that a conforming PDFax Creator or Renderer’s implementation MUST NOT support the indicated operation, object, attribute, or attribute value unless the Profile(s) in ‘<>’s are also SUPPORTED, in which case it is then REQUIRED.

OPTIONAL DEPENDENCY (OPT-DEP) – an adjective used to indicate that a conforming PDFax Creator or Renderer's implementation MUST NOT support the indicated operation, object, attribute, or attribute value unless the Profile(s) in '<>'s are also SUPPORTED, in which case it is then OPTIONAL.

IGNORED – an adjective used to indicate that a conforming PDFax Creator or Renderer implementation NEED NOT support the indicated operation, object, attribute, or attribute value; but this feature MAY be added to a future version of this specification.

AS SPECIFIED – is used to indicate that a conforming PDFax Creator or Render implementation MUST, MAY, or MUST NOT support the indicated operation, object, attribute, or attribute value as is defined in the indicated specification.

OR – a conjunction that specifies a logical 'or', implying that a choice of one or more of the choices specified.

XOR – a conjunction that specifies a logical 'exclusive or', implying that a choice of one and only one of the choices specified.

AND – a conjunction that specifies a logical 'and', implying a selection of all choices specified.

2.2 Other Terminology

The following terms are introduced and capitalized in order to indicate their specific meaning:

Implement – The specified feature is present in the Document.

Support – A Creator has the capability of Implementing the feature specified, or the Renderer has the capability of understanding and acting on the Implementation.

Document – The PDFax-formatted electronic representation of a set of one or more pages that the Sender sends to the Receiver.

Renderer – This is the agent (software, hardware or some combination) that converts the Document into a displayed or printed form.

Creator -- This is the agent (software, hardware or some combination) that creates the Document.

Interpolation – See 'Interpolation' in [pdf] pg. 273.

Forward-Reference – In indirect object reference (See [pdf] Section 3.2.9) to an object that appears later in the Document.

3 PDFax Support

3.1.1 Image Profiles

The following tree diagram shows the relationship among PDFax Image Profiles:

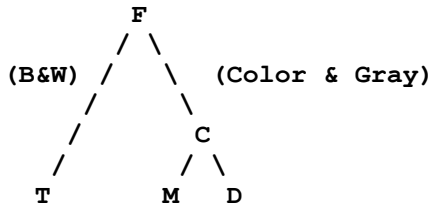


Table 3-1: Image Profiles

Profile	Image Implementation	Reference
'F'	'CCITTFaxDecode' Filter	[pdf] Section 3.3.5
'D'	'FlateDecode' Filter	[pdf] Section 3.3.3
'T'	'JBIG2Decode' Filter	[pdf] Section 3.3.6
'M'	Masked Images	[pdf] Section 4.8.5
'C'	'DCTDecode' Filter	[pdf] Section 3.3.7
'P'	Single Image	(See below)

All PDFax Renderers and Creators MUST Support PDFax Profile 'F', which is the root node of the tree. All color OR gray scale image Renderers and Creators of PDFax MUST Support PDFax Profile 'C'. Creators and Renderers that Support a particular profile MUST also Support those profiles on the path that connect it to the root node, and MAY optionally Support profiles not on the path connecting it to the root node. For example, a Creator or Renderer that Supports PDFax Profile 'D' MUST also Support PDFax Profiles 'C' AND 'F', and MAY optionally Support PDFax Profile 'M', OR 'T'. For another example, a Creator or Renderer that Supports PDFax Profile 'C' MUST also Support PDFax Profile 'F', AND MAY optionally Support PDFax Profile 'T'.

Single Image:
This profile indicates that the file has a single page with a single (possibly masked) image. The Document SHOULD specify this Profile if all of the following are true:

- The Document Implements only one 'Page Object'.
- The 'Content Stream' for the page Implements only one '**cm**' operator.
- The Document does not Implement Profile '1', nor Profile '2'; see below.

3.1.2 Security Profiles

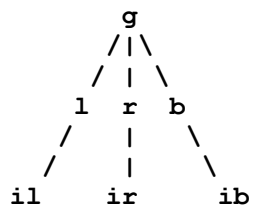
There are several options that MAY be Supported by a Creator or Renderer with regard to security:

Table 3-2: Security Profiles

Profile	Security Implementation	Reference
'1'	'Standard' Encryption	[pdf] Section 3.5.2
'2'	'PPKLite' Encryption	[pdf-ppk] Section 3
'3'	Digital Signature	[pdf-ppk] Section 2.2

3.1.3 Color Profiles

The following tree diagram shows the relationship among PDFax Color Profiles:



There are several color spaces that may be Supported by a Creator or Renderer. These Profiles only apply to Creators or Renderers that Support Image Profiles ‘C’ or ‘D’. All PDFax Renderers and Creators that Support Image Profiles ‘C’ OR ‘D’ MUST Support PDFax Color Profiles ‘g’ AND ‘r’. Other Color Profiles are OPTIONAL. Creators and Renderers that Support a particular profile MUST also Support those profiles on the path that connect it to the root node, and MAY optionally Support profiles not on the path connecting it to the root node. For example, a Creator or Renderer that Supports PDFax Profile ‘ib’ MUST also Support PDFax Profiles ‘b’ AND ‘g’, and MAY optionally Support PDFax Profile ‘l’, OR ‘r’, OR ‘il’, OR ‘ic’.

Table 3-3: Color Profiles

Profile	Color Space Implementation	Reference
‘g’	‘CalGray’	[pdf] Page 182
‘r’	‘CalRGB’	[pdf] Page 184
‘l’	‘Lab’	[pdf] Page 187
‘b’	‘ICCBased’	[pdf] Page 189
‘il’	‘Indexed’ AND ‘Lab’	[pdf] Page 199, 187
‘ir’	‘Indexed’ AND ‘CalRGB’	[pdf] Page 199, 184
‘ib’	‘Indexed’ AND ‘ICCBased’	[pdf] Page 199, 189

‘ICCBased’ and ‘Indexed’ Color Profiles SHOULD be compressed using ‘FlateDecode’ Filter to minimize Document size (See [pdf] Section 3.3.3). If ‘FlateDecode’ is used in this manner, Profile ‘D’ MUST be specified as being used in the Document.

3.2 PDF Object Requirements

For the table shown below, if an Object/Filter is not Implemented then its associated Profile is not Implemented.

Key:

Requirement: Applies to both the Creator and the Renderer of the Document.

Profile: If the indicated ‘PDF Object/Filter’ is Implemented then the Document Implements the indicated Profile.

Dependencies: In order to Implement the ‘PDF Object/Filter’ the Profiles indicated in the Dependencies column MUST also be implemented. Note that a comma ‘,’ in this column indicates an ‘AND’.

358

Table 3-4: PDF Object Requirements

PDF Object/Filter	Requirement	Profile	Dependencies	Reference
'ASCIIHexDecode' Filter	PROH			[pdf] Section (3.3.1)
'ASCII85Decode' Filter	PROH			[pdf] Section (3.3.2)
'LZWDecode' Filter	PROH			[pdf] Section (3.3.3)
'RunLengthDecode' Filter	PROH			[pdf] Section (3.3.4)
Incremental Updates	PROH			[pdf] Section (3.4.5)
Functions	PROH			[pdf] Section (3.9)
Files	PROH			[pdf] Section (3.10)
Graphics State	PROH			[pdf] Section (4.3)
Path objects	PROH			[pdf] Section (4.4)
'DeviceGray' Color Space	PROH			[pdf] Section (4.5.3)
'DeviceRGB' Color Space	PROH			[pdf] Section (4.5.3)
'DeviceCMYK' Color Space	PROH			[pdf] Section (4.5.3)
Pattern Color Space	PROH			[pdf] Section (4.5.5)
Separation Color Space	PROH			[pdf] Section (4.5.5)
DeviceN Color Space	PROH			[pdf] Section (4.5.5)
Pattern Objects	PROH			[pdf] Section (4.6)
Inline Image Objects	PROH			[pdf] Section (4.8.6)
Form Xobjects	PROH			[pdf] Section (4.9)
Postscript Xobjects	PROH			[pdf] Section (4.10)
Text Objects	PROH			[pdf] Section (5)
Transparency	PROH			[pdf] Section (7)
'CCITTFaxDecode' Filter	REQ	F		[pdf] Section (3.3.5)
File Header	REQ			[pdf] Section (3.4.1)
Cross-Reference Table	REQ			[pdf] Section (3.4.3)
File Trailer	REQ			[pdf] Section (3.4.4)
Document Catalog	REQ			[pdf] Section (3.6.1)

Page Tree Nodes	REQ			[pdf] Section (3.6.2)
Page Objects	REQ			[pdf] Section (3.6.2)
Content Streams	REQ			[pdf] Section (3.7.1)
Resource Dictionaries	REQ			[pdf] Section (3.7.2)
Image XObjects	REQ			[pdf] Section (4.8)
'FlateDecode' Filter	OPT	D	C	[pdf] Section (3.3.3)
'JBIG2Decode' Filter	OPT	T		[pdf] Section (3.3.6)
'DCTDecode' Filter	OPT	C	g,r	[pdf] Section (3.3.7)
Encryption Dictionary 'Standard' Encryption	OPT	1		[pdf] Section (3.5)
Encryption Dictionary 'PPKLite' Encryption	OPT	2	1	[pdf-ppk] Section (3)
'CalGray' Color Space	OPT	g	C	[pdf] pg. 182
'CalRGB' Color Space	OPT	r	C	[pdf] pg. 184
'Lab' Color Space	OPT	l	C	[pdf] pg. 187
'ICCBased' Color Space	OPT	b	C	[pdf] pg. 189
'Indexed' Color Space	OPT	i	l OR r OR b	[pdf] pg. 199
Masked Images	OPT	M	C	[pdf] Section (4.8.5)
Interactive Form Dictionary AND Annotation Field Dictionary AND Signature Dictionary	OPT	3		[pdf] Section (8.6.1-3) [pdf-ppk] Section (2)

3.3 PDF Field Specification

The following list describes the object field values of the REQUIRED and OPTIONAL PDF objects in PDFax. The numbers in '( )'s refer to section numbers in the PDF Specifications [pdf], unless otherwise noted. 'AS SPECIFIED' refers to [pdf] unless otherwise noted.

3.3.1 'PDFax' object

A new 'PDF Name Registry' (See [pdf] – Appendix E) object that is REQUIRED for a PDFax document. The existence of this dictionary object is the one and only way to determine if the PDF in question is a PDFax. Spec:

Table 3-5: PDFax Object

KEY	TYPE	VALUE
PDFax	Array of Numeric Objects	[**IMAGES SECURITY COLOR MEMORY**]

See [pdf] Section 3.2.5 for definition of an 'Array Object'. See [pdf] Section 3.2.2 for definition of a 'Numeric Object'.

Where:

IMAGES, SECURITY, COLOR: Each is a 'Numeric Integer Object' ([pdf] Section 3.2.2) that is the sum of all of the Integer equivalents of the binary 'Bit Positions' indicated in the appropriate table, for the Profiles that are Implemented in the Document. The 'Bit Positions' are numbered from 1 (low-order) to 32 (high-order). A '1' in a 'Bit Position' indicates the Profile is Implemented. Note that PDF Numeric Integer Objects in fact are represented in signed twos-complement form.

For example, to indicate that Profiles 'D' (100 binary) and 'M' (10000 binary) are Implemented, the value of '20' (10100 binary) should be used as the value for the 'IMAGES' field.

Table 3-6: PDFax Object 'IMAGES' Element

Profile	Bit Position
F	1
T	2
D	3
C	4
M	5
P	6

Table 3-7: PDFax Object 'SECURITY' Element

Profile	Bit Position
1	1
2	2
3	3

Table 3-8: PDFax Object 'COLOR' Element

Profile	Bit Position
g	1
r	2
l	3
b	4
i	5

All Profiles that are to be indicated as Implemented MUST have their associated 'Value' summed together and recorded in the indicated element ('IMAGES', XOR 'SECURITY', XOR 'COLOR') of the 'PDFax' array. For example, if the Creator wishes to indicate that Color Profile's 'r' and 'b' are Implemented, the value of '10' (10(2) + 1000(8)) MUST be written in the 'COLOR' PDFax array element.

The Creator of the Document MUST NOT Implement a Profile that is not indicated in this field. The Creator of the Document MAY Implement all Profiles indicated in this field, but is NOT REQUIRED.

Rationale: Since this object must be Implemented at the beginning of the Document, it may not be known for certain which Profiles will be Implemented. This field is an advisory indicator to a Renderer as to which Profiles they MUST Support in order to be able to render the Document for certain. If all Profiles indicated are not Supported, the Document may still be rendered if a non-Supported Profile is indicated but is not actually Implemented in the Document.

Note that even though a Profile is higher in the Image Profile tree it SHOULD NOT be indicated in this object unless that feature is Implemented in the document. For example, if the document contained 'Flate' (FlateDecode) images but no 'JPEG' (DCTDecode) images, only Profile 'D' should be indicated.

MEMORY: A 'Numeric Object' that is the decimal value of the minimum amount of cache memory the Renderer will need to cache all objects necessary to render any particular page.

The value specified for 'MEMORY' is in addition to a base memory requirement of 2 Megabytes (2^21 bytes)

An example of the PDFax object for a Document containing a CalRGB color space (Profile 'g'), masked (Profile 'M'), JPEG image (Profile 'C') that's Standard encrypted (Profile '1') would look like this:

```
1 0 obj
<<
/PDFax [24 1 1 0]
>>
endobj
```

3.3.2 'FlateDecode' Filter

See [pdf] Section 3.3.3, [RFC1950], and [RFC1951].

Table 3-9: FlateDecode Filter

Field	Specification
<All Fields>	AS SPECIFIED

3.3.3 'CCITTFaxDecode' Filter

See [pdf] Section 3.3.5, [T.4], and [T.6]. Note that only Group 4 images are Supported by PDFax, see 'K', below.

Table 3-10: CCITTFaxDecode Filter

Field	Specification
'K'	MUST have a value of -1.
'EndOfLine'	AS SPECIFIED
'EncodedByteAlign'	AS SPECIFIED
'Columns'	AS SPECIFIED
'Rows'	AS SPECIFIED
'EndOfBlock'	AS SPECIFIED
'BlackIs1'	AS SPECIFIED

‘DamagedRowsBeforeError’	AS SPECIFIED

3.3.4 ‘JBIG2Decode’ Filter

See [pdf] Section 3.3.6, and [jbig2].

Table 3-11: JBIG2Decode Filter

Field	Specification
<All Details>	AS SPECIFIED

3.3.5 ‘DCTDecode’ Filter

See [pdf] Section 3.3.7, [ps-pdf], [ps], and [jpeg].

Table 3-12: DCTDecode Filter

Field	Specification
<All Details>	AS SPECIFIED

3.3.6 File Trailer

See [pdf] Table 3.12.

Table 3-13: File Trailer

Field	Specification
‘Size’	AS SPECIFIED
‘Prev’	PROHIBITED
‘Root’	AS SPECIFIED
‘Encrypt’	AS SPECIFIED
‘Info’	AS SPECIFIED
‘ID’	MUST use a pseudo-random number in place of ‘File Size’ when generating this value. See [pdf] Section 9.3. Rationale: This is due to the requirements of using this field in generating the encryption key for the ‘standard encryption’ algorithm ([pdf] Step 5 of Algorithm 3.2, pg. 78): file size will not be known at the time this field is needed.

3.3.7 Encryption Dictionary

See [pdf] Table 3.13 and [pdf-ppk] Table 3.

Note that if a Document is Standard encrypted (Profile ‘1’), the ‘ID’ field of the File Trailer MUST be calculated before the Encryption Dictionary is written. The ‘ID’ MUST then be cached until the ‘File Trailer’ is written.

Table 3-14: Encryption Dictionary

Field	Specification
‘Filter’	MUST have a value of either ‘Standard’ or ‘Adobe.PPKLite’.

‘V’	MUST have a value of ‘2’.
‘Length’	AS SPECIFIED
‘R’	AS SPECIFIED
‘O’	AS SPECIFIED but REQ-DEP <1>
‘U’	AS SPECIFIED but REQ-DEP <1>
‘P’	AS SPECIFIED but REQ-DEP <1>
‘SubFilter’	MUST have a value of ‘adbe.pkcs7.s4’, but REQ-DEP <2>
‘Recipients’	AS SPECIFIED but REQ-DEP <1>

3.3.8 Document Catalog

See [pdf] Table 3.16.

Table 3-15: Document Catalog

Field	Specification
‘Type’	AS SPECIFIED
‘Version’	AS SPECIFIED
‘Pages’	AS SPECIFIED
‘PageLabels’	IGNORED
‘Names’	IGNORED.
‘Dests’	IGNORED.
‘ViewerPreferences’	IGNORED.
‘PageLayout’	IGNORED.
‘PageMode’	IGNORED.
‘Outlines’	IGNORED.
‘Threads’	IGNORED.
‘OpenAction’	IGNORED.
‘AA’	IGNORED.
‘URI’	IGNORED.
‘AcroForm’	AS SPECIFIED but REQ-DEP <3>
‘Metadata’	IGNORED.
‘StructTreeRoot’	IGNORED.
‘MarkInfo’	IGNORED.
‘Lang’	IGNORED.
‘SpiderInfo’	IGNORED.
‘OutputIntents’	PROHIBITED.

3.3.9 Page Tree Nodes

See [pdf] Table 3.17.

Table 3-16: Page Tree Nodes

Field	Specification
‘Type’	AS SPECIFIED
‘Parent’	AS SPECIFIED
‘Kids’	AS SPECIFIED
‘Count’	AS SPECIFIED

3.3.10 Page Objects

See [pdf] Table 3.18.

Table 3-17: Page Objects

Field	Specification
‘Type’	AS SPECIFIED
‘Parent’	AS SPECIFIED
‘LastModified’	AS SPECIFIED
‘Resources’	MUST NOT be inherited
‘MediaBox’	MUST NOT be inherited
‘CropBox’	MUST NOT be inherited
‘BleedBox’	AS SPECIFIED
‘TrimBox’	AS SPECIFIED
‘ArtBox’	AS SPECIFIED.
‘BoxColorInfo’	PROHIBITED.
‘Contents’	AS SPECIFIED.
‘Rotate’	MUST NOT be inherited
‘Group’	PROHIBITED.
‘Thumb’	IGNORED.
‘B’	IGNORED.
‘Dur’	IGNORED.
‘Trans’	IGNORED.
‘Annots’	IGNORED.
‘AA’	IGNORED.
‘Metadata’	IGNORED.
‘PieceInfo’	IGNORED.
‘StructParents’	IGNORED.
‘ID’	IGNORED.
‘PZ’	IGNORED.
‘SeparationInfo’	PROHIBITED.
‘Type’	AS SPECIFIED

3.3.11 Content Stream Operators

See [pdf] Table 4.1.

Table 3-18: Content Stream Operators

Field	Specification	Reference
‘q’	AS SPECIFIED	[pdf] Table 4.7
‘Q’	AS SPECIFIED	[pdf] Table 4.7
‘cm’	MUST be [Sx 0 0 Sy Tx Ty], See Below	[pdf] Table 4.7
‘Do’	AS SPECIFIED	[pdf] Table 4.34
‘MP’	IGNORED	[pdf] Table 9.8
‘DP’	IGNORED	[pdf] Table 9.8
‘BMC’	IGNORED	[pdf] Table 9.8
‘BDC’	IGNORED	[pdf] Table 9.8
‘EMC’	IGNORED	[pdf] Table 9.8
‘BX’	AS SPECIFIED	[pdf] Table 3.20
‘EX’	AS SPECIFIED	[pdf] Table 3.20

<All other Operators>	PROHIBITED	

cm: See [pdf] Section 4.2.3.

Given:

W = 'Width' field value in 'Image XObjects'.

H = 'Height' field value in 'Image XObjects'.

R = Resolution of the image in dots per inch

X = Horizontal translation in inches.

Y = Vertical translation in inches.

The following MUST be true:

Sx = (W / R) * 72

Sy = (H / R) * 72

Tx = X * 72

Ty = Y * 72

3.3.12 Resource Dictionaries

See [pdf] Table 3.21.

Table 3-19: Resource Dictionaries

Field	Specification
'ExtGState'	PROHIBITED.
'ColorSpace'	AS SPECIFIED.
'Pattern'	PROHIBITED.
'Shading'	PROHIBITED.
'XObject'	AS SPECIFIED.
'Font'	PROHIBITED.
'ProcSet'	'Text' Proc Sets PROHIBITED, all others AS SPECIFIED.
'Properties'	IGNORED.

3.3.13 Color Spaces

See [pdf] Section 4.5.

Table 3-20: Color Spaces

Field	Specification
'Lab'	AS SPECIFIED
'DeviceGray'	PROHIBITED
'DeviceRGB'	PROHIBITED
'DeviceCMYK'	PROHIBITED
'CalGray'	AS SPECIFIED
'CalRGB'	AS SPECIFIED

‘ICCBased’	AS SPECIFIED, but may be compressed using ‘FlateDecode’ if Profile ‘D’ is indicated in the ‘PDFax Object’.
‘Indexed’	AS SPECIFIED, but may be compressed using ‘FlateDecode’ if Profile ‘D’ is indicated in the ‘PDFax Object’.
‘Pattern’	PROHIBITED
‘Separation’	PROHIBITED
‘DeviceN’	PROHIBITED

3.3.14 Image XObjects

All pixels of all images MUST be square.

Both the Creator and Renderer MUST be capable of creating or rendering a Document with the following minimum resolutions, other resolutions are OPTIONAL.

Table 3-21: Image Resolutions

Profile	Resolution in Dots Per Inch
F	600
T	600
D	300
C	300
M	300

See [pdf] Table 4.35 for description of the following table.

Table 3-22: Image XObjects

Field	Specification
‘Type’	MUST be ‘XObject’
‘Subtype’	MUST be ‘Image’
‘Width’	AS SPECIFIED
‘Height’	AS SPECIFIED
‘ColorSpace’	AS SPECIFIED
‘BitsPerComponent’	AS SPECIFIED
‘Intent’	PROHIBITED.
‘ImageMask’	AS SPECIFIED, if Profile ‘M’
‘Mask’	AS SPECIFIED, if Profile ‘M’
‘SMask’	PROHIBITED.
‘Decode’	AS SPECIFIED.
‘Interpolate’	MUST be ‘true’
‘Alternates’	IGNORED
‘Name’	IGNORED.
‘StructParent’	IGNORED.
‘ID’	IGNORED.
‘OPI’	PROHIBITED.
‘Metadata’	IGNORED.

3.3.15 Masked Images

See [pdf] Section 4.8.5

Table 3-23: Masked Images

Field	Specification
<All Fields>	AS SPECIFIED

3.3.16 Interactive Form Dictionary

See [pdf] Table 8.47.

Table 3-24: Interactive Form Dictionary

Field	Specification
'Fields'	MUST be an indirect object of an 'Annotation Field Dictionary'.
'NeedAppearances'	PROHIBITED
'SigFlags'	MUST be '3'
'CO'	PROHIBITED
'DR'	PROHIBITED
'DA'	PROHIBITED
'Q'	PROHIBITED

3.3.17 Annotation Field Dictionary

See [pdf] Tables 8.10 & 8.49. This dictionary consists of entries from both a 'Annotation Dictionary (Table 8.10) and a 'Field Dictionary' (Table 8.49).

Table 3-25: Annotation Field Dictionary

Field	Specification
'Type'	MUST be 'Annot'
'Subtype'	MUST be 'Widget'
'Contents'	IGNORED
'P'	IGNORED
'Rect'	MUST be '[0 0 0 0]'
'NM'	IGNORED
'F'	IGNORED
'BS'	IGNORED
'Border'	IGNORED
'AP'	IGNORED
'AS'	IGNORED
'C'	IGNORED
'CA'	IGNORED
'T'	IGNORED
'Popup'	IGNORED
'A'	IGNORED
'AA'	IGNORED
'StructParent'	IGNORED

‘FT’	MUST be ‘Sig’
‘Parent’	PROHIBITED.
‘Kids’	PROHIBTED.
‘T’	AS SPECIFIED.
‘TU’	AS SPECIFIED.
‘TM’	IGNORED.
‘Ff’	MUST be ‘1’.
‘V’	MUST be an indirect object of a ‘Digital Signature’.
‘DV’	IGNORED.
‘AA’	IGNORED.

513
514

3.3.18 Signature Dictionary

See [pdf] Table 8.60 and [pdf-ppk] Table 2.

The Digital Signature format MUST only be in the ‘Raw Format’, see [pdf-ppk] Section 2.2.

Table 3-26: Signature Dictionary

Field	Specification
‘Type’	MUST be ‘Sig’
‘Filter’	MUST be ‘Adobe.PPKLite’
‘SubFilter’	MUST be ‘adbe.x509.rsa_sha1’
‘Name’	AS SPECIFIED.
‘Reason’	AS SPECIFIED.
‘Location’	AS SPECIFIED.
‘M’	AS SPECIFIED.
‘ByteRange’	PROHIBITED (Implies all bytes in the Document with the exclusion of the bytes represented by the value of the ‘Cert’ field. See [pdf] for this field)
‘Contents’	AS SPECIFIED.
‘Cert’	AS SPECIFIED.
‘R’	AS SPECIFIED.
‘V’	AS SPECIFIED.
‘ADBE_Build’	AS SPECIFIED.
‘ADBE_AuthType’	AS SPECIFIED.
‘ADBE_PwdTime’	AS SPECIFIED.

519

3.3.19 Document Information Dictionary

See [pdf] Table 9.2.

Table 3-27: Document Information Dictionary

Field	Specification
‘Trapped’	PROHIBITED.
<All other fields>	AS SPECIFIED.

523

3.4 Cached Objects

If an object MAY be used for more than a single page, it may be practical to maintain the object in the Renderer's memory. To accomplish this, the Creator should invoke the 'Cache Hold' mechanism. Once an object is cached, it no longer has to abide by 'Creator Conformance Requirements' 7 and 8 (See Section 4.1).

An object that is held in the Renderers cache by the 'Cache Hold' mechanism MUST be maintained in the cache until one of the following conditions is met:

The 'Cache Release' mechanism is invoked.

The 'Document Catalog' is reached.

3.4.1 Cache Hold

To specify that an object should not be discarded once the current page is rendered, the object to be 'cached' should have the following 'Name Object' ([pdf] Section 3.2.4) in its 'Dictionary' ([pdf] Section 3.2.6):

/PDFax_cache

3.4.2 Cache Release

To release an object from the Renderer's memory; the following 'Name Object' MUST be placed in the 'Page Object' of the first page in which the object is no longer needed. For example, if the object is question was first found on page 1 and was last used on page 3, the 'Cache Release' should occur in the 'Page Object' for page 4.

/PDFax_cache OBJECTS

Where:

OBJECTS: is an array (contained in '[]'s) of indirect object references of the objects that were previously cached and are no longer needed. Indication of an object number that was never cached MUST be ignored.

Example:

```
3 0 obj
/PDFax_cache                  %First object to be cached.
…
endobj
…
7 0 obj                       %Second object to be cached.
/PDFax_cache
…
endobj
…                             %One or more Page objects in between.
45 0 obj
/Type /Page                   %Page object
/PDFax_cache [3 0 R 7 0 R]    %Objects 3 and 7 are no longer needed.
. . .
```

3.5 Implementation Details

4 Conformance Requirements

This section specifies the conformance requirements for Renderers and Creators.

4.1 Creator conformance requirements

In order to conform to this specification, a Document Creator:

1. MUST specify the PDF as being 'PDF 1.4'.

2. MUST place the 'PDFax' object as the first object in the PDF.

3. MUST place any 'Encryption Dictionary' object as the second object in the PDFax Document, if the Document is encrypted.

4. MUST NOT include any private 'PDF Name Registry' values/objects (See [pdf] – Appendix E) that effect printed output.

5. MUST place the objects: 'Interactive Form Dictionary', 'Field Dictionary' and 'Digital Signature' object as the last three objects (in that order) in the Document, if the Document is Digitally Signed. Note that in a situation where the Renderer cannot cache the entire document before rendering, the detection of a valid or invalid Digital Signature will only occur after rendering of the entire Document.

6. MUST ensure that each non-IGNORED object have at least one Forward-Reference to such object. Objects that do not have to follow this rule are: the 'PDFax Object', 'Encryption Dictionary', all 'Page Objects', the 'Document Information Dictionary', and the 'Document Catalog', Rationale: This will aid the Renderer with knowing which objects will need to be cached and which can be ignored.

7. MUST ensure that all non-IGNORED objects appear in the PDF AFTER the 'Page Object' in which they are first referenced (Satisfied by Requirement 7) and BEFORE the next 'Page Object' unless the object is a Cached Object (See Section 3.4).

4.2 Renderer conformance requirements

In order to conform to this specification, a Document Renderer:

1. MUST Support all of the REQUIRED PDFax objects.

2. MUST cache all REQUIRED or Supported OPTIONAL objects as they are encountered (sequentially) in the Document until the next 'Page Object' is encountered. At that point, the page can be rendered and the cache emptied of all non-Cached objects.

3. MUST Interpolate images up or down in resolution, as required, to match the Renderer's Supported image resolution(s).

4. MAY ignore all IGNORED objects that the Creator added to the PDFax Document.

4.3 File Layout

Given that a Document is fully compliant with this specification, a PDFax Document will, nominally, take on the following format:

Table 4-1: File Layout

	Object
A	Header
B	Encryption Object (if Profile '1' XOR '2')
C	Page object for page 1
D	Resources for page 1
E	Content object for page 1
F	Color Space(s) for page 1 (if Profile D OR C)
G	Image Mask(s) for page 1 (if Profile M)
H	Image XObject(s) for page 1
I	[Repeat C – H for all remaining pages, in order]
J	Document Catalog
K	Page Node(s)
L	Interactive Form Dictionary (if Profile '3')
M	Annotation Field Dictionary (if Profile '3')
N	Signature Dictionary (if Profile '3')
O	File Trailer

5 Issues

- Should we allow non-square image resolutions?
- What should be the minimum image resolutions for JPEG, JBIG2, CCITT, and Flate or does this document even need to specify?
- Should the Creator be allowed to produce a JBIG2 image for multiple pages to optimize compression? Decode memory requirements on the Renderer must be know in advance. JBIG2Globls MUST appear BEFORE the image data. (See [T.89]) Memory requirement levels: 1 Meg, 2 Meg, Unspecified (See [T.30] Table 2, bits 117, 118).
- Should Support for specific JBIG2 profiles be called out in the specification or is support for all JBIG2 profiles more prudent?

6 Sample PDFax PDFs

The 'source' of all of the sample documents in this section can be viewed with any text editor but should only be modified with a binary editor, as the stream data contained therein is not compatible with text editors. Comments on the format of the documents are contained within the documents themselves.

All of the samples are different versions of the same document.

1: The first sample is an unencrypted, single page, 'CCITTFaxDecode' masked, 'DCTDecode' color ICCBased color space foreground image with a 'FlateDecode' gray scale Indexed ICCBased color space background image. The images use 'FlateDecode' compression on the 'ICCBased' and 'Indexed' Color Spaces.

ftp://pwg.org/pub/pwg/QUALDOCS/SamplePDFax/base-02.pdf

2: The next sample has been encrypted with 'Standard' encryption. The 'user' password is '12345'; the 'owner' password is '54321'. The document has also been Digitally Signed: the

document will fail a digital signature check since it has been tampered with. To see the digital signature in Acrobat (or Acrobat Reader), select the 'Signature' tab on the left side of the screen.
ftp://pwg.org/pub/pwg/QUALDOCS/SamplePDFax/stdEncryptSigned-02.pdf

7 Normative References

[pdf]
Adobe Systems, "PDF Reference, third edition, Adobe Portable Document Format Version 1.4", Addison-Wesley, December 2001, **http://partners.adobe.com/asn/developer/acrosdk/docs/filefmtspecs/PDFReference.pdf**. Also see errata: http://partners.adobe.com/asn/developer/acrosdk/docs/PDF14errata.txt.

[pdf-ppk]
Pravetz, J., "PDF Public-Key Digital Signature and Encryption Specification", Version 3.2, Adobe Systems, September 2001, **http://partners.adobe.com/asn/developer/pdfs/tn/ppk_pdfspec.pdf**

[ps-jpeg]
Adobe Systems Incorporated, "Supporting the DCT Filters in PostScript Level 2", November 1992, http://partners.adobe.com/asn/developer/pdfs/tn/5116.DCT_Filter.pdf

[ps]
Adobe Systems Incorporated, "PostScript Language Reference third edition", Addiseon-Wesley, 1999, http://partners.adobe.com/asn/developer/pdfs/tn/PLRM.pdf. Also see errata: http://partners.adobe.com/asn/developer/pdfs/tn/PSerrata.txt.

[ifx]
Moore, Songer, Hastings, "IPPFAX/1.0 Protocol" PWG Draft Standard D0.12, 2002, ftp://pwg.org/pub/pwg/QUALDOCS/pwg-ifx-pdfax-D12-021028.pdf

[ifx-req]
Moore, P., "IPP Fax transport requirements", October 16, 2000, ftp://pwg.org/pub/pwg/QUALDOCS/requirements/ifx-transport-requirements-01.pdf

[T.4]
ITU-T Recommendation T.4, "Standardization of group 3 facsimile apparatus for document transmission", October 1997

[T.6]
ITU-T Recommendation T.6, "Facsimile coding schemes and coding control functions for group 4 facsimile apparatus", November 1988

[RFC2119]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, September 2000, http://www.rfc-editor.org/cgi-bin/rfcdoctype.pl?loc=RFC&letsgo=2119&type=ftp&file_format=txt.

[RFC2911]
Hastings, Herriot, deBry, Isaacson, Powell, "Internet Printing Protocol/1.1: Model and Semantics", September 2000, http://www.rfc-editor.org/cgi-bin/rfcdoctype.pl?loc=RFC&letsgo=2911&type=ftp&file_format=txt.

[jpeg]
JTC 1/SC 29, “Information technology – Digital compression and coding of continuous-tone images: Requirements and guidelines”, ISO/IEC 10918-1:1994, 1994.

[jbig2]
JTC 1/SC 29, “Information technology – Lossy/lossless coding of bi-level images”, ISO/IEC 14492:2001, December 2001.

[RFC1950]
Deutsch, Gailly, “ZLIB Compressed Data Format Specification version 3.3”, May 1996, ftp://ftp.isi.edu/in-notes/rfc1950.pdf.

[RFC1951]
Deutsch, “DEFLATE Compressed Data Format Specification version 1.3”, May 1996, ftp://ftp.isi.edu/in-notes/rfc1951.pdf.

8 Informative References

[RFC2542]
Masinter , "Terminology and Goals for Internet Fax", RFC2542, March 1999, http://www.rfc-editor.org/cgi-bin/rfcdoctype.pl?loc=RFC&letsgo=2542&type=ftp&file_format=txt.

9 Revision History (to be removed when standard is approved)

Revision	Date	Author	Notes
1	10/9/02	Rick Seeler, Adobe Systems	Initial version
2	10/23/02	Rick Seeler, Adobe Systems	

10 Contributors

John Pulera - Minolta mailto:jpulera@minolta-mil.com
Gail Songer - Peerless mailto:gsonger@peerless.com
Tom Hastings - Xerox mailto:hastings@cp10.es.xerox.com
Rob Buckley - Xerox mailto:rbuckley@crt.xerox.com
Lloyd McIntyre - Xerox mailto:Lloyd.McIntyre@pahv.xerox.com

11 Acknowledgments

Kari Poysa - Xerox mailto:Kari.Poysa@usa.xerox.com

12 Author’s Address

Rick Seeler
Adobe Systems Incorporated
321 Park Ave., E13
San Jose, CA 95110
Phone: 1+408 536-4393
Fax: 1+408 537-8077
e-mail: mailto:rseeler@adobe.com